Intelligent Traffic Signal Automation Based on Computer Vision Techniques Using Deep Learning

Muhammad Talha Ubaid
Al Khawarizmi Institute of Computer Science
University of Engineering and Technology, Lahore
muhammadtalhaubaid@gmail.com

Hafiz Umer Draz
Al Khawarizmi Institute of Computer Science
University of Engineering and Technology, Lahore
umer.daraz@kics.edu.pk

Muhammad Usman Gani Khan
Al Khawarizmi Institute of Computer Science
University of Engineering and Technology, Lahore
usman.ghani@kics.edu.pk

Tanzila Saba
Artificial Intelligence & Data Analytics Lab CCIS
Prince Sultan University
Riyadh Saudi Arabia
drstanzila@gmail.com

Amjad Rehman
Artificial Intelligence & Data Analytics Lab CCIS
Prince Sultan University
Riyadh Saudi Arabia
rkamjad@gmail.com

Hoshang Kolivand
Department of Computer Science, Liverpool John Moores University, Liverpool, UK, L3 3AF
School of Computing and Digital Technologies, Staffordshire University, Staffordshire , UK
H.Kolivand@ieee.org

***Abstract*—Traffic congestion in highly populated urban areas is a huge problem these days. A lot of researchers have proposed many systems to monitor traffic flow and handle congestion through different techniques. But the current systems are not reliable enough to perceive traffic signals in real-time. Therefore, we aim to build a system that can efficiently perform real-time environments to solve the traffic congestion problem through signal automation. Since vehicle detection and counting are crucial in any traffic system, we use state-of-the-art deep learning techniques to detect and count vehicles in real-time. We then automate the signal timings by comparing the count of traffic on all sides of a junction. These automated signal timings sufficiently reduced congestion and improve traffic flow. We prepared a dataset of 4500 images and achieved about 91% accuracy by training it on Faster RCNN.**

Keywords*— vehicle detection, vehicle counting, signal automation, faster RCNN

I. Introduction

In 2019, the global population grew by one billion people to 7.7 billion, up from 7.6 billion in 2007. The world's population will be between 8.5 and 8.6 billion people by 2030 [1]. The number of cars on the road increases as the population grows. Therefore, a lack of successful traffic management will result in significant economic losses in energy consumption, greenhouse gas emissions, and time. With continued population growth, traffic congestion has become one of the most significant impediments to city economic development, resulting in high fuel consumption, increased commute costs, and pollution of the atmosphere.

Traffic congestion is a significant problem in urban areas that we all face in our daily life that never-ending patience, hoping the roads will clear up before we get late for whatever job or places we're going. It makes us feel impotent while wasting time and related resources such as fuel, energy, and effort. Pakistan's most populous & popular cities, such as Karachi **[2]**, Lahore, Faisalabad, etc., face a dire traffic congestion condition. An approximately Rs 47.9 million yearly and 1 million per week is wasted due to congestion at a patch of 1.1 kilometers of a road in Lahore. At traffic signal junctions, mainly the traffic jam happens because most of the signals are working traditionally. The green signal opens for a fixed number of seconds for each side without caring about the congestion on either side of the road. In [3], Zhang et al. proposed a system for improving the traffic flow by using the sensor that observes the traffic flow for priority purposes; the traffic lights were controlled [4].

To overcome this problem, we proposed signal automation as signal timings are crucial in the junction. The traffic signal automatically updated on the presence of the number of vehicles at the intersection. The proposed system will get a video from the camera installed on the traffic junction. From that video, frames are extracted, then from these vehicles are recognized and counted. Afterward, updates the green and red light time of junction on the base of vehicle count. We fine-tuned the Faster RCNN algorithm that performs very well for our problem and gives us the best accuracy and results for recognition purposes. We also generate a local dataset for the research community that will be available soon. Traffic signal automation means counting the number of vehicles at the junction for each side and adjusting the signal times of all sides, i.e., updating every side's green time.

The rest of the paper is organized in the following sections. Section 2 presents a literature survey of the existing works in this area using deep learning and computer vision techniques. Section 3 explains our methodology that offers an overview of solving traffic congestion problems through convolutional neural networks and state-of-the-art deep learning techniques. In Section 4, we overview our dataset and explain the results we achieve through experiments. Finally, Section 5 concludes the research discussion with a brief conclusion and gives future work suggestions.

II. Literature Survey

Vehicle detection and counting is a significant challenge in traffic control systems. Most designs can easily detect traffic speed and volume, but measuring traffic density is still a major challenge these days **[4]**. Since traffic density is the most decisive parameter in automated traffic control systems, researchers focus on computer vision and deep learning techniques to detect track and count vehicles **[5][6]**. These techniques have gained very accurate results in real-time compared to traditional methods and revolutionized the IOV(Internet of Vehicles) field.

For detection purposes, different techniques were used in the past, like many used background subtraction techniques for detecting stationary objects and foreground subtraction using background subtraction technique [7], subtract recent frame from the background and then detects the moving objects. Parekh et al. [8] concentrated on tracking objects by splitting them into three stages to identify items, recognize objects, monitor objects, and contrast the methods suggested in each step. Shahre and Shende [9] focused on identifying moving objects in the face of a static and dynamic context, and their concern was fixed camera and moving image. In [10], Liu et al. proposed the multi-intersections and multi phases traffic light systems. They use Adaptive Dynamic Programming (ADP) to find the best and optimal traffic signal policy.

The current deep learning techniques for detecting and tracking vehicles are supervised, unsupervised, or reinforcement learning. Reinforcement learning has been in use since the 1990s for traffic control. Zhang et al. **[11]** used deep reinforcement learning(DRL) in a partially observable environment for DSRC based intelligent traffic signals (ITS). The proposed system has a low waiting time for vehicles detected at an intersection, even when the detection rate is low. In **[12]**, Khan et al. proposed a methodology for detecting and classification the six different vehicles using CNN.

Wan et al. **[13]** summarize the DRL methods used for adaptive traffic signal control(ASTC). They also propose a DRL-based ASTC method with two neural networks DNN and CNN. They find that their proposed CNN is competent in feature extraction and learning for different traffic movement situations. But existing research using reinforcement learning has some limitations. Traffic signals with fixed-time intervals and random switching of signals make these methods inefficient and unsafe for drivers.

The background subtraction method **[14]** detects vehicles using low rank and sparse decomposition. The proposed algorithm uses the Kalman filter algorithm to track vehicles in multiple frames to avoid repetitive counting. Tayara et al. **[15]** proposed a system based on Convolutional Regression Neural Networks for vehicle detection and counting in aerial images. The resulting method gives good precision and recall compared to shallow-learning or deep-learning-based methods and can detect vehicles in high-resolution images.

Zhe et al. [16] proposed a video-based vehicle detection and counting framework. The proposed system uses a three-step process, vehicle detection, tracking, and vehicle path processing, to obtain information about traffic flow. The model is trained using YOLO, and the resulting system gives 90% accuracy on Vehicle Counting Dataset (VCD).

The existing methods still have some limitations. These systems are either slow, expensive, or not very accurate. Hence they are not suitable for real-time processing and cannot handle congestion precisely—the existing method used sensors that had some specific range for detecting the vehicles. In contrast, we used a camera capable of detecting vehicles up to 20 meters of distance. The proposed system solves these limitations. Faster RCNN is used to get accurate results in minimum time and manages congestion at all four sides of a junction independently. Also, current systems mostly focus on a specific type of situation. We build a flexible approach to work on problems and data, like weather, day or night conditions, highways, different road structures, and junction or additional vehicles, if the data is adequately trained. The previous system used the rule-based approach for the detection of the region. In Faster R-CNN Region Proposal Network (RPN) proposed the region for the detection. Before this, these detections were not trainable, and the backbone architecture was also trainable. The backbone architecture used the ResNet-101 for the classification.

III. METHODOLOGY

We build up a decentralized architecture that can be implemented at the traffic signal intersections on aspect computing gadgets. The high-stage evaluation of the proposed model at traffic signal junction is shown in Fig. 1.

The problem is to automate the traffic signals placed at the junction using deep learning techniques and design our algorithm for the traffic flow. As for the deep learning models, we use the Faster R CNN [17]. The proposed system improves the traffic infrastructure at intersections on the road. For this, we estimated the density and count of the vehicle from every side of the junction. The green and red light time is updated on the basis of the count of the vehicle on every side of the junction. Vehicle detection is quite complicated and challenging, especially from different camera views and angles. For vehicle detection vehicles, notable designs, i.e., hand-crafted features and occlusion modeling, are used **[18]**.

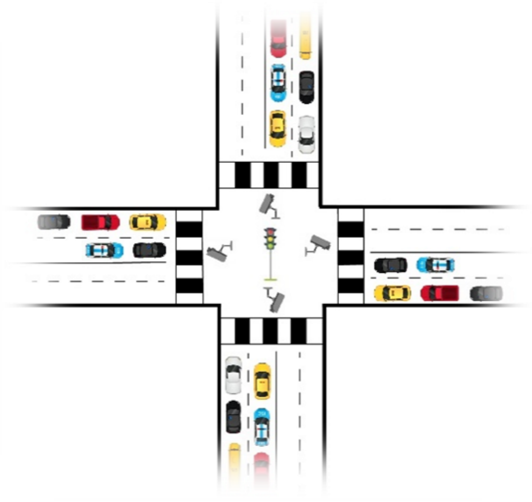

Fig. 1. Traffic Signal Junction

These methods are getting outdated as the current approaches are based on deep neural networks. Excellent results are achieved on large datasets through these procedures. There are specific vehicle counting problems, such as the vehicles' overlapping, diversity of different vehicles, perspective view, and the minimum size of detected objects. For overlap, we overcome this by doing efficient annotation of our dataset. There are different types of vehicles to deal with diversity. The collected dataset was suitable for the dedicated problem. To get the view of every lane from the junction, four different cameras have been used. Capturing every lane's position around the intersection, cameras are placed at each lane's start at a specific height and position. For instance, in Lahore city, different cameras are installed at almost every traffic intersection by the Safe City. Still, those are not placed according to the requirements of our problem. Directions of camera and traffic must be aligned to get a better and detailed view of lanes.

The system takes input from these CCTV cameras. The algorithm is designed to regulate the intersection's traffic flow and shows a system diagram of the model in Fig. 2. A traditional traffic signal has a fixed time of cycle green light for intersection and updates the red-light time for other sides. The system's primary effort is to automate that green light time and update each side's red-light time on every cycle. At first, it initialized the same time of green light to all sides of the junction. Red light time is also constant for every side in the initial run. As from the traffic signal management system, one side is always open for traffic flow. A few seconds before the green light time of a particular side, this model would take the image frame from the video on CCTV cameras. Images are taken from all three other sides of the traffic intersection.

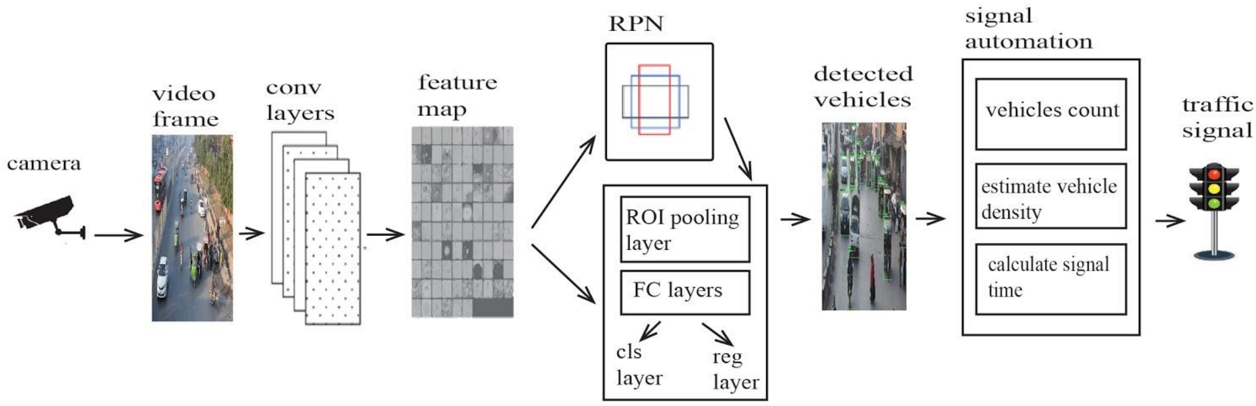

Fig. 2. System Diagram

From these images, vehicles are classified and counted. The detection of vehicles was real-time, as in detecting the vehicle and counting before updating the green time. The system was trained on Faster R CNN. In this, real-time detection and counting of vehicles have been done. After obtaining the vehicles' count for each closed side, each side's green light time may increase or keep constant. The red-light time of every closed side on every cycle has to be modified. This green light depends on the vehicle count. If the vehicles are increased at a specific rate, then timers could be adjusted according to the count of the flowing traffic. The red-light time also increased or kept stable for closing sides of the intersection.

In this model, the input is taken from the CCTV camera frames are being taken out from the video. It is also made sure that the user frame is not blurred or fuzzy. For an instant, initially set the time of green light was at 10 seconds for each side of the traffic signal junction and red-light time would be 30 seconds for each side—five seconds before the ending time of the green light of side A. Image frames are attained from all three remaining sides B, C, and D cameras. From these vehicles are detected and counted using trained model faster R CNN. It gives us the count of vehicles for sides B, C, and D. Now, green light time is modified according to the number of vehicles (used parameter ten vehicles in 1 sec). As the number of vehicles on sides B, C, and D increases, we also update the green time, and the red-light time is also updated for other sides. The next step before expiring side B calculates the traffic flow for the sides of A, C, and D. We updated the A and C green light time and kept the D time the same because there is no much change in the count side D. For the particular threshold of vehicles increase, we also increase the timer while we won't increase the time in case of a change of values. This is the entire cycle of traffic signal intersection is completed.

Detecting and counting the number of vehicles is the sole purpose of this model. The whole project's accuracy lies in this, for the training of an extensive experiment using Faster R CNN on the dataset. In Faster RCNN, an image as input is passed to ConvNet. This input image, first of all, is resized. ConvNet then extracts features from that image. The extracted features by ConvNet are smaller than the given input image. Feature size depends on the convolutional network's stride; after that, the network needs to learn whether an object is present in the given image for each point of the feature map and guess its size.

For this purpose, anchors are placed over the input image for every output feature map extracted by a convolutional network. Such anchors represent objects and their possible sizes at this position in different dimensions and respective aspect ratios. As RPN runs over each pixel of the output feature map, it checks whether the anchors spread over the input image have any objects and cleans up those anchors and coordinates to make bounding boxes for a region of interest. For this purpose, a 3x3 convolution is applied with 512 units to the convolutional network feature map, which returns two layers first 1x1 convolutional with 18 object classification units and the second 1x1 convolutional with 36 bounding box regression units. Then 18 classification units are used to give probabilities for every point in the convolutional network, whether it contains the object or not. And 36 regression units are used to find regression coefficients used to improve coordinates of anchors with objects. The output feature map has a fixed number of anchors. If the number of anchors crosses the fixed boundary at training time, then those anchors are ignored, and the anchors that are ignored will not contribute to the loss. Training loss for the dataset is calculated using the given equation (1). Here i is the index of an anchor and p_i tells the probability of anchor I containing an object. p_i^* is the actual label of the object, and it is 1 for positive anchor and 0 for negative anchor t_i is a vector containing four coordinates of the predicted bounding box and t_i^* Is the vector containing the ground-truth bounding box coordinates. L_{cls} is the total loss across the classification layer and L_{reg} is the total loss across the regression layer. N_{cls} and N_{reg} represent the total number of training examples for classification and regression layers, respectively. Lambda is the regularization hyperparameter for balancing weights across these layers

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}\,(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}\,(t_i, t_i^*) \quad \textbf{(1)}$$

.

IV. Experiments and Results

A system is designed that managed traffic congestion for which a customized real-time dataset was prepared.
The architecture of the Faster RCNN as the proposed model was also modified, which eventually improved its working and result compared to the other models on which the dataset was trained/tested.

A. Data Set

Data plays a vital role in deep learning models. The problem with deep learning algorithms is that they're data-hungry. To get the competent result data set should be according to the problem. For the system, a custom dataset was prepared. Different videos with 5 minutes of duration were recorded. The dataset was collected at different points of traffic intersection in Lahore city.

We have almost every vehicle category in these vehicles, i.e., bus, car, bike, rickshaw, truck, loader, Quincy, etc. We extract the image frame from these videos at the rate of 60 frames per minute from each video. So we get almost 9000 images. Afterward, we cleaned that data and labeled these images. Annotation had been done to these cleaned images using the LabelImg tool. We labeled detect vehicles as "Vehicle" class.

Discussion

The model was implemented on the Nvidia 1080 ti GPU of 10 GB, having 32GB of RAM, and trained on Faster R CNN. While training used tensor flow and Nvidia CUDDN libraries to get things started. The training and testing of others model for comparison was also done it.

There was a total of 4500 images to train the model. The model was trained with parameter setting of batch size equal to one, initial learning rate 0.0003, momentum 0.9 and 7000 epochs. In the beginning, the loss was around 3.5468 as the model continued to train the loss decreases. At the same time, the accuracy of the model also increased.

The system's accuracy is more than 91% after 5000 epochs to detect vehicles trained on our own data set. The model was also trained on the previous paper's proposed model [18] for comparison and achieved 87.8% accuracy. The results showed that the model used in the proposed methodology is more accurate than the YOLO model used in the previous paper. The Signal Shot Detector (SSD) [19] was also trained on the dataset and achieved 83.33% accuracy for the detection of the vehicles. The comparison of different model's accuracy on the customized dataset is shown in Table 1. The results are better with the proposed method as compared to Yolo and SSD.

Table 1. Comparison of Different Models on our dataset with a similar setting.

Models	Accuracy	FPS
Fined Tuned Proposed Faster R-CNN	91%	16
SSD [19]	83.33%	19
YOLOv3 [20]	87.8 %	20.7

Additionally, we evaluated the VDD dataset **[17]** on the proposed model and achieved better accuracy on the dataset using Faster RCNN. Fig. 3. shows that Faster RCNN gives 92.8% accuracy, while the YOLO model achieved 90% accuracy on VDD. Hence the proposed model is better at detecting vehicles on any dataset.

The proposed system did real-time working as it counts the number of vehicles on each side of the junction and adjusts the traffic signal's timer. We tested this system in the simulation environment.

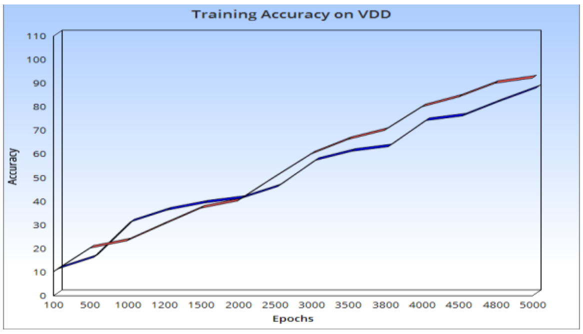

Fig. 3. Accuracy on our dataset using Faster RCNN and YOLO

V. Conclusion

In this paper, an automatic signal automation system based on traffic videos is proposed. The results show that the dataset's detection accuracy using fined tuned Faster RCNN is very high, even for complex situations. Moreover, our automation algorithm efficiently switches the signals based on traffic intensity on different sides of a junction. The running rate of our system is pretty high, making it suitable for real-time situations. In the future, we plan to collect datasets for different scenarios and train them on our model. We also plan to detect and count pedestrians and other huge vehicles on highways and install our system in other cities and settings.

References

[1] UN DESA. World Population Prospects 2019: Highlights (ST/ESA/SER. A/423); United Nations Department for Economic and Social Affairs: New York, NY, USA, 2019.

[2] Ali MS, Adnan M, Noman SM, Baqueri SF. Estimation of traffic congestion cost-a case study of a major arterial in Karachi. Procedia Engineering. 2014 Jan 1;77:37-44

[3] Tufuor EO, Rilett LR. Validation of the Highway Capacity Manual urban street travel time reliability methodology using empirical data. Transportation research record. 2019 Apr;2673(4):415-26.

[4] Tran, P. V., Bui, T. T., Tran, D., Pham, P. Q., & Tran, A. V. T. (2016, March). Approach to priority-based controlling traffic lights. In *Asian Conference on Intelligent Information and Database Systems* (pp. 745-754). Springer, Berlin, Heidelberg.

[5] Hu HN, Cai QZ, Wang D, Lin J, Sun M, Krahenbuhl P, Darrell T, Yu F. Joint Monocular 3D Vehicle Detection and Tracking. InProceedings of the IEEE International Conference on Computer Vision 2019 (pp. 5390-5399).

[6] Seenouvong N, Watchareeruetai U, Nuthong C, Khongsomboon K, Ohnishi N. A computer vision based vehicle detection and counting system. In2016 8th International Conference on Knowledge and Smart Technology (KST) 2016 Feb 3 (pp. 224-227). IEEE.

[7] Bayona Á, SanMiguel JC, Martínez JM. Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques. In2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance 2009 Sep 2 (pp. 25-30). IEEE.

[8] Parekh HS, Thakore DG, Jaliya UK. A survey on object detection and tracking methods. International Journal of Innovative Research in Computer and Communication Engineering. 2014 Feb;2(2):2970-8

[9] Shende MD. A survey on moving object detection in static and dynamic background for automated video analysis.

[10] Liu, D., Yu, W., Baldi, S., Cao, J., & Huang, W. (2019). A Switching-Based Adaptive Dynamic Programming Method to Optimal Traffic Signaling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

[11] Zhang R, Ishikawa A, Wang W, Striner B, Tonguz O. Intelligent traffic signal control: Using reinforcement learning with partial detection. arXiv preprint arXiv:1807.01628. 2018 Jul 4.

[12] Khan, M. Z., Hassan, S. U., Draz, H. U., Khan, M. U. G., & Iqbal, R. (2020). Detection and Identification of Vehicles from High-Resolution Aerial Images Using Deep Learning Approaches with the Tuned Parameters. In *Unmanned Aerial Vehicles in Smart Cities* (pp. 65-83). Springer, Cham.

[13] Wan CH, Hwang MC. Adaptive Traffic Signal Control Methods Based on Deep Reinforcement Learning. In Intelligent Transport Systems for Everyone's Mobility 2019 (pp. 195-209). Springer, Singapore.

[14] Choudhury S, Chattopadhyay SP, Hazra TK. Vehicle detection and counting using a haar feature-based classifier. In2017 8th annual industrial automation and electromechanical engineering conference (IEMECON) 2017 Aug 16 (pp. 106-109). IEEE.

[15] Tayara H, Soo KG, Chong KT. Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network. IEEE Access. 2017 Dec 11;6:2220-30.

[16] Dai Z, Song H, Wang X, Fang Y, Yun X, Zhang Z, Li H. Video-based vehicle counting framework. IEEE Access. 2019 May 1;7:64460-70.

[17] Ren S, He K, Girshick R, Sun J. Faster r-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems 2015 (pp. 91-99).

[18] S. Sivaraman and M. M. Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. Intelligent Transportation Systems, IEEE Transactions on, 14(4):1773–1795, 2013.

[19] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

[20] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.