I. J. Engineering and Manufacturing, **2021, 6, 11-18**
Published Online December 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2021.06.02

A Reinforcement Learning-based Offload Decision Model (RL-OLD) for Vehicle Number Plate Detection

Yadavendra Atul Sakharkar
SCOPE, Vellore Institute Technology, Vellore, TN, India
Email: prajwal1619@gmail.com

Mrinalini Singh
SCOPE, Vellore Institute Technology, Vellore, TN, India
Email: Mrinalini1012@gmail.com

Kakelli Anil Kumar
Associate Professor, SCOPE, Vellore Institute Technology, Vellore, TN, India
Email: anilsekumar@gmail.com

Aju D
Associate Professor, SCOPE, Vellore Institute Technology, Vellore, TN, India
Email: daju@vit.ac.in

Received: 30 July 2021; Revised: 12 August 2021; Accepted: 28 August 2021; Published: 08 December 2021

Abstract: Vehicle license number plate detection is essential for road safety and traffic management. Many existing systems have been proposed to achieve high detection precision without optimization of computer resources. Existing models have not preferred to use devices like smartphones or surveillance cameras because of high latency, data loss, bandwidth costs, and privacy. In this article, we propose a model of unloading decisions based on reinforcement learning (RL-OLD) for recognition and detection of vehicle license plates for high precision with optimization of computer resources. The proposed model detected different categories of vehicle registration plates by effectively utilizing edge computing. Our model can choose either the compute-intensive model of the cloud or the lightweight model of the local system based on the properties of the number plate. This approach has achieved high accuracy, limited data loss, and limited latency.

Index Terms: Edge computing, Decision Offloading, License plate, Recognition, YOLOV3, SSDLite MobilenetV2, Reinforcement Learning.

1. Introduction

Edge computing allows computing on the local device, which solves many issues related to deploying the deep learning model into the cloud. However, the greatest limitation is the limited computing power. Our model is therefore integrated into the proposed application which will use the two technologies such as edge computing and the deep learning model of the cloud. Hence, we get a trade-off between speed and accuracy. Vehicle registration plate recognition is a better fit for this application run on mobile devices and monitoring devices are the primary peripherals for the application. The detection of the numbers on the number plate is the most difficult task to recognition with high accuracy as shown in figure 1. We have modeled YOLOV3 model as the detection model on the cloud and for processing the image data we have used a computing system. The YOLOV3 model is an object detection model to detect license plates up to 97% on the data set of 400 images. It was tested on various images with different parameters like the size of the numbers and position. The same model was found to be the best of the industry according to Laroca R. et al [1]. For the local model, we modeled a lightweight model known as MobileNetV2-SSDLite that uses SSD (Single-shot Multibox Detector) and MobilenetV2 for high accuracy using limited computing power. The mobile device is using an edge device and tested. This model detected registration plates in approximately 94.5% of the 400 images during the initial test phase. The proposed RL-OLD model is preferred on different images with different angles and illumination. RL-OLD used Google's Tesseract-OCR engine to identify every character in the license plates. To achieve precise accuracy with RL-OLD, difflib is used to match sequences to calculate the similarity between outputs. Accuracy proved to be 90.7% of the images detected by the YOLOV3 model, and 82.5% of the images detected by the SSD-Lite model individually.

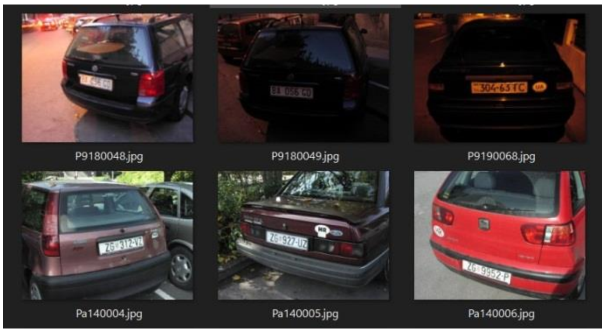


Fig.1. Images of cars in different conditions

2. Related Work

Li Et al. [2] have proposed a system model which is idle for the environment with multiple users and various offloading points, and structured tasks. To overcome the complexity, the authors suggested an approach based on enhanced genetics To obtain the solution, backtracking was used. algorithm with a greedy strategy. Its performance was good based on the total cost of all users, the time required to complete it, and the total use of resources. Fang Liu et al. [3] have reviewed various existing systems and open-source projects which were based on edge computing and categorized based on their innovations and design. Recent advanced systems can reduce processing time and improve the efficiency of mobile data analysis. This study established several energy-saving and efficiency strategies in terms of performance and deployed deep learning models.

In edge computing, data processing and computing can take place on the peripherals of a network in proximity to end-users. Edge computing in our model can work on the cloud and its devices for downstream data and upstream data respectively. Some of the many problems of state-of-the-art computing are programmability, naming, privacy, and security. In the edge computing paradigm, the “things” are data consumers and data producers. These edges are not only performed data storage but also computing offloads [4, 5, 6] creating some synergy between the “edge computing with it” and the emerging blockchain and AI technologies that could potentially generate many beneficial impacts [7]. The researchers have proposed various methods for license plate recognition systems. Each method has three main steps which are extraction of the area of interest, extraction of numbers on the plate, and character recognition. Automatic License Plate Recognition (ALPR) was found to be efficient for traffic monitoring [8]. A robust real-time end-to-end ALPR system was built using state-of-the-art YOLO object detection CNN's tested on the datasets of 4500 images. The main two steps are character segmentation and recognition stages [9]. N Saif et. al [10] has proposed a system to detect and recognize Bangla license plate by using CNN. CNN was used for its configuration for the end-to-end pipeline. This model was implemented using YOLOv3 which had 53 model layers in the dataset of 200 images. Anish Lazrus et al. [11] suggested a method for detecting the license plate from a grayscale image and segment the image of the characters contained by the Indian License plate. This approach used the wiener2 filter to eliminate noise in the images and the Sobel filter to smooth the images and then the bw tag to calculate the connected component. Their results were observed using 50 images of different registration plates from different states. Hanit Karwal et al. [12] is developed an efficient method for recognition for Indian vehicle number plates. To address issues related to scaling and recognition of the position of characters, this study has proposed a VNPD System algorithm based on template matching. The author has proposed in this work an algorithm that is enhanced Otsu’s method for threshold partitioning. In another study [13] proposed a segmentation-free method and object detection based on a state-of-the-art deep learning model. Their dataset contains 2000 real-life images captured on roads. Priyanka Prabhakar et al. [14] proposed an approach for localization, segmentation, and recognition of the characters within the number plate. The pictures were captured using a stationary camera or from video sequences. Later these pictures were converted into grayscale images. The main purpose of this study is to specify that the proper approach can accomplish high accuracy by optimizing various constraints that have an advanced recognition rate than the typical ways. Ramneek et al. [15] worked on

challenges related to multi-access edge computing (MEC) for supporting network slicing for QoS provisioning in different 5G use cases and discussed how the operating system (OS) level optimizations can help to overcome them. P.K. Suri et al. [16] developed the method that used Sobel Edge detection to detect the number present in vehicle plates. In addition to the Sobel edge detection technique, the author used the masking, smoothing, simple color conversion, edge detection, and connector measurement technique. Mahesh Babu K et al. [17] built a method of vehicle number plate detection and recognition using Bounding Box Method. The author suggested four main steps preprocessing of a captured image, extracting license number plate region, segmentation, and character recognition of license plate.

Gopika premsankar et al. [18] advocated advanced computing for emerging IoT applications that exploit sensor streams to increase interactive applications. They concluded the achievement of the current edge computing platforms and how emerging technologies will impact the deployment of future IoT applications. Satyanarayanan et. al [19] brought up the idea of a virtual machine that was based on Cloudlets as one of the first works on Cloudlet Computing. A small cell base station with Intel's T3K Concurrent Dual-Mode system-on-chip (SoC) has a 4-core ARM-based CPU and limited memory, which is insufficient for running Apache Spark that requires at least 8-core CPU and 8-gigabyte memory for good performance [20]. The algorithm may divide individual portions of the unloading task group into different groups, each with very specific combinations of unloading characteristics. The findings of this research depict that the proposed approach can save around 50 percent of the energy needed as compared with local execution while only slightly sacrificing response time [21].

3. The Proposed Methodology

This section focuses on the approach taken for license plate recognition. This section is divided into 3 sections namely Detection, Recognition, and RL-based offload decision model. Figure 2 shows the flow of the license plate recognition system.

3.1. Detection Model

The detection has been performed using two different models namely YOLOV3 and SSDLite-MobilenetV2. Even though YOLOV3 has the best performance it could only be used as a cloud model as it requires us to store weight files of at least 250MB. The detection was done with two different models, YOLOV3 and SSDLite-MobilenetV2. Hence SSDLite model was chosen as the local model as it is very lightweight and also provides good accuracy. It also takes a much shorter time to detect than YOLOV3. All the datasets were collected from Kaggle and GitHub and images from various conditions. YOLOV3 originally has 53 layers and it uses 53 more layers for object detection.

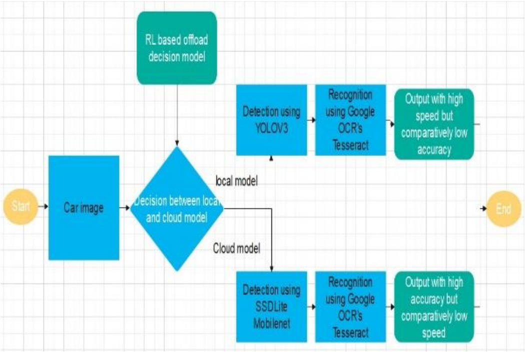

RL based offload decision model
Detection using YOLOV3
Recognition using Google OCR's Tesseract
Output with high speed but comparatively low accuracy
local model
Start
Car image
Decision between local and cloud model
End
Cloud model
Detection using SSDLite Mobilenet
Recognition using Google OCR's Tesseract
Output with high accuracy but comparatively low speed

Fig. 2. Flow diagram of the proposed system

Table 1. Structure of MobileNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

We tested it on the dataset which consists of English license plates. The model was able to detect license plates from 97% of the images of the dataset containing images of cars in different conditions [9]. The proposed model is used for the local model MobileNetV2-SSDLite which integrates MobileNetV2 which is an improvement on MobileNetV1 and also uses SSD (Single-shot multi-box detector). SSD is an object detection network that can directly predict the target category and location [25, 26]. We tested this model on our dataset and it was able to detect 94.5 percent of the images of the dataset containing images of cars in different conditions as shown in table 1.

3.2. Recognition of Characters

Now that the detection was done the next task is to check how well the images have been detected. Google's Tesseract-OCR Engine was used with a custom config to recognize the characters in the cropped license plates. First, the cropped images were converted to gray-level images to increase the accuracy and to reduce redundant information on the license plate. To calculate the accuracy of the output that was produced by tesseract, a python library called difflib was used which uses sequence matching between the original label and the output obtained [22, 23, 24]. The accuracy obtained was 90.7% of the images detected from YOLOV3 model and The accuracy obtained was 82.5% of the images detected from the MobileNetV2-SSDLite model. This discrepancy in accuracy between the two models makes the use of the offload decision model all the more necessary. The average detection and recognition time of each image using the YOLOV3 and Tesseract model was 1.97 seconds. And the average time taken by each image to detect and recognize with the MobileNetV2-SSDLite and Tesseract model was 0.48 s. This means that there is also a time lag between the two models. Therefore, this solution could have different results on different specifications, but the design and model will always do the same job in these scenarios as well. This difference in time can be different based on the specifications of the cloud and mobile device, however, there will still a gap in time as the time take for the image to send to the cloud and get the response might be greater than the PC using the cloud technology.

3.3. Offload Decision Model

The proposed offload decision model is the Reinforcement Learning model because there are no definite labels that the prediction is based upon. However, the results are faster and also accurate, the two parameters that can be used as pseudo labels are speed and accuracy. Now we first calculated the time and accuracy for both models on 400 images and then saved it in a CSV file. Now both the speed and accuracy have to create 1 value and hence each image will have 2 values corresponding to YOLOV3 model and SSDLite-MobileNetV2 model. However, both speed and accuracy should be normalized as both factors are important. Hence we used a normalizing function given below to normalize both time and accuracy to range from 0 to 1.

$$newVal = \frac{oldV\ al - minV\ al}{(maxVal - minVal)} \tag{1}$$

In this formula, minVal and maxVal are the minimum and maximum values of both YOLOV3 values and SSDLite values together. This formula is used on both accuracy and speed to normalize it to range between 0 to 1. Then time is convertedto speed using the below formula

$$Speed = (1 - time) \tag{2}$$

As time will only range from 0 to 1 and also it is inversely proportional to speed, this formula is derived. Now that we have the speed and accuracy of both the models we need to create a single variable from it. In our case we gave the same weightage to both time and accuracy, hence we used the below formula to convert the values to Qvalue.

$$QValue = \frac{speed + accuracy}{2} \tag{3}$$

This formula is used to create 2 values for each image corresponding to Qvalue of SSDLite model and YOLOV3 model. Now that the pre-processing and formation of labels is the building model is the next step. A CNN model is used to predict which model to use from the Qvalues and images provided.

4. Network Model

The CNN model developed is made up of 6 layers as the following

1. Layer 1: A convolutional layer kernel of size 5, a reLU layer, and a max-pooling layer of kernel size (2,2).
2. Layer 2: Repetition of the first layer.
3. Layer 3: Flatten layer which flattens all dimensions except batch.
4. Layer 4: Set of Linear layers and reLU layer
5. Layer 5: Repetation of the layer 4.
6. Layer 6: A linear layer with 2 output channels.

The model is trained on 200 images for 100 epochs as the model and the task itself are not too complicated. Afterward, the model predicts the 2 Qvalues, and the model corresponding to the bigger value is chosen as the final output.

5. Result Analysis and Discussion

License plates detected by the YOLOV3 model are shown in figure 3 and the license plates detected by the MobileNetV2- SSDLite model is shown in figure 4.

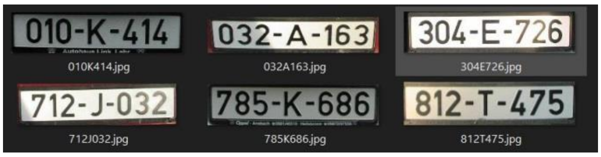


Fig. 3. License plates detected by YOLOV3

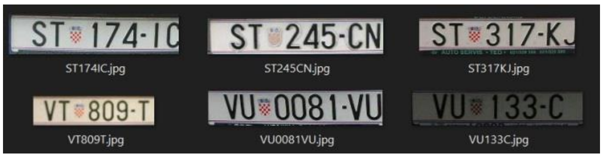
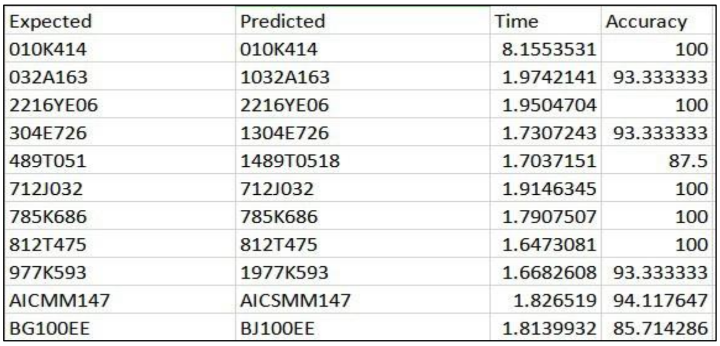


Fig. 4. License plates detected by MobileNetV2-SSDLite

From the above images, we can observe that the detected plates are very accurate in both cases. There were very few cases where the models were not able to detect the license plate which included the cases where the license plate not being visible to the naked eye and lighting problems, etc. The recognition results are shown in table 2 and table 3.

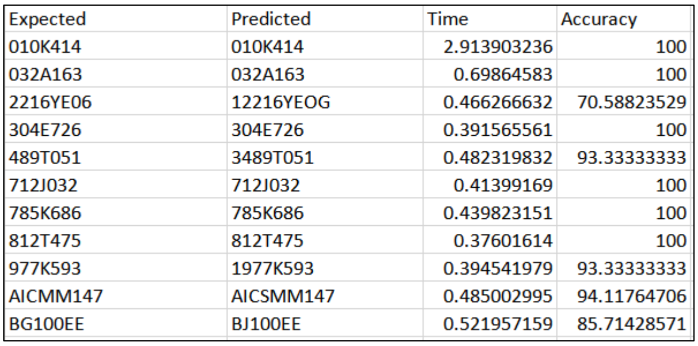
In many cases, results in YOLOV3 de-tested plates are more accurate compared to the results on MobileNetV2-SSDLite detected plates. It is also clearly visible that MobileNetV2-SSDLite is faster than the YOLOV3 model at the time given in table 2 and table 3 is the time taken for detection and recognition. In the case of time in YOLOV3 model the time also increases slightly due to the traveling time of the image to the cloud. The travel time may vary based on various situations.

Table 2. Recognition results on YOLOV3 detected plates



Expected	Predicted	Time	Accuracy
010K414	010K414	8.1553531	100
032A163	1032A163	1.9742141	93.333333
2216YE06	2216YE06	1.9504704	100
304E726	1304E726	1.7307243	93.333333
489T051	1489T0518	1.7037151	87.5
712J032	712J032	1.9146345	100
785K686	785K686	1.7907507	100
812T475	812T475	1.6473081	100
977K593	1977K593	1.6682608	93.333333
AICMM147	AICSMM147	1.826519	94.117647
BG100EE	BJ100EE	1.8139932	85.714286

Table 3. Recognition results on MobileNetV2-SSDLite detected plates



Expected	Predicted	Time	Accuracy
010K414	010K414	2.913903236	100
032A163	032A163	0.69864583	100
2216YE06	12216YEOG	0.466266632	70.58823529
304E726	304E726	0.391565561	100
489T051	3489T051	0.482319832	93.33333333
712J032	712J032	0.41399169	100
785K686	785K686	0.439823151	100
812T475	812T475	0.37601614	100
977K593	1977K593	0.394541979	93.33333333
AICMM147	AICSMM147	0.485002995	94.11764706
BG100EE	BJ100EE	0.521957159	85.71428571

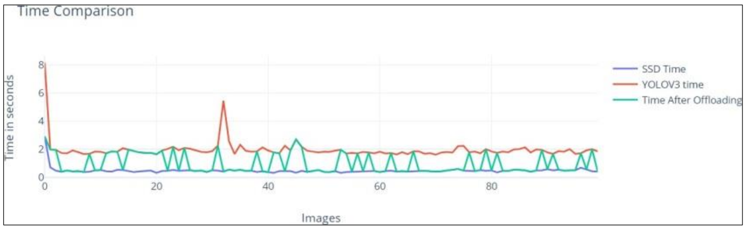


Fig. 5. Time comparison between YOLOV3, SSDLite-MobileNetV2, and final model after offloading

After the decision offloading model is trained and tested on 100 images the results were compared with YOLOV3 model and SSDLite-MobileNetV2 model. We can see from the graphs that at the end offloading decision is making use of both the models to get a good time and better accuracy as well. The final comparison between all the models is shown below in figure 6 and table 4. Based on the above results we can see that the average time taken by each image is between the average time of YOLOV3 and SSDLite MobileNetV2. But the best results are shown in the accuracy department as the accuracy is more than that we got into YOLOV3 model. This is because there are some cases where images detected by SSDLite-MobileNetV2 get better accuracy than YOLOV3. The offloading decision model is making use of SSDLite MobileNetV2 in some of these cases which increases overall accuracy and also this offloading decision model can also be used in other applications which have similar requirements. This result might vary based on various factors such as specifications and the time taken for the traveling of images and data may also vary based on various factors, however, the offloading decision model can be trained in those conditions and it will still perform well and providea good combination based on the given conditions.

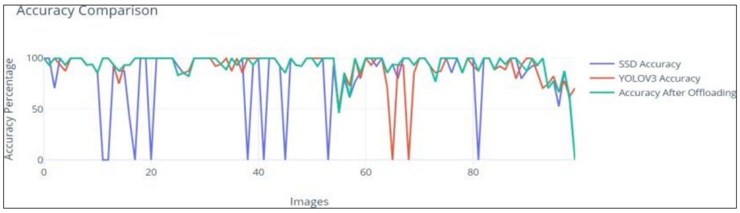


Fig. 6. Accuracy comparison between YOLOV3, SSDLite-MobileNetV2, and final model after offloading

Table 4. The comparative results of YOLOV3 and SSDLite MobileNetV2

Factors	**YOLOV3**	**SSDLite MobileNetV2**	**Final results after offloading**
Average Time	1.97 seconds	0.48 seconds	0.98 seconds
Average Accuracy	90.7 percent	82.5 percent	93.15percent

6. Conclusion

Local and cloud detection models have been integrated into the proposed system and both models can be used following the requirements. This strategy has made it possible to resolve the limitations of existing models that are only deployed in the cloud. Our proposed system is used and implemented on edge devices to process and make decisions with high accuracy and speed in a real-life situation. The load-off decision model based on reinforcement learning achieved the best results in terms of speed as the YOLOV3 model and recognition accuracy reached 93.15%. In future work, we test our proposed model on a variety of other datasets to get greater accuracy in real-time.

References

[1] Rayson Laroca, Evair Severo, Luiz A Zanlorensi, Luiz S Oliveira, Gabriel Resende Gonçalves, William Robson Schwartz, and David Menotti. A robust real-time automatic license plate recognition based on the yolo detector. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2018.

[2] Li Kuang, Tao Gong, Shuyin OuYang, Honghao Gao, and Shuiguang Deng. Offloading decision methods for multiple users with structured tasks in edge computing for smart cities. *Future Generation Computer Systems*, 105:717–729, 2020.

[3] Fang Liu, Guoming Tang, Youhuizi Li, Zhiping Cai, Xingzhou Zhang, and Tongqing Zhou. A survey on edge computing systems and tools. *Proceedings of the IEEE*, 107(8):1537–1562, 2019.

[4] Weisong Shi and Schahram Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, 2016.

[5] Kusumlata Jain and Smaranika Mohapatra. Taxonomy of edge com- puting: Challenges, opportunities, and data reduction methods. In *Edge Computing*, pages 51–69. Springer, 2019.

[6] Jianli Pan and James McElhannon. Future edge cloud and edge computing for internet of things applications. *IEEE Internet of Things Journal*, 5(1):439–449, 2017.

[7] Jianli Pan and Zhicheng Yang. Cybersecurity challenges and opportuni- ties in the new" edge computing+ iot" world. In *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, pages 29–32, 2018.

[8] Vinay Kumar and Saroj Kumar Gupta. A review paper on license plate recognition system. *European Journal of Business and Social Sciences*, 7(6):294–299, 2019.

[9] Saeed Khazaee, Ali Tourani, Sajjad Soroori, Asadollah Shahbahrami, and Ching Y Suen. A real-time license plate detection method using a deep learning approach. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 425–438. Springer, 2020.

[10] Nazmus Saif, Nazir Ahmmed, Sayem Pasha, Md Saif Khan Shahrin, Md Mahmudul Hasan, Salekul Islam, and Abu Shafin Moham- mad Mahdee Jameel. Automatic license plate recognition system for bangla license plates using convolutional neural network. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pages 925–930. IEEE, 2019.

[11] Anish Lazrus, Siddhartha Choubey, and GR Sinha. An efficient method of vehicle number plate detection and recognition. *International journal of machine intelligence*, 3(3):134–137, 2011.

[12] Hanit Karwal and Akshay Girdhar. Vehicle number plate detection system for indian vehicles. In *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, pages 8–12. IEEE, 2015.

[13] Alperen Elihos, Burak Balci, Bensu Alkan, and Yusuf Artan. Deep learning based segmentation free license plate recognition using roadwaysurveillance camera images. *arXiv preprint arXiv:1912.02441*, 2019.

[14] Priyanka Prabhakar, P Anupama, and SR Resmi. Automatic vehicle number plate detection and recognition. In *2014 International Confer- ence on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pages 185–190. IEEE, 2014.

[15] Seung-Jun Cha, Seung Hyub Jeon, Yeon Jeong Jeong, Jin Mee Kim, Sungin Jung, Sangheon Pack, Patrick Hosein, et al. Multi-access edge computing in 5g network slicing: Opportunities and challenges. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 30–32. IEEE, 2019.

[16] PK Suri et al. Vehicle number plate detection using sobel edge detectiontechnique. 2010.

[17] K Mahesh Babu and MV Raghunadh. Vehicle number plate detection and recognition using bounding box method. In *2016 International Conference on Advanced Communication Control and Computing Tech- nologies (ICACCCT)*, pages 106–110. IEEE, 2016.

[18] Gopika Premsankar, Mario Di Francesco, and Tarik Taleb. Edge computing for the internet of things: A case study. *IEEE Internet of Things Journal*, 5(2):1275–1284, 2018.

[19] Mahadev Satyanarayanan. The emergence of edge computing. *Com- puter*, 50(1):30–39, 2017.

[20] Blesson Varghese, Nan Wang, Sakil Barbhuiya, Peter Kilpatrick, and Dimitrios S Nikolopoulos. Challenges and opportunities in edge computing. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 20–26. IEEE, 2016.

[21] Huaming Wu, Yi Sun, and Katinka Wolter. Energy-efficient decision making for mobile cloud offloading. *IEEE Transactions on CloudComputing*, 8(2):570–584, 2018.

[22] Kumar, K. A. A. D,"An Internet of Thing based Agribot (IOT-Agribot) for Precision Agriculture and Farm Monitoring,". *Int. J. Educ. Manag. Eng*, *10*(4), 33-39, 2020.

[23] Kumar, K. A., & Dhadge, O. A Novel Infrared (IR) Based Sensor System for Human Presence Detection in Targeted Locations. *International Journal of Computer Network & Information Security*, *10*(12), 2018.

[24] Tiwari, M. G. D., & Kakelli, A. K. Secure Online Voting System using Visual Cryptography. *Walailak Journal of Science and Technology (WJST)*, *18*(15), 8972-14, 2021.

[25] Kumar, K. A., Krishna, A. V., & Chatrapati, K. S. Congestion control in heterogeneous wireless sensor networks for high-quality data transmission. In *Proceedings of the international congress on information and communication technology*, Springer, Singapore, pp. 429-437, 2016.

[26] Kumar, K. A., Krishna, A. V., & Chatrapati, K. S. (2016). Interference minimization protocol in heterogeneous wireless sensor networks for military applications. In *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2* (pp. 479-487). Springer, Cham.

Authors' Profiles

Mr. Yadavendra Atul Sakharkar is Graduate of Bachelor's of Technology in Computer Science and Engineering at Vellore Institute of Technology, India. His major fields of interest include Artificial Intelligence, Machine Learning, Computer Vision, Image Processing, Data Visualization and has worked on several research projects and papers for the same. He is a budding Computer Science Engineer with a keen interest in latest technology, MERN Stack Developer and an aspiring Data Scientist looking to further his knowledge and experience.

Ms. Mrinalini Singh is Graduate of Bachelor's of Technology in Computer Science and Engineering at Vellore Institute of Technology, India. She is an aspiring programmer, and event manager. She has developed interests in latest technologies like Block Chain technologies**,** Speech Recongnization, App desinging, Google Cloud technologies. Her major fields of interest include Network security, Block chain and Crypto currecy, Machine Learning, Computer Vision, Image Processing, Data Visualization and has worked on several research projects and papers for the same.

Kakelli Anil Kumar is an Associate Professor of the School of Computer Science and Engineering at the Vellore Institute of Technology (VIT), Vellore, TN, India. He earned his Ph.D. in Computer Science and Engineering from Jawaharlal Nehru Technological University (JNTUH) Hyderabad in 2017, and graduated in 2009 and under-graduated in 2003 from the same university. He started his teaching career in 2004 and worked as an Assistant Professor, and Associate Professor, and HOD in various reputed institutions of India. His current research includes wireless sensor networks, the internet of things (IoT), cyber security and digital forensics, Malware analysis, Block-chain, and Crypto-currency. He has published over 40 research articles in reputed peer-reviewed international journals and conferences.

Dr. Aju. D received his PhD. in Computer Science and Engineering from VIT University, Vellore, India. He received his M.Tech. degree in Computer Science and IT from Manonmaniam Sundaranar University, Tirunelveli. He received his M.C.A degree from Madras University, India. Presently, he is working as Associate Professor at VIT University in the department of Computer Science and Engineering. He has published 30 research articles in different reputed international peer-reviewed journals. And, he has served as reviewer for few international peer-reviewed journals. He is having more than 15 years of teaching and research experience. Consecutively, he has received research awards from 2014 to 2019 for his outstanding contribution towards research and publication at VIT University. His area of interest includes Digital Image Processing, Medical Imaging, Computer Graphics, Multimedia Systems and Digital Forensics.

How to cite this paper: Yadavendra Atul Sakharkar, Mrinalini Singh, Kakelli Anil Kumar, Aju D, " A Reinforcement Learning-based Offload Decision Model (RL-OLD) for Vehicle Number Plate Detection ", International Journal of Engineering and Manufacturing (IJEM), Vol.11, No.6, pp. 11-18, 2021. DOI: 10.5815/ijem.2021.06.02