



# Autonomous Vehicular Surveillance using License Plate Recognition over Cloud Computing Architecture

Baby. D. Dayana, Bikramjeet Chanda, Vaidya Kabir Vinayak , Sheikh Fahim Raza

**Abstract:** License Plate Recognition (LPR) is the extraction and identification of licence plate numbers from license plates. The extraction process requires ample image pre-processing using normalization, gray scaling and edge removal techniques. These extracted plates can then be identified using image processing techniques such as neural networks and support vector machines. These license plates are captured using stationary video cameras, which extracts images from their feed as inputs into the image extraction algorithm. For the purposes of vehicular surveillance, these cameras are inefficient, as a lot of them will be required to monitor vehicles effectively. Hence there is a need for a larger scale model to carry out effective vehicular surveillance. For this purpose, the cameras embedded in self driving cars are utilised as replacements to stationary video cameras. These cameras have to advantage of being constantly mobile, hence being able to carry out a larger scale of surveillance. These cameras capture meaningful images of license plates from their video feed, and upload these images to the cloud using a Vehicular Cloud Computing (VCC) architecture. This centralized cloud carries out the image extraction and image processing tasks. The identified license plates can be used to monitor the cars they belong to. The cloud compares them to a database of license plates that are flagged by law enforcement. If the license plate is found to be flagged, then the respective law enforcement authorities are notified of the location of the car. If the plate belongs to a car with a history of misbehavior, the car capturing the plate is informed of this, making it easier to safely navigate around the problematic driver.

**Keywords:** License Plate Recognition, Self Driving Cars, Vehicular Cloud Computing, Support Vector Machine.

## I. INTRODUCTION

While there are multiple pre existing systems that ensure that a miscreant behind the wheels will be apprehended for his/her actions there aren't any that do so while also attempting to warn the other vehicles in the vicinity. A result of which might be collateral damage.

Revised Manuscript Received on October 30, 2019.

\* Correspondence Author

**Baby D Dayana\***, Assistant Professor, Department of Computer Science and Engineering, Government College of Engineering, Tirunelveli, (Tamil Nadu), India.

**Bikramjeet Chanda**, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, (Tamil Nadu), India.

**Vaidya Kabir Vinayak**, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, (Tamil Nadu), India.

**Sheikh Fahim Raza**, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, (Tamil Nadu), India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

To prevent this possibility of collateral damage a system must be devised that manages to inform the other vehicles to take certain precautions while around the aforementioned hazardous vehicle. License Plate Recognition (LPR) is a system where license plate numbers are extracted from video streams or still images of license plates, or vehicles carrying license plates. This technology has numerous security applications in various areas, such as parking lots, apartment complexes, and most importantly, open roads. Often times, LPR is implemented using still cameras, due to restrictions imposed by the diversity of license plates, natural conditions such as fog, rain and unfavourable lighting [1], and distances between vehicles and the camera [2]. LPR operations usually take place in four steps. First, an image of a vehicle is extracted from the video stream. The location of the license plate within this image is found using edge detection algorithms [3]. The next step is to pre-process the image to make it easy to analyse. Tilt correction, greyscaling, and light correction are some of the most commonly used methods for this purpose. After pre-processing, character segmentation is done to separate characters from each other. Character recognition is the next phase, where the features of each character are extracted from the various segments. These features are fed into a Support Vector Machine (SVM) to identify the license plate numbers. Using LPR technology to procure the license plates can allow for one to take multiple measures against atrocious drivers. The system allows for one to access the information of other vehicle's license plates in order to prevent any possible accidents from occurring. As an addendum to this model, one can allow for law enforcement to flag certain vehicles. Using the LPR from other cars they can procure the location of the car sending the message and also the location of the flagged car by proxy. Standard vehicle surveillance systems make use of steady cams at well places roadside locations. These systems offer a good success rate of detecting license plates. However, with the rapid growth of the self-driving car industry, such systems might turn obsolete. Self-driving make use of images, captured by a series of front cameras, to make pathing decisions [11]. These cameras, compiled with the computational capabilities of the car itself, are more than effective at navigation based solely on images. All self-driving Cars on the streets can be connected to a centralized cloud architecture. The cloud possesses greater computing capabilities than any one car itself. Hence, it can utilise this architecture to perform license plate detection as the car is running on the streets. Self-driving cars handle traffic navigation solely on a mathematical basis, taking into account relative speeds and time-to-impact.

However, it can safely be said that human intervention is the leading cause of all on-road accidents. Self-driving cars must be taught to consider the human factor while navigating in populated areas. The following project proposes a simple way of doing this, informing the car of the driver's prior accidents based on license plate information. If the car is informed of the fact that the neighboring car is driven by someone with a history of on-road misconduct, it can maintain a safe distance, to avoid it's passengers from erratic human behaviour. Law enforcement has also been known to take keen interest in license plates, as they can be used to track down miscreants on the run. This process can be aided by self-driving cars on the road by identifying a car that's being tracked, it can report its location to the respective authorities.

## II. RELEVANT WORKS

With modern vehicles being capable of computing and storage operations, the Internet of Vehicles didn't take long to evolve as the potential future of smart cities. Yingying et.al. [8] proposed a system of making use of idle parked vehicles to implement vehicular fog, as a means of sharing computing resources with nodes in a vehicular network. This approach however is not suitable for surveillance purposes, as surveillance is most required on roads to help prevent live accidents. As proposed here, having moving vehicles carry out surveillance would be of far greater benefit. License Plate Recognition (LPR) is the most straightforward technique to implement vehicular surveillance, and a number of algorithms and techniques have been proposed to implement LPR systems. Chiun-Mu et.al. [7] proposed a simple Convolutional Neural Networks based approach to LPR implementation. However, they have failed to mention the steps leading up to character recognition, i.e., image pre-processing, which is a vital step in the LPR process. Without proper image pre-processing, the effectiveness of the classifier algorithm reduces significantly. LPR operations normally occur on machines that possess a high computational power. A number of proposed systems speculate on how a cluster of modern vehicles can act as edge devices on a fog system [4] however due to very obvious network bandwidth limitations this model is hypotecal at best. The closest alternative to the aforementioned system is to directly use cloud computing. built-in cameras from modern vehicles, such as dash cams and rear-view cameras can help procure and provide images for LPR functionality. Smart vehicles can not account for the erratic behaviour of a human driven car, taking a "time to contact" based approach [5] while calculating relative speed or velocity to prevent any collisions. Another possible implementation would be to provide smart vehicles with information pertaining to bad drivers (through LDR and cloud computing) in the current vicinity allowing for it take measures prior to an impending collisions. A system must be devised to access the clouds since a number of vehicles on the road are expected to ping the server farm at the same time. A concept that can be used to breakdown the permission seeking process of the cars in the system is called game theory [4]. According to game theory, the cars in the system will attempt to perpetually send a packet to the cloud regardless of the other cars. This poses a problem for the system since a number of cars attempting to access it at once will result in a buffering

period. Using non cooperative game theory, the amount of time taken for a particular packet to reach the cloud from a node(car) can be calculated since the nash equilibrium for the system can be derived prior to the deployment of the system.

## III. ARCHITECTURE OF THE MODEL

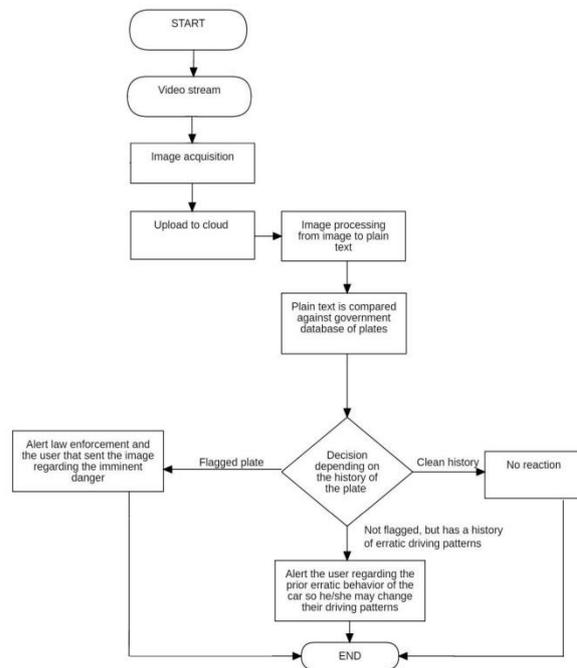


Fig. 1. Architecture Flow Diagram

The base architecture of the system accounts for a continuous video stream to exist in perpetuity. A frame off the said stream is parsed out at a periodic rate of ten seconds, said frame is then sent to the nearest server farm for processing where the car plate number is detected; assuming a plate does exist in the image procured. Edge detection algorithms are used to detect license plates, and image preprocessing methods like normalization, binarization and angle correction are implemented to increase the success rate of recognition algorithms. The detected images are read using a CNN-SVM system, The detected images are fed into a Convolutional Neural Network (CNN), which is pre-trained with standard license plate images. The CNN then performs feature extraction, where interesting features are extracted from the image. These images are the input to the Support Vector Machine (SVM) classifier, which identifies license plate numbers from the extracted features. This combined CNN-SVM system produces the best success rates of about 93%. A pre existing database [9] is used as a reference for vehicles that have been blacklisted as notorious or hazardous, if the vehicles vin number is relatively clean then there are no actions taken and the data is thrown out. However if there is a scenario where the vin number shows there is an abundance of outstanding cases of misconduct and road rash on the scanned vehicle then the driver that send the image is notified along with any other vehicles in the vicinity equipped to receive that information.

Along with notifying the civilian that sent the image, the law enforcement agencies present in the area will also be notified in the scenario where the aforementioned criteria is satisfied but is also coupled a flag placed on the number by any law enforcement body.

#### IV. CAR LICENSE PLATE EXTRACTION

##### A. Overview Of Extraction Algorithm

In the image preprocessing phase of existing License Plate Recognition (LPR) algorithms, canny-edge algorithm is often used to filter the license plate from the image. This method, however, sometimes causes images that are not license plates to be detected as such. This reduces the success rate of the algorithm. Hence in this project, we use a combination of Canny-Edge Algorithm and Laplacian Algorithm, similar to the algorithm used by Chi-Sung et. al. [3]. The License Plate image is captured from the vehicle at a rate of 1 image per 10 seconds. This image is then sent via a cloud-based architecture, to the centralized cloud, where plate detection and recognition takes place.

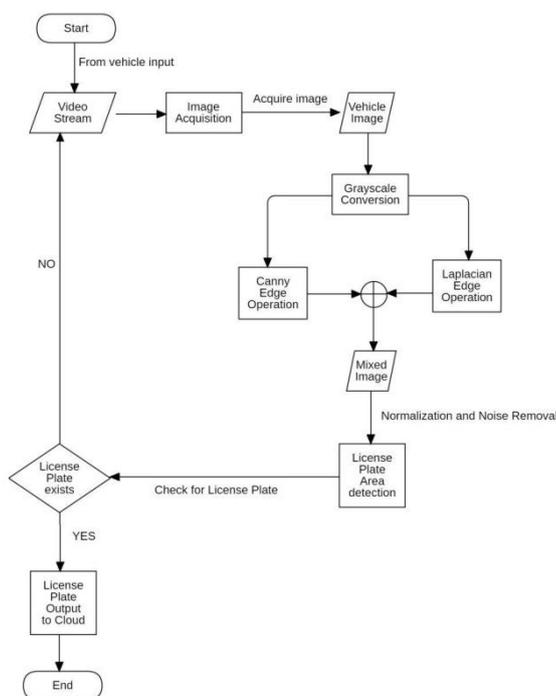


Fig. 1. Flowchart of License Plate Extraction

##### B. Flowchart Of Algorithm

The vehicle image initially undergoes gray scale conversion, to make the image easier to read by the system. The license plate in the image is detected using a combination of two edge operations, namely Canny-Edge and Laplacian. The products of these operations are OR'd together to produce a mixed image. The final mixed image then undergoes normalization and noise removal process, where operations like binarization, angle-correction etc. are applied to make the license plate easier for reading by the Neural Networks. Before forwarding the image to the inner layers of the neural networks for recognition, we analyse the image to check the success of the recognition algorithm. If it is unsuccessful, it can be deduced that either the algorithm failed to identify a license plate, or a license plate

does not exist in the input image. In either case, the image is discarded and a new image is processed from the vehicle.

#### V. TRANSMITTING TO THE CLOUD

Transmitting raw data to the cloud from the self-driving vehicle occurs with the help of game theory[4]. To be more precise, non-cooperative game theory. Since the individual vehicles are attempting to reach the cloud on a cloud computing architecture, they all may attempt to transmit simultaneously. This however, while not ideal, won't be a massive hindrance since the buffer rate can be calculated and compensated for using the nash equilibrium constants [10].

Every vehicle attempting to transmit data, does so over a simple internet connection which presumably works over an Intermediate System to Intermediate System network, the most common protocol for larger public networks.

Thus every vehicle must transmit the frame of the video in the form of a packet to an L1 router[12] which is subsequently transmitted to an L2 router which feeds into the server farm that is tasked with processing the given raw data. The cloud and the server farm are a singular entity.

#### VI. READING LICENSE PLATES

Convolutional Neural Networks (CNN) are deep learning algorithms that are primarily used for image processing. They are capable of taking image inputs and assigning importance, in the form of weights and biases to classify them. The amount of image pre-processing is relatively less in CNN based classifiers, which means they require less amount of processing power. However, solely using CNN classifiers doesn't provide the necessary level of success. Hence, a CNN-SVM classifier is used, similar to the classifier used in [16].

The CNN is trained using sample images of cars with their license plates. The CNN implements the Car License Plate Extraction Module, where the first layer learns to perform edge correction on the given dataset of sample images. The inner layers of the CNN first analyze if a license plate exists in the image. If it does, then the CNN extracts features from the image that it deems important for reading license plates. These image features serve as inputs for the Support Vector Machine. A multiclass SVM is trained using these training features with known plate numbers. This CNN-SVM combination classifier is then applied to the images captured by front cameras of self-driving cars.

#### VII. FLAGGING BACKLISTED PLATES

The LDR passes the processed image to the flagging component of the server once it has been preprocessed. The plaintext version of the plate or VIN number is cross referenced with that of [9] pre-existing databases.

This database already possess the prior records of the car and can be used to judge whether it is being driven by a competent individual or if it hazardous.

Since the plates are already meticulously recorded, they will have a tag saying whether they are stolen or have committed instances of road rash that require law enforcement attention. If it is deemed as such then a warning to the local law enforcement agencies is issued.

It may also deem the car as hazardous but not worthy of being reported to the authorities. This process happens in

The classifier algorithm can be further refined to allow for faster and more effective License Plate Recognition.

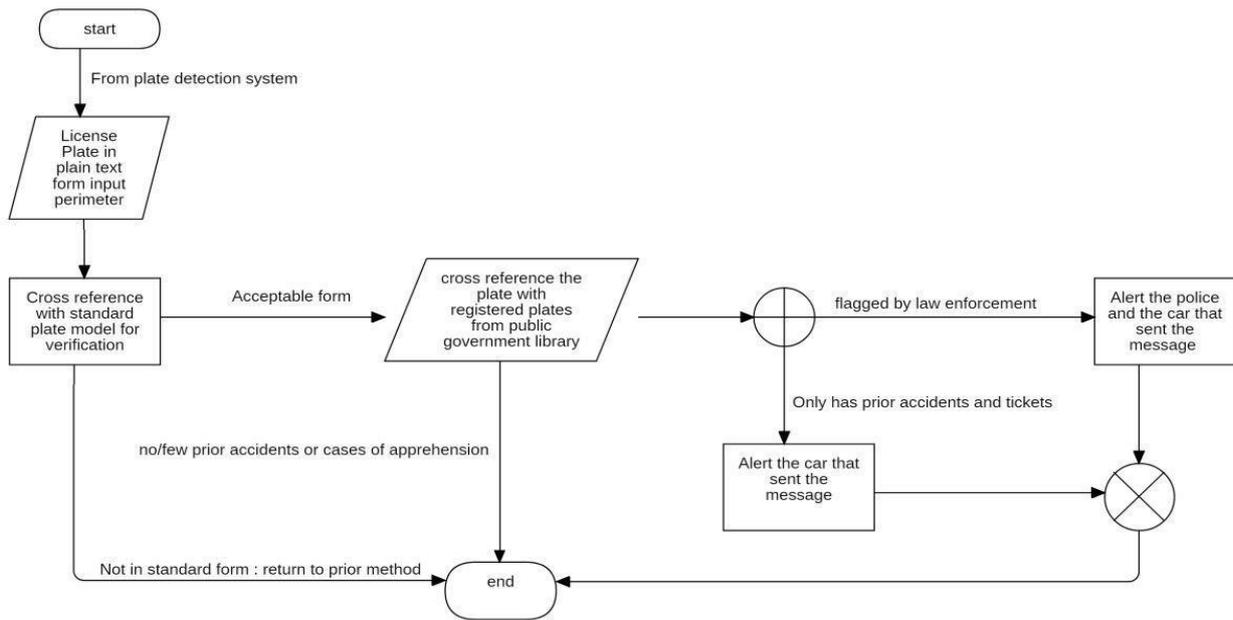


Fig. 3. Flowchart of License Plate Cross-Reference

perpetuity with then formation coming in from the other modules , assuming that a plate number can be pulled

## VIII. CONCLUSION

The system proposed is trying to push the limits of smart vehicular surveillance, by using the on board cameras in self-driving cars to implement License Plate Recognition (LPR). A combination of Canny Edge and Laplacian edge algorithms are used to extract a license plate from the images taken by the front and rear camera of a self-driving car. A Convolutional Neural Network (CNN) and Support Vector Machine (SVM) based model is used to identify license plate numbers from an input of license plate images. This CNN-SVM model

The backbone of the proposed system is cloud architecture based, connecting every vehicle to a centralized cloud to share computing and storage resources.

The given system is effective and implementable but relies on a number of constant variables such as the smart vehicles ability to offload excess data onto a cloud, while it is a simple procedure it does not have the current required infrastructure beyond the ability of the car to unload gather the information required.

While the idea is far more lucrative than that of stationary surveillance modules at street corners it requires the advent of cloud computing to become far reaching and easily accessible for the vehicles.

Hence, while it is hypothetical to utilise this on a large scale it should become implementable once the large scale adoptions of 5g [17] has taken place, making this a viable model for vehicular surveillance.

## FUTURE WORKS

Detection logic can be implemented in the self-driving cars, such that they only send images that have plates in them to the cloud. This would reduce the amount of processing necessary in the cloud servers.

Load sharing and resource planning algorithms can be implemented at the nodal level, to enable smoother access to the centralised cloud.

Cloud architecture can be expanded to allow for more computation and storage resources.

Edge computing or Fog computing methods can be implemented to reduce the stress and dependence on a singular cloud or computing unit.

## REFERENCES

1. F. Wang, L. C. Man, B. Wang, Y. Xiao, W. Pan, and X. Lu, "Fuzzy-based algorithm for color recognition of license plates," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 1007–1020, May 2008.
2. T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust license-plate recognition method for passing vehicles under outside environment," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2309–2319, Nov. 2000.
3. Ahn, C.-S., Lee, B.-G., Yang, S.-S., & Park, S.-C. (2017). "Design of car license plate area detection algorithm for enhanced recognition plate." 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT).
4. LiWang, M., Wang, J., Gao, Z., Du, X., & Guizani, M. (2019). Game Theory based Opportunistic Computation Offloading in Cloud-Enabled IoV." *IEEE Access*, 1–1. doi:10.1109/access.2019.2897617
5. Wang, L., & Horn, B. K. P. (2017). Time-to-Contact control for safety and reliability of self-driving cars. 2017 International Smart Cities Conference (ISC2). doi:10.1109/isc2.2017.8090789
6. Neto, A. J. V., Zhao, Z., Rodrigues, J. J. P. C., Camboim, H. B., & Braun, T. (2018). Fog-Based Crime-Assistance in Smart IoT Transportation System. *IEEE Access*, 6, 11101–11111.
7. Chuin-Mu Wang, & Jian-Hong Liu. (2015). License plate recognition system. 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)



8. Yao, Y., Chang, X., Mistic, J., & Mistic, V. B. (2018). Reliable and Secure Vehicular Fog Service Provision. IEEE Internet of Things Journal, 1–1.
9. <https://vincheck.info/free-license-plate-lookup/>
10. <http://www.columbia.edu/~rs328/NashEquilibrium.pdf>
11. Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, Karol Zieba, "End to End Learning for Self-Driving Cars"
12. [https://www.cisco.com/en/US/tech/tk365/technologies\\_white\\_paper09186a00800a3e6f.shtml#wp38640](https://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a00800a3e6f.shtml#wp38640)
13. MATLAB, Computer Vision System Toolbox, R2016a, MathWorks, <http://www.mathworks.com/help/vision/examples/image-categoryclassification-using-deep-learning.html>, 2016.
14. Alex Krizhevsky, Ilya Sutskever, and Geoffrey. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012: Neural Information Processing Systems, 2012
15. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
16. Yoshihiro Shima, "Extraction of Number Plate Images Based on Image Category Classification Using Deep Learning", 2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS2016) 17-20 December 2016
17. Gupta, A., & Jha, R. K. (2015). A Survey of 5G Network: Architecture and Emerging Technologies. IEEE Access, 3, 1206–1232.

### AUTHORS PROFILE



**Baby D Dayana** received her M.E. in Computer Science and Engineering in Government College of Engineering, Tirunelveli. She is currently an Assistant Professor at SRM Institute of Science and Technology since 2017.



**Bikramjeet Chanda** is a student of Computer Science and Engineering, currently working on attaining his bachelors of technology at SRM Institute of Science and Technology, Ramapuram. His research interests include Cloud Computing and networking as well as classical algorithms.



**Vaidya Kabir Vinayak** is a student of Computer Science and Engineering, currently pursuing a bachelor's degree technology at SRM Institute of Science and Technology, Ramapuram, Chennai. His research interests include Artificial Intelligence and Image Processing.



**Sheikh Fahim Raza** is a student of Computer Science and Engineering, currently pursuing a bachelor's degree in technology at SRM Institute of Science and Technology, Ramapuram, Chennai. His research interests include Cloud Computing and Database Security.