Stolen car detection using image processing and Machine Learning

KANISHKHA L. D., POOJAH G., EDNA ELIZABETH N.
Department of Electronics and Communication engineering, SSN College of Engineering Chennai, INDIA

Abstract: In recent years, vehicular stealing has been steadily increasing around the world. Existing theft control methods to find stolen vehicles are inadequate in this current scenario. Thus, the motive of our project is to explore image processing and machine learning techniques to find an effective method to detect vehicles and extract registration numbers. We aim to capture frames only when there is a motion, then run the License Plate Recognition (LPR) algorithm, thereby reducing computational complexity. Once the license plate is detected by comparing it with the records of the stolen car, the owner of the car is intimated with the location of the marked vehicle using the Global Positioning System (GPS) of the cameras.

Keywords: Image processing, machine learning, LPR, neural network

Received: July 12, 2021. Revised: June 28, 2022. Accepted: August 19, 2022. Published: September 28, 2022.

1. Introduction

Surveillance of high-traffic areas is essential nowadays because of the rapid increase in the number of vehicles and, consequently, the number of stolen vehicles. Video processing techniques can be used for determining the density of the vehicles on the road, and analysis of the surveillance of the videos proves helpful in traffic management systems. Systems performing video analysis use high-performance CPUs and application-oriented circuitry due to the enormous amount of data to be processed in real-time.

Different techniques have been proposed to detect vehicles and track them on computers. One method detects vehicles based on the rear lamp and license plate with dedicated traffic surveillance cameras. The system uses a stationary camera to capture the traffic video, extract frames from the video sequences and work on each video frame. The parts are combined using Markov Random Field to model the relationship once the localization of the vehicle parts is completed.

Vehicle detection can be done on a model based on the Hybrid image template by extracting various features using 50–1000 frames. Vehicles are detected in three stages using the SUM-MAX procedure. In each step, the local frame region with the maximal score is computed. The results of this model demonstrate the detection of all vehicles under certain conditions.

Many systems have been developed to detect, track and count the vehicles on the road, which are expensive, limiting the number of systems to be used at different locations. Ambadas B. et al. proposed work for the extraction of vehicle number plate information from an image. Irrespective of the research that has taken place in this field, none proposed an efficient real-time mechanism. Hence, we proposed a system for extracting the information from vehicle number plates in real-time.

P.M. Daigavane et al. consider road situation analysis tasks for traffic control and ensuring safety. The following image processing algorithms are proposed: vehicle detection and counting algorithm, road marking detection algorithm. The algorithms are designed to process images obtained from a stationary camera. The developed vehicle detection and counting algorithm was implemented and tested for an embedded platform of smart cameras.

Ye Li et al. proposed a vehicle detection method based on a deformable hybrid image template. Their method contains two steps: constructing a hybrid image template and its probability model and detecting vehicles from traffic images using a three-stage SUM-MAX procedure. Small image patches constituting the hybrid image template are automatically learned from training images in the template construction step.

We have designed a new portable system which can change its capture rate at any given time and can access video data in real time through mobile or desktop computers from remote places. We introduce Raspberry Pi for detection and tracking of vehicles. It also analyses the input video and provides the vehicle count at any given time.

2. Methodology

2.1 Flowchart

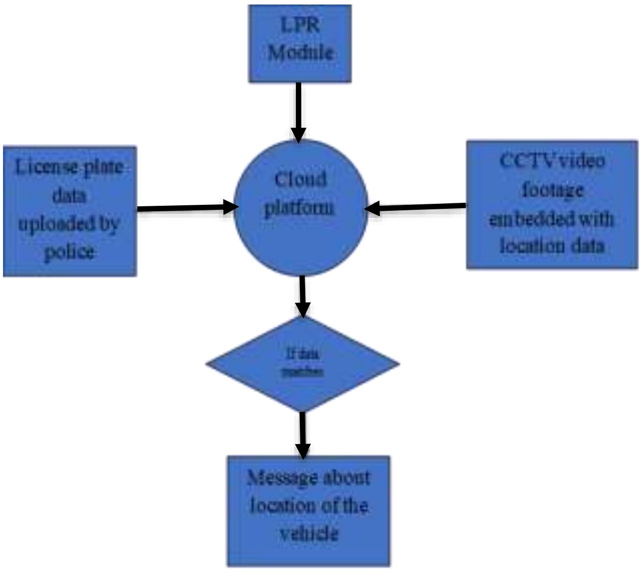

Fig 1. Flowchart of the methodology of implementation

The conventional ways of identifying license plates include algorithms such as You Only Look Once (YOLO) and morphological techniques followed by any conventional ML algorithm. In this project, Mask Region-based Convolutional Neural Network (RCNN) is used to carry out the task since it focuses on instance segmentation instead of semantic segmentation.

Figure 1 describes the implementation flowchart, wherein the license plate data, LPR module consisting of Nomeroff net and OCR to predict the license plate, and real-time CCTV video footage are integrated onto the cloud platform to locate stolen cars. Once a stolen car is identified, a message is sent to the police and the complainant about the vehicle's location.

2.2 Mask RCNN

RCNN is a type of machine learning model whose architecture is specifically designed for object detection. Its approach utilizes the concept of bounding boxes across the object, which is used for evaluating convolutional networks independently on all the Regions of Interest (ROI) to classify multiple regions.

RCNN was further improvised into Fast RCNN, an optimized form of RCNN built to enhance the computation speed. This advances by learning the attention mechanism with Region Proposal Network, a simple Neural Network that proposes multiple objects available within a particular range, and Fast RCNN architecture, which extracts features using ROI pooling and performs classification and regression.

The architecture of Mask RCNN was developed on top of Fast RCNN. The crux of this algorithm is dependent on the concept of image segmentation.

There are 2 types of image segmentation: Semantic segmentation and Instance segmentation.

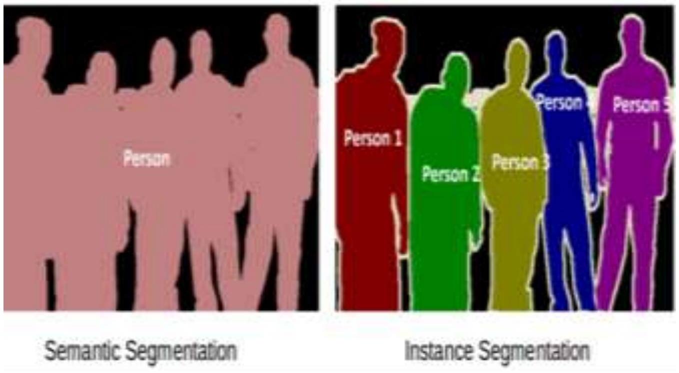

Fig 2. Semantic segmentation vs. Instance segmentation

As shown in Fig 2, semantic segmentation works on identifying similar objects as a single class from pixel level. Instance segmentation deals with the correct detection of all objects in an image while also precisely segmenting each instance. It is, therefore, a combination of object detection, object localization, and object classification.

The Mask R-CNN framework is built on top of Faster R-CNN. The algorithm is as follows:

Step 1: ResNet 101 architecture is used to extract feature maps.

Step 2: Feature maps are passed on to Region Proposal Network (RPN), which returns the candidate bounding boxes.

Step 3: ROI pooling layer is then used to bring uniformity in shape.

Step 4: For all the predicted regions, Intersection over Union (IoU) with the ground truth is computed. We consider it to be ROI if it crosses the threshold.

Step 5: A mask branch is added to the architecture, which produces a segmentation mask.

Step 6: Finally, the proposals are passed to a fully connected layer to classify and output the bounding boxes for objects.

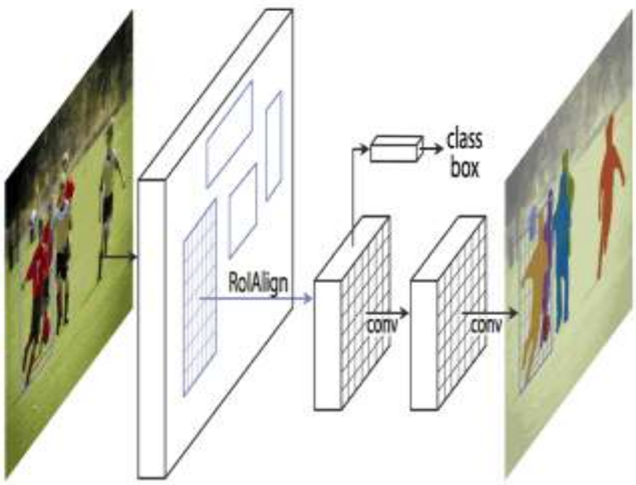

Fig 3. Architecture of Mask RCNN

2.3 Nomeroff-net

The State-of-the-art technology for LPR is the concept of Nomeroff-net. This open-source python LPR framework is based on applying the Mask RCNN architecture and customized Optical character recognition (OCR) module powered by Gated Recurrent Units (GRU) architecture, as shown in Fig 4. This returns the number of the license plate as a string. This architecture is at the initial development stages, and the performance can be maximized by training the model with self-dataset.

Figure 4 explains the architecture of the software implemented. The input image goes through the convolution and pooling layer twice and is reshaped. The reshaped image is passed on to a fully connected one through two Gated Recurrent Units (GRU), whose outputs are added elementwise. This is passed through two GRUs, and their results are concatenated and passed on to the fully connected layer and through the SoftMax activation function to get the final output.

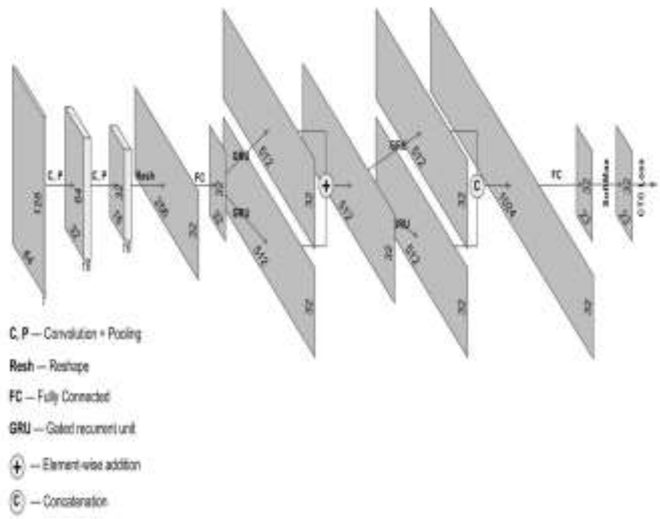

Fig 4 Nomeroff-net with GRU architecture

3. Implementation

3.1 Hardware Implementation

In the hardware setup shown in Fig 5, two cameras are connected to a Raspberry Pi 4B model. The video feed will be seen through a monitor. Motion eye OS is installed in the RPi, which triggers the camera to capture the frame only if it detects a motion.

These frames are fed into the cloud platform from where we import the files to perform the LPR algorithm. Two cameras are used to mimic the real-life scenario.

Fig 5. Hardware setup

3.2 Software Implementation

For the software implementation, as discussed in the previous section, we use an open-source python framework, Nomeroff-net, which uses the concept of Mask RCNN and a customized OCR module to detect the license plate. Once the license plate is detected and finds a match in the database containing the stolen vehicles' details, it sends a cellular message to the owner with the location where the car was detected.

4. Results and Conclusion

Using the aforementioned hardware and software setup, the results have been obtained and presented below.

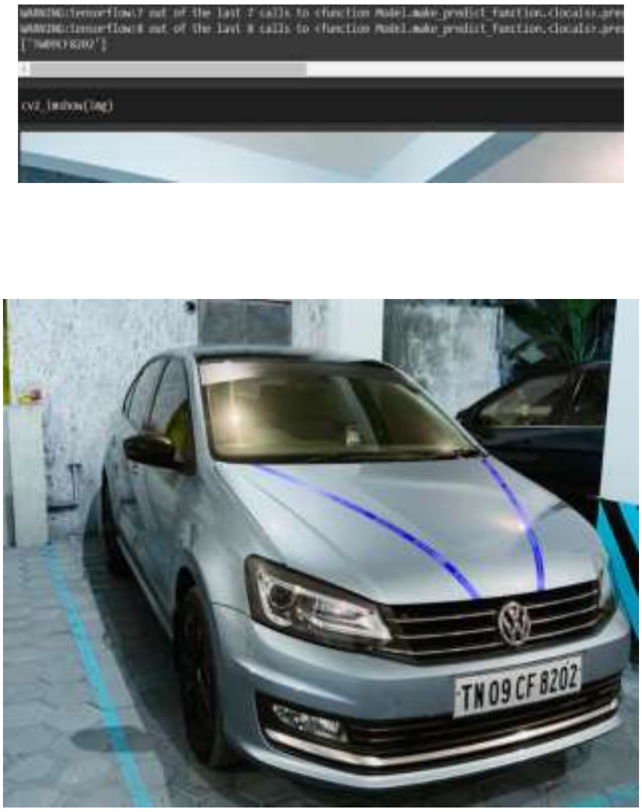

Fig 6. Image testing

Fig 6 is the test image that was given to the algorithm to detect the license plate.

Fig 7. Camera 1 Input images

Fig 8. Camera 2 Input images

Fig 7 shows the input images captured by camera 1 and Fig 8, that of camera 2 triggered by the motion eye OS. This was done to recreate the real-life scenario.

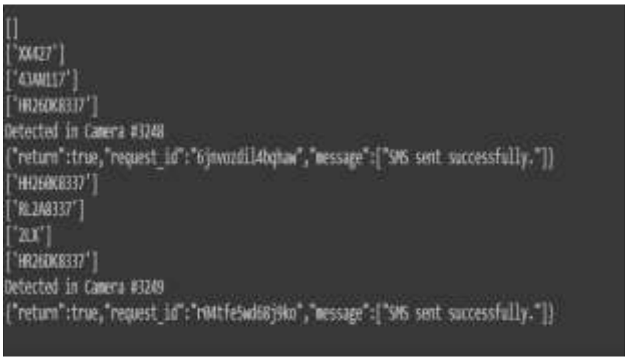

Fig 9. Output of LPR module

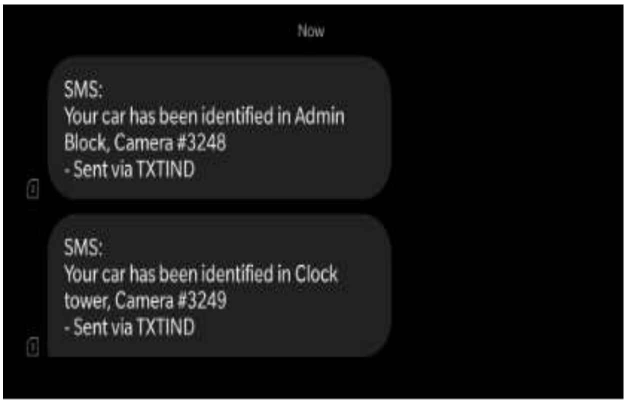

Fig 10. Message alert to the Customer/Police office

Fig 9 shows the output of the LPR module for every image. Once the value matches the record in the database, an alert message is triggered to the customer or the police station with the embedded location. The output of the message is shown in Fig 10.

In conclusion, cameras are being installed everywhere around the city to monitor traffic rule breakage and congestion. To make the installation more beneficial for the public, Machine learning and computer vision can be used to alert the nearest police station if a stolen car passes by the camera. In a common database, police will register the details of a stolen car, and the camera will detect the license plate of the ones it captures and compares it with the details of stolen cars in the common database; if the details match, it will send an alert to the nearest police station that a stolen car is passing by in the area.

This product focuses on locating stolen vehicles by implementing real-time detection of moving vehicles using surveillance camera footage. Current products available in the market have not implemented real-time detection and processing of vehicle data, which is the novelty of our project.

5. Acknowledgment

SSN College of Engineering has sponsored this work and supported its development from nascent stages to its completion.

References

[1] Bin Tian; Ye Li; Bo Li; Ding Wen, Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance, in: IEEE Transactions on Intelligent Transportation Systems, vol.15, no.2, pp. 597–606 (April 2014).

[2] Ye Li; Bo Li; Bin Tian; Qingming Yao, Vehicle detection based on the and– or graph for congested traffic conditions, in: IEEE Transactions on Intelligent Transportation Systems, vol.14, no.2, pp.984–993 (June 2013).

[3] M. Anandhalli, V.P. Baligar, Improvised approach using background subtraction for vehicle detection, in: Advance Computing Conference (IACC), 2015 IEEE International, pp. 303–308, 12–13 (June 2015)

[4] Ye Li; Bo Li; Bin Tian; Fenghua Zhu; Gang Xiong; Kunfeng Wang, Vehicle detection based on the deformable hybrid image template, in: 2013 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pp.114–118 (28–30 July 2013).

[5] Xinyi Zhou, Wei Gong, WenLong Fu, Fengtong Du, "Application of Deep Learning in Object Detection", International Journal of Innovative Research in Advanced Engineering (IJIRAE), June 2014. ISSN: 2349-2163

[6] Yong-Deuk Shin, Jae-Han Park, Ga-Ram Jang, Moon Hong Baeg, "Moving Objects Detection using Freely Moving Depth Sensing Camera", 21st International Conference on Pattern Recognition, November 2012.

[7] Ashwini Patil, Shobha Mondhe, Tejashri Ahire, Gayatri Sonar, "Auto-Theft Detection using Raspberry Pi and Android App", International Journal of Research in Engineering Application & Management (IJREAM), October 2016. ISSN: 2494-9150

[8] D. Pham, C. Xu, and J. Price, "A survey of current methods in medical image segmentation," Annual Review of Biomedical Engineering, vol. 2, 2000.

[9] G. Litjens, T. Kooi, B. E. Bejnordi et al., "A survey on deep learning in medical image analysis," Medical Image Analysis, vol. 42, pp. 60–88, 2017.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, Columbus, OH, USA, June 2014.

[11] Shu, Jian-Hua; Nian, Fu-Dong; Yu, Ming-Hui; Li, Xu (2020). An Improved Mask R-CNN Model for Multiorgan Segmentation. Mathematical Problems in Engineering, 2020(), 1–11. doi:10.1155/2020/8351725