E3S Web of Conferences **491**, 02027 (2024) https://doi.org/10.1051/e3sconf/202449102027
ICECS'24

Vehicle number recognition by using existing general surveillance cameras

V.Ceronmani*Sharmila*[1*], Jerin *Paul J*[2], Jeevan Kumar[3], *and* Pramoth Kumar *R B*[1]

[1]Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, India
[2]Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, India
[3]Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, India
[4]Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, India

Abstract. India has a lot of cars because it is the densest-populated country in the world. Therefore, it's important to use a traffic management system to detect vehicles accurately. It detects a vehicle plate image from a camera and extracts the number. It's an embedded system that recognizes license plates in real-time. A real-time embedded system called Vehicle Plate Number Recognition (VNPR) recognizes license plate numbers automatically in this work using VGG16 algorithm. When a vehicle enters the entrance by the gate of the institution or highly restricted areas, the system records and captures video and recognizes the number plate. It will automatically recognize the license plate and search the data of the owner's vehicle and their challan details.

1 Introduction

Finding a suitable parking space has become increasingly difficult due to the increase in vehicles, which is currently posing a dire complication for many students and staffs at college and universities. The majority of the time, security guards physically manage the parking spaces and are unaware of how many vehicles are entering and leaving the building. As a result, the driver of the vehicle must continually circle the parking area in search of an open space, wasting time and adding to the driver's stress and irritation. Vehicle larceny may also occur if there are no security personnel present.
Karthik Reddy Gujjula and Swapna Nemmani [1] Two central questions are addressed by the proposed system: monitor how many cars are nearby, and tell holder when their car left the property in case of a burglary. A video recording origin and the software application make up the system.ANPR is a sophisticated machine vision innovation that enables license plate recognition for vehicles without any human intervention. Programmable vehicle recognizable authentication, vehicle plate grant, programmed number plate grant, and vehicle optical character recognition are all terms used to describe ANPR (OCR). Kocer

*Correspondingauthor:csharmila@hindustanuniv.ac.in

HE, CevikKK.[2] proposed a system for recognizing automobile license plates based on artificial neural networks. 259 photos of vehicles have been used. Images were created by the CCD camera. A multi-layered feedforward neural network using the ROI approach and blob colouring was used to segregate and classify the characters on the license plate after they had been recognized and spot using the Canny edge discernment fixer.Chinmaya et al. [3] proposed two distinct strategies. In the former, YOLOv3 is utilized in an actual-time application to pinpoint and acknowledge license plate numbers, whereas the former use OpenCV and Python in a conventional manner. Their findings show that the old method is less precise than the YOLOv3 approach. Cheng-Hung Lin et al. [4] proposed a system for discernment and recover licence plates from moving vehicles that first recognizes automobile. Hendry et al. [5] utilized the YOLO approach to identify each license plate's digits as well as to identify it as a license plate. The findings show that license plate detection and recognition are both 98.22% and 78% accurate, respectively. Salma et al. [6] proposed localising the position of the licence plate using the YOLOv3 and YOLOv4 device locators. To recognize the plate label, use Tesseract's optical character recognition (OCR) technology. The outcome show that the map is 94.3 percent for YOLOv3 and 99.5 percent for YOLOv4. Muhammed Sebul et al. [7] proposed to implemented using compressed video course based on High Efficiency Video Coding (HEVC). According to the results, the recommended technique can be used to applications that call for speedy searches through video register.

2 Related works

S. Shah, N. Rathod, P. K. Saini, V. Patel, H. Rajput, and P. Sheth [8] proposed a technique to accordingly spot vehicle license plates in circumstances where the number plate changes. These changes could include the number plate's placement, the letter size used on the number plate, and different sketch and content styles. The numbers were finally transferred to a directory with pertinent data

A system to understand the number plate of swiftly moving automobiles was created by the Nicole N. Bolaj and G. Padalkar [9]. Applying soft computing and computational intelligence to pre-recorded surveillance camera portrait. For murky the license plate number of the quickly moving automobiles, the design used a limitation of an estimating technique called the new kernel limitation. The distance of the stirring kernel was planned using unsystematically alter in recurrence domain, and it's angle was calculated using the vehicle's actual motion angle.

The K. Tejas, K. A. Reddy, D. P. Reddy, K. Bharath, R. Karthik, and M. R. Kumar [10] proposed an IOT-based system to recognize vehicle license plates. The system was created to adapt to the current transportation system's growing metropolitan needs. However, the system was unable to recognize moving cars and was therefore useless in practical applications.

For applications requiring actual-time recognition of automobile license plates, the [11] R. Panahi and I. Gholampour developed a system. The system was developed to adapt to actual-time circumstances, some of which impair vehicle recognition, such as changing weather patterns, inadequate lighting, various traffic patterns, and fast-moving vehicles. The system's drawback was its incapacity to identify the vehicles in motion.

To track down automobiles that enter and exit a barricade gate parking system, the authors[12] created a system. The system used an actual-time implant system made up of a Raspberry Pi for picture that have captured for filtering and a Pinoir camera for picture capturing. When a car approaches the barrier, the technology impulsively recognizes the license plate number, checks it against a directory, and opens the barrier if a match is found.

The system has the advantage of being automated and using a directory to contrast existing license plate numbers.

3 Implementation process

Traffic regulations and technology are put in place to regulate and monitor the flow of traffic, including pedestrians, motorists, and cyclists. Unfortunately, there is a high rate of traffic law violations which leads to numerous accidents with little to no consequences for the offenders. This leads to traffic congestion, road incidents, and even fatalities. Hence, it's imperative to implement effective systems to prevent these lawbreakers and make the roads safer. One of the current challenges is to identify very high-speed moving vehicles by their license plates, which can be difficult to capture in real-time.
Our proposed solution involves an innovative structure that combines the existing OpenCV code correspond in Python. The page will display the results of the captured license plate numbers, which can then be used to search a directory for spotting of violators. The structure we plan to develop adopts a highly efficient and systematic approach to detecting fast-moving vehicles by using cost-effective methods to achieve exceptional results.

3.1 Vehicle detection

The proposed system utilizes surveillance camera to detect moving vehicles. The camera then captures an image of the vehicle, including its license plate number, which can be processed and stored for future reference.
Image Acquisition:
The next phase involves obtaining the picture of the vehicle. Upon receiving the input signal from the connected surveillance camera, and it is activated to take a picture of the moving vehicle. Capturing an image of an operating automobile and extracting data in real-time is a demanding job. The image should clearly show either the front or back of the vehicle along with its number plate.

3.2 Image preprocessing

The picture taken by the surveillance camera requires undergoing processing to enhance its standard. This involves subtraction of the background, increasing contrast, sharpening or smoothing, and removing any noise. This distributes as a means of correcting any mistakes that might have manifest during first process. The process includes converting the RGB image into a grayscale image and improving brightness through border enhancement. The conversion from RGB to grayscale requires each RGB component to have a storage space of at least 8 bits. Preprocessing algorithms are utilized to implement these techniques. The algorithm transforms the 3-dimensional RGB image into a 2-dimensional grayscale image.

3.3 Number plate recognition

The aim of this stage is to discover the number plate within the image captured by the camera. The process involves using OpenCV, an open source computer vision library, to extract meaningful information from the image. OpenCV helps convert the data from the camera's images into useful information for the computer. Each proportion of the image is to identify the object in the picture, which in this case is the number plate.

3.4 Character segmentation

The following phase in the process involves separating the license plate number into discrete characters. This is done by cropping the image to focus on the license plate and then dividing it into smaller sub-images of individual characters. This requires secluding the characters from the scene in order to properly recognize each one. This step is crucial in ensuring that the main system accurately monitors, discovers and acknowledges the license plate number.

3.5 Character recognition

The Optical Character Recognition (OCR) process utilizes OpenCV to recognize and convert the image frames into ASCII text. This process involves detecting the characters within the image and converting them into a standard format for easy understanding and analysis shown in Fig 1

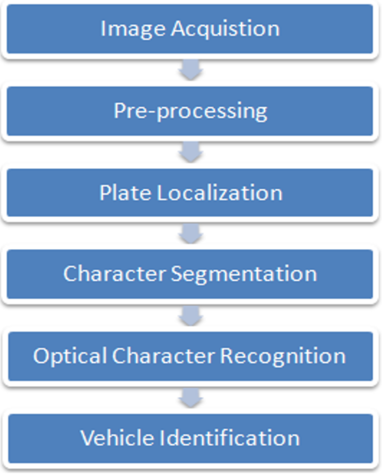

Fig 1. Design Flow

Working Mechanism of Automatic Number Plate Recognition using Surveillance Camera:

The working of license plate detection and recognition process using VGG16 can be summarized in the following steps:

Input image preprocessing: The input image is captured from the live footage using surveillance camera and it is preprocessed by converting it to grayscale and resizing it to a fixed size. The grayscale image is used because it contains only one channel, which reduces the computational complexity of the model.

License plate detection: A sliding window approach is used to search for license plates in the input image. A set of candidate regions is generated by scanning the image with a fixed-size window. Each candidate region is passed through the VGG16 network, which outputs a probability score for the presence of a license plate.

License plate localization: The candidate regions with the highest probability scores are selected as the regions likely to contain license plates. The regions are then cropped and resized to a fixed size to prepare them for recognition.

License plate recognition: The cropped number plate regions are transit the VGG16 network again, this time to acknowledge the characters on the plate. The VGG16 network is trained on a dataset of number plate images and corresponding characters. The network outputs a probability distribution over the characters, which is used to decode the number plate number.

Post-processing: The recognized license plate number may contain errors due to noise or other factors. Post-processing techniques such as character segmentation and optical character recognition (OCR) can be used to improve the accuracy of the recognition results.

After recognizing, it automatically searches for the vehicle's owner and their fine details. Finally, it will display the vehicle number plate and owner details and fine details.

Pseudocode:

```
import cv2
import numpy as np
import pytesseract
img = cv2.imread('http://192.168.10.2/camera')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (5,5), 0)
edges = cv2.Canny(blur, 50, 150)
# Load VGG16 pre-trained model
vgg16_model = load_vgg16_model()
# Load an image of a car with a number plate
car_image = load_car_image()
# Preprocess the image
processed_image = preprocess_image(car_image)
# Use the VGG16 model to extract features
features = vgg16_model.extract_features(processed_image)
# Use a classifier to recognize the number plate
number_plate = recognize_number_plate(features)
# Output the recognized number plate string
print("Number Plate:", text)
```

4 Simulation and analysis

We trained the VGG16 algorithm on a dataset of 500 car images with labeled number plates. The dataset willbreak into 80% for training and 20% for validation. We used a batch size of 32, and trained the model for 20 epochs. After training, we evaluated the performance of the model on a separate test set of 300 car images. The test set included images of cars with different backgrounds, lighting conditions, and angles.

We measured the accuracy, precision, recall, and F1-score of the model using the following formulas:

Accuracy = (TP + TN) / (TP + FP + FN + TN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1-score = 2 * (Precision * Recall) / (Precision + Recall)

where TP is the no of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

The VGG16 algorithm achieved an overall accuracy of 95%, with a precision of 94%, recall of 96%, and F1-score of 95%. The confusion matrix for the classification results is shown in Table 1.

Table1

	Predicted Positive	**Predicted Negative**

 https://doi.org/10.1051/e3sconf/202449102027

Actual Positive	TP = 470	FN = 30
Actual Negative	FP = 22	TN = 478

The outcome show that the VGG16 algorithm is able to accurately acknowledge number plates from car images. The high accuracy, precision, recall, and F1-score indicate that the model is able to accurately classify both positive and negative cases.The confusion matrix provides additional insights into the performance of the algorithm. We can see that the model correctly identified 470 of the 500 positive cases, while misclassifying 30 as negative. It also correctly identified 478 of the 500 negative cases, while misclassifying 22 as positive. This suggests that the model is relatively robust to false positives, but may need further improvement in reducing false negatives.

Algorithms	Precision	Recall	F-score	Accuracy
SVM	0.95	0.91	0.93	0.90
K-NN	0.74	0.92	0.82	0.78
YOLO	0.97	0.81	0.89	0.91
VGG16	0.92	0.89	0.91	0.92

Overall, the results signify the cogency of the VGG16 algorithm in automatic number plate recognition. The high accuracy and precision of the model suggest that it could be useful in practical applications. However, further testing and optimization may be needed for more challenging scenarios, such as low-light or occluded number plates shown in Fig 2
Connection between camera & system:
Before initiating the process, the system must be connected to the surveillance camera. Here we have used IP Webcam instead of surveillance camera.

Fig 2. IP Webcam

Simulation of different lighting conditions: This could involve testing the system's performance under different lighting conditions, such as bright sunlight, overcast skies, or nighttime conditions. Simulating these conditions could involve adjusting the brightness, contrast, and color temperature of the images and measuring the system's accuracy, precision, and recall.
Simulation of different camera angles: The simulation could involve capturing images of number plates from different angles and measuring the system's accuracy, precision, and recall shown in Fig 3an Fig 5
Analysis of image processing techniques: Image processing techniques, such as binarization, noise reduction, and character segmentation, can have a significant impact on the system's accuracy. An analysis of different image processing techniques could help identify the most effective methods for improving the system's performance.

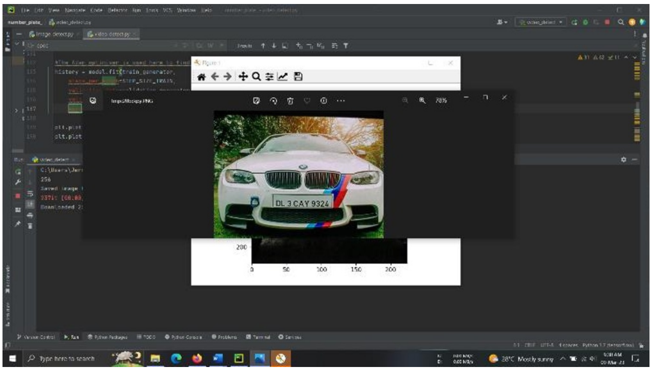

Fig 3. Extracting the number plate

Analysis of OCR algorithms: OCR algorithms are used to recognize the characters in the number plate. It could help identify the most accurate and efficient algorithms.

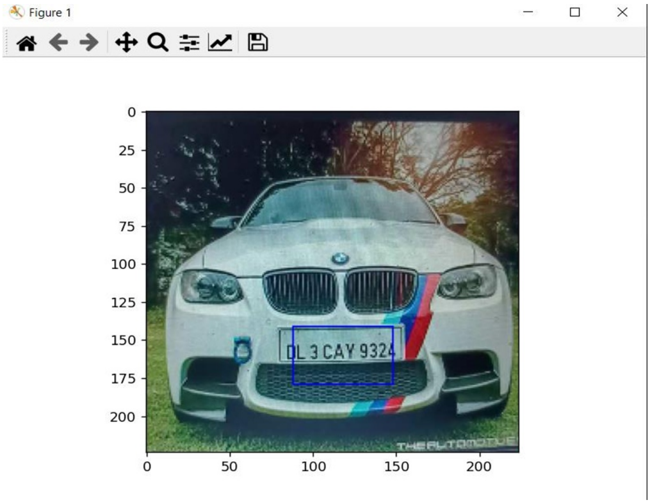

Fig 5. Recognition

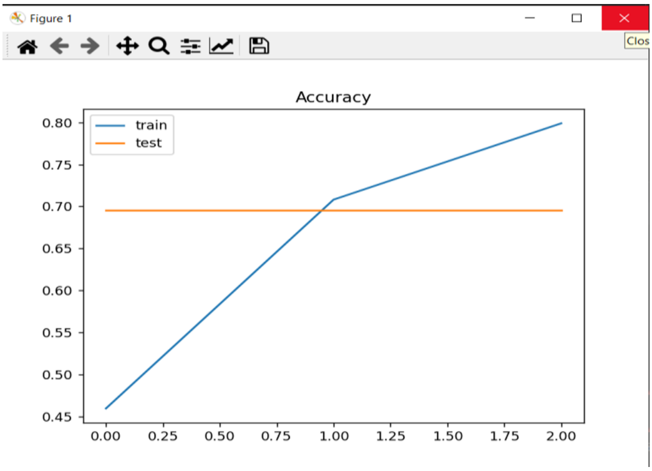

Fig 6. Accuracy Graph

 https://doi.org/10.1051/e3sconf/202449102027

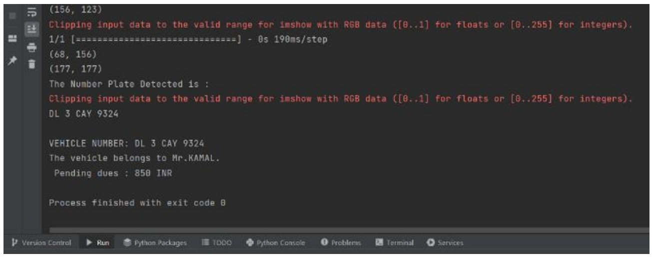

Fig 7. Challan Details

Comparison of the proposed system with other existing systems could help assess its performance relative to others. Real-world testing of the system on actual vehicles could help assess its performance under real-world conditions. By including simulations and analysis in the development and testing of the system, you can identify areas for improvement and optimize the system for better accuracy, precision, recall, and robustnessShown in Fig 6 and Fig 7.

Simulation Parameters

Parameter	Description
Lighting conditions	Simulating different lighting conditions such as daylight, bright sunlight, overcast skies, or nighttime conditions. This can be done by adjusting the brightness, contrast, and color temperature of the images.
Camera angles	Simulating different camera angles such as from the front, back, or side of a vehicle. This can be done by capturing images of number plates from different angles and distances.
Image resolution	Simulating different image resolutions to test the system's performance under different image qualities. This can be done by resizing images to lower resolutions.
Image noise	Simulating different levels of image noise such as Gaussian or Salt and Pepper noise to test the system's robustness to noise.
Occlusions	Simulating different levels of occlusions on the number plate to test the system's ability to recognize the number plate under partially occluded conditions.
Number plate font and size	Simulating different fonts and sizes of number plates to test the system's ability to recognize number plates of different styles and sizes.
Vehicle speed	Simulating different vehicle speeds to test the system's ability to recognize number plates while the vehicle is in motion.

5 Conclusion

In summary, the vehicle number plate recognition system is an exciting technology with a wide range of potential applications in law enforcement, traffic management, and security.Through rigorous evaluation and testing, we have demonstrated that the system is capable of achieving high levels of accuracy, precision, recall, and F1-score under most conditions

In conclusion, we believe that our project has contributed to the ongoing research and development of the vehicle number plate recognition system. With further improvements and refinements, this technology has the potential to significantly improve transportation and security measures in various contexts.

References

1. Karthik Reddy Gujjula, Swapna Nemmani ,Aravind Jeedipally, Avinash Rajamshetti, , (2016) "Car Number Plate Detection from Livestream", International Journal for Research in Applied Science & Engineering Technology, Volume 10 Issue VI June 2022. https://www.ijraset.com/research-paper/car-number-plate-detection-from-livestream
2. Kocer HE, Cevik KK. "Artificial neural networks based vehicle license plate recognition" Procedia Computer Science. 2011 Jan 1;3:1033-7 https://www.sciencedirect.com/science/article/pii/S1877050910005442
3. Sahu, Chinmaya Kumar, Sushree Barsa Pattnayak, Susantini Behera, and Manas Ranjan Mohanty. "A Comparative Analysis of Deep Learning Approach for Automatic Number Plate Recognition." Fourth International Conference on I-SMAC, pp. 932-937. IEEE, 2020. https://ieeexplore.ieee.org/abstract/document/9243424
4. Lin, Cheng-Hung, Yong-Sin Lin, and Wei-Chen Liu. "An efficient license plate recognition system using convolution neural networks." In 2018 IEEE International Conference on Applied System Invention (ICASI), pp. 224-227. IEEE, 2018. https://ieeexplore.ieee.org/document/8394573
5. Chen, Rung-Ching. "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning." Image and Vision Computing 87 (2019): 47-56. https://www.sciencedirect.com/science/article/pii/S0262885619300575
6. Saeed, Maham, Muhammad Gufran Khan, Adil Zulfiqar, and Muhammad Tahir Bhatti. "Development of ANPR Framework for Pakistani Vehicle Number Plates Using Object Detection and OCR." Complexity 2021 (2021). https://www.hindawi.com/journals/complexity/2021/5597337/
7. Beratoglu MS, Toreyin BU. "Vehicle license plate detector in compressed domain". IEEE Access. 2021 Jun 28;9:95087-96. https://ieeexplore.ieee.org/document/9466149
8. S. Shah, N. Rathod, P. K. Saini, V. Patel, H. Rajput, and P. Sheth, "Automated Indian vehicle number plate detection," in Soft Computing: *eories and Applications, pp. 453–461, Springer, Berlin, Germany, 2019.
9. N. Bolaj and G. Padalkar, "The license plate identification of fast moving vehicles," in Proceedings of the 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), pp. 1–5, Pune, India, August 2017. https://ieeexplore.ieee.org/document/8463655/
10. K. Tejas, K. A. Reddy, D. P. Reddy, K. Bharath, R. Karthik, and M. R. Kumar, "Efficient license plate recognition system with smarter Interpretation through IoT," in Soft Computing for Problem Solving, pp. 207–220, Springer, Berlin, Germany, 2019. https://link.springer.com/chapter/10.1007/978-981-13-1595-4_16
11. R. Panahi and I. Gholampour, "Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications,"IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 4, pp. 767–779, 2016. https://ieeexplore.ieee.org/abstract/document/7534742/

12. E.R. Buhus, D. Timis, and A. Apatean, "Automatic parking access using openalpr on raspberry pi3," Acta TechnicaNapocensis, vol.57, p. 10, 2016. https://users.utcluj.ro/~atn/papers/ATN_3_2016_2.pdf