Computers, Materials & Continua
DOI:10.32604/cmc.2022.017681
Article

Automatic License Plate Recognition System for Vehicles Using a CNN

Parneet Kaur[1], Yogesh Kumar[1], Shakeel Ahmed[2,*], Abdulaziz Alhumam[2], Ruchi Singla[3] and Muhammad Fazal Ijaz[4]

[1]Department of Computer Science, Chandigarh Group of Colleges, Landran, 140307, India
[2]Department of Computer Science, College of Computer Sciences and Information Technology, King Faisal University, Al-Ahsa, Saudi Arabia
[3]Department of Research, Innovations, Sponsored Projects & Entrepreneurship, Chandigarh Group of Colleges, Landran, 140307, India
[4]Department of Intelligent Mechatronics Engineering, Sejong University, Seoul, 05006, Korea
[*]Corresponding Author: Shakeel Ahmed. Email: shakeel@kfu.edu.sa
Received: 07 February 2021; Accepted: 02 April 2021

Abstract: Automatic License Plate Recognition (ALPR) systems are important in Intelligent Transportation Services (ITS) as they help ensure effective law enforcement and security. These systems play a significant role in border surveillance, ensuring safeguards, and handling vehicle-related crime. The most effective approach for implementing ALPR systems utilizes deep learning via a convolutional neural network (CNN). A CNN works on an input image by assigning significance to various features of the image and differentiating them from each other. CNNs are popular for license plate character recognition. However, little has been reported on the results of these systems with regard to unusual varieties of license plates or their success at night. We present an efficient ALPR system that uses a CNN for character recognition. A combination of pre-processing and morphological operations was applied to enhance input image quality, which aids system efficiency. The system has various features, such as the ability to recognize multi-line, skewed, and multi-font license plates. It also works efficiently in night mode and can be used for different vehicle types. An overall accuracy of 98.13% was achieved using the proposed CNN technique.

Keywords: Deep learning; intelligent transportation services; morphological; convolutional neural network; accuracy

1 Introduction

Intelligent Transportation Systems (ITS) [1] have undergone significant development over the past few decades. These systems are designed and used to enhance the movement of traffic. They aid in minimizing the adverse impact of vehicles on the environment. They gather all the data related to a vehicle to support a real-time monitoring mechanism [2]. A significant component of ITS is an automatic license plate recognition system. Each vehicle has a license plate with a unique identification number that is used for recognition purposes so that no tags, external cards, or transmitters are required [3]. The Automatic License Plate Recognition (ALPR) system was designed to recognize the license plates of vehicles. Machine Learning (ML) techniques were utilized for the execution of the character recognition step. ML is a term that signifies the capability of the system to learn and grow from its experiences. These techniques have been completely overshadowed by Deep Learning (DL) techniques, which provide better results. DL is mainly used for decision-making purposes but can include many other applications. This type of learning is highly beneficial in the fields of image analysis, speech and facial recognition, translation of languages, and emotion recognition [4]. These two learning techniques have been useful for traffic law enforcement and ITS, as these are the driving forces for designing an efficient ALPR system.

The ALPR system is a key component of modern transportation facilities. It aids in the provision of smart services, like traffic law enforcement, smart parking, electronic toll collection [5], and border control. The system has problems recognizing license plates due to weather conditions, orientation of captured images, and license plate color, texture, and shape. This work contributes a newly designed ALPR system that has the following features:

- A CNN-based architecture that can detect and recognize the license plate of a vehicle
- The system works on variations of license plates, such as those that have multiple fonts, are skewed, or have multiple lines of text.
- The system applies to different classes of vehicles, such as cars, auto-rickshaws, and bikes.
- The proposed CNN-based ALPR system is feasible for working in night mode.

The rest of this paper is organized as follows. Section 2 introduces the existing works in the research domain, including the various types of license plates and capturing modes. Section 3 describes the system design, method, and features of the proposed work, along with the graphical results and input images. Section 4 depicts the comparative analysis of existing work and the average recognition accuracy table. Finally, section 5 concludes and discusses future work.

2 Related Work

In this section, we briefly explain some recent research related to ALPR systems. The template matching technique is used for vehicle character recognition. In one study [6], the researchers discussed an ALPR system using extracted features based on template matching. Their approach to license plate data detection used gray scaling, dilation, normalization, and the Sobel operator for vertical edge detection while segmentation was performed by binarizing the image. OCR was performed by template matching using cross-correlation. A similar approach was proposed in [7], where researchers designed a program to recognize OCR-based car number plates by using template matching and the bounding box method. One study [8] recommended an ALPR system that uses technology for effective vehicle tracking based on machine learning. The extraction of the license plate data was performed by the bounding box technique and the template matching technique was used for character recognition.

Indian license plate recognition has always been a critical issue. Thus, several authors have developed ALPR systems specifically for Indian vehicles. In one study [9], researchers designed an ALPR system for an Indian university campus. The approach had three main modules: a) Faster R-CNN for training, b) Pre-processing that involves the Top-hat and Black-hat transforms and Gaussian smoothing filter, and c) the Tesseract OCR. In a separate study, researchers [10] proposed an ALPR system in which the input image was detected using the YOLO network, and the character recognition step was executed via YOLO.

One of the features that should be considered while designing an efficient ALPR system is its capability to recognize different types of license plates. Some of the issues related to plates that have been discussed in the literature include the following:

a) Multiple license plates: The ALPR system should be designed in such a manner so that it can recognize multiple license plates in a single frame. In one study, researchers [11] designed an ALPR system that consisted of two main parts: the detection phase, which included pre-processing done by applying a Sobel filter with equalized histogram and thresholding, and the recognition phase, which was performed by ANN.

b) Multi-angle license plates: An efficient system should recognize plates captured at varying angles. Some researchers [12] proposed an optimized learning model for multi-angle license plates. Character recognition was achieved through one-stage object detection by the Tiny-YOLOv2 model. The number of grids and anchor boxes was increased to correctly recognize the characters.

c) Multi-line license plates: In practical use, it is important to develop a system that can recognize plates that consist of more than one line. A team of researchers [13] proposed an ANN for both single and double-line license plate recognition that is free from the character segmentation step and they utilized a Bidirectional-LSTM model that predicts the sequence label for the characters.

d) Multi-style license plates: One study [14] proposed a Multi-Style License Plate Recognition (MSLPR) system for varying font styles that used a 2-layer Feed forward Back-Propagation ANN for the recognition of characters.

ALPR is a significant component of the Intelligent Transportation System (ITS). One team [1] discussed the scope of ITS services in the near future and proposed a novel architecture for it. Their work also listed some of the challenges in the field of ITS. Another team [15] designed an ALPR system that used a modified ACO algorithm for detecting the edges; segmentation of characters was performed in two phases: the CCA and Kohonen neural network. Character recognition was performed using inductive learning and the SVM approach. In one study [5], researchers proposed the implementation of an OCR system based on the Field Programmable Gate Array (FPGA). The framework of the system consists of a feed-forward neural network that uses an efficient neuron for the implementation of the OCR system. The system achieved an accuracy of 98.2% for character recognition. Similarly, researchers [16] proposed an ALPR system that used YOLO's seven layers for the detection of a single class. A sliding window was used to detect every character of the license plate and each window was detected by a YOLO framework. There was only one detection recognition phase that applies to dark and blurry images. One study [2] focused on using deep learning to resolve intelligent transportation issues. The use of deep learning techniques significantly decreased the delay of data transmission and enhanced the accuracy of the system.

3 Proposed Work

The literature review and background analysis of the existing systems led us to develop an ALPR system with specific objectives. The method proposed by the authors, the workflow of the system, and various techniques used in designing the systems are addressed in the next sections.

3.1 Objectives

The central aim of this work was to design an effective ALPR system. The proposed work addresses the following objectives:

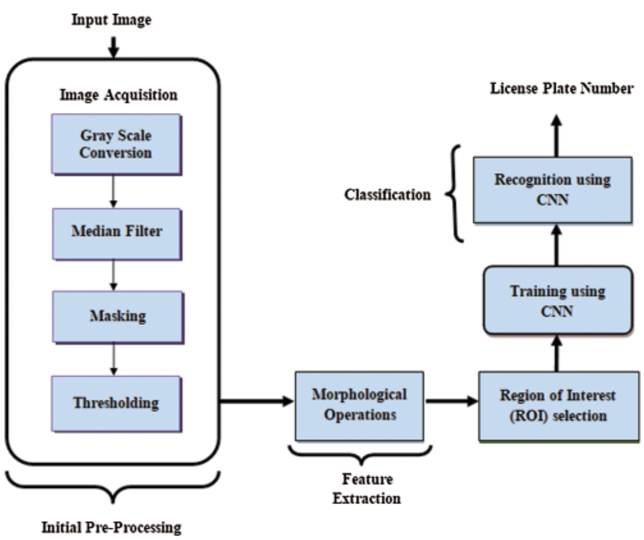
(1) To study existing methods of character segmentation and recognition and investigate the various pre-processing techniques available.
(2) To improve the quality of images taken from input sources using pre-processing methods and to extract the region of interest.
(3) To develop an efficient vehicle license plate recognition system that uses deep learning techniques and recognizes license plates with a good accuracy.

3.2 System Design

We developed an effective ALPR system that uses a CNN for the recognition of license plate characters. The system utilizes deep learning techniques. Existing systems have drawbacks with pre-processing. This system achieves good results in pre-processing via the median filter, masking, and, thresholding. Certain morphological operations were applied to perform feature extraction efficiently. Feature extraction is an essential step; the license plate character area was identified and then extracted for further recognition, as shown in Fig. 1.

3.3 Method

The ALPR model proposed in this work has a main GUI that involves features such as acquiring the image, pre-processing, applying thresholding on the image, masking, training the model, and performing the classification. All these features are executed using push buttons.



Step 1. Select an input image from the specified folder "images".
Step 2. Perform filtering on the selected image by initially converting the colored image into a greyscale image and then applying the Median filter on the input image.
Step 3. Implement the Masking on the input image to remove unwanted impurities and perform edge detection to normalize the image.
Step 4. Perform Thresholding on the image that divides the pixels into two groups.
Step 5. Perform Training by using the CNN model.
Step 6. Further, Morphological operations are applied in the feature extraction step.
Step 7. The Region of Interest that is the region containing the license plate is detected and segmented from the vehicle image.
Step 8. Extraction of the region is done.
Step 9. Classification of the region of interest is performed using the CNN network.
Step 10. Finally, the classified license plate is displayed after recognition and the performance evaluation is done.

Figure 1: Workflow and steps of the proposed system

3.3.1 Implementation of the Image Acquisition Step

The system begins working when the "Upload Image" button is pressed. Its function is to upload the image to the system for the execution of further steps. Since the execution of the ALPR system demands specific types of vehicle images, a folder named "images" was created in the current directory, which consisted of 160 images with all types of vehicles, multi-line license plate images, multi-font license plate images, images captured in different illumination conditions. This folder was created because of the drawbacks of the datasets in use by existing systems.

3.3.2 Implementation of the Pre-processing Step

Our system executes this step by applying four techniques: gray scaling, median filter, thresholding, and masking as shown in Fig. 2.

- *Gray scaling:* This is performed to convert the colored image components into grayscale components of the image that lie between 0 and 255.
- *Median filter:* The main function of using this filter is to reduce noise. It is performed to restore an image that is free of any impurities.
- *Masking:* A vertical Sobel operator is used initially for the detection of edges followed by border replication to reduce any dark areas created around the edges.
- *Thresholding:* Otsu's method is executed to select a threshold that decreases the variance between the classes. It divides the pixels into two groups.

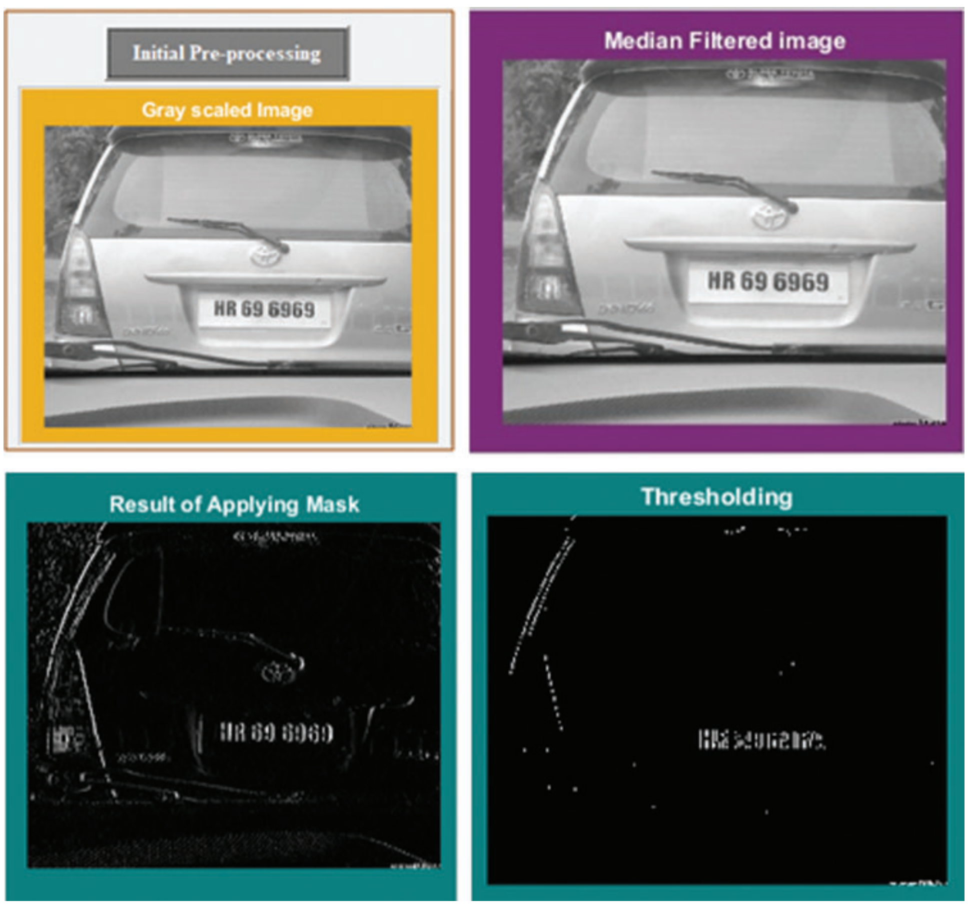


Figure 2: Results of pre-processing

3.3.3 Basic Architecture of the CNN

A convolutional neural network (CNN) (such as ConvNet [17], Fig. 3) is a type of machine learning in which a model learns to perform classification for applications that call for object detection and computer vision. CNN eliminates the need for manual feature extraction since the features are directly learned by the network. CNNs can be retrained for new tasks that can build on pre-existing networks.

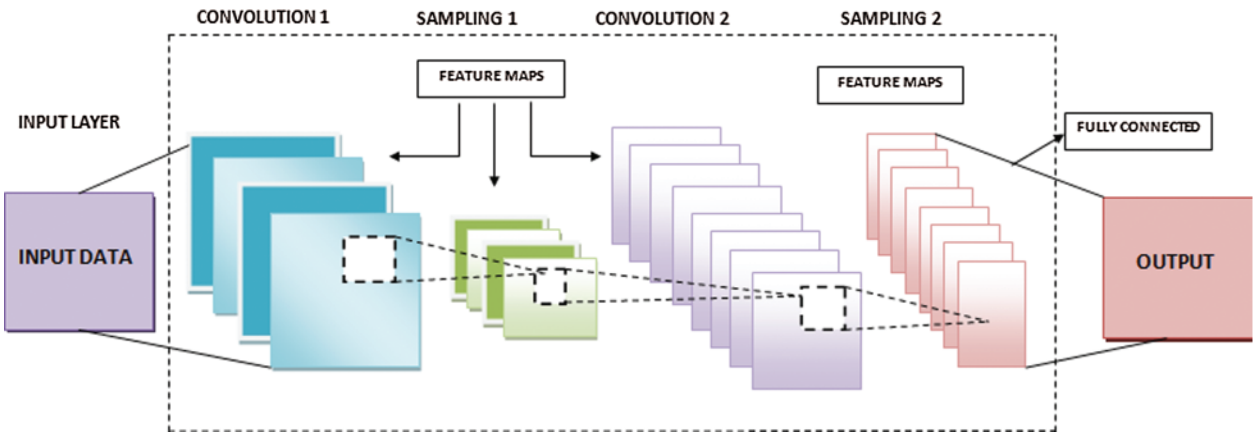


Figure 3: Architecture of CNN

The basics of CNN are as follows:

- *Feature Learning*

A CNN network can be composed of several layers that learn to detect the unique features of an image independently. Every training image passes through a filter at varying resolutions, and the result of every convolved image is fed to the next layer of the network.

- *Layers*

Like any other network, a CNN comprises certain layers that perform data-altering operations with the motive of feature learning. Some of the most commonly used layers are convolution, ReLU, and pooling. A brief description of the CNN layers is given in Tab. 1.

- *Classification*

The step of feature learning conducted in several layers is followed by the final classification step. The classification is performed using the softmax layer and classification layer. Tab. 2 describes some of the output layers.

3.3.4 CNN Training

Deep learning is a subset of machine learning that encompasses networks capable of learning unsupervised data. Deep learning tries to imitate the functions of the human brain to process data and design patterns for decision making. ConvNet is one of the most widely used deep learning algorithms that performs classification directly from images or videos. It is a neural network that has more than one convolutional layer. Below are the subtasks required to achieve the training of a CNN:

- *CNN Masking Layer:* The mask is created to improve the region of focus before training, as shown in Fig. 4.
- *Morphological Operations:* Several operations for image processing, such as dilation, erosion, and filling, are tried on the image to improve the focus on the license plate region.

- *CNN Training:* The proposed system uses the Deep Learning Toolbox provided by MATLAB to train the networks graphically. This system trains the CNN network using the trainingOptions () and the trainNetwork () functions. The CNN training is shown in Fig. 4.

Table 1: Different layers of CNN

Layer name	Description
Input	These layers provide the network with an input of 2 dimensional and 3-dimensional images. Also, they perform normalization of data.
Convolution	These layers are a group of filters that are responsible for activating specific features from the input images.
Fully Connected	This layer is responsible for multiplying the inputs by a certain weight and adding a bias factor.
Sequence	These are responsible for entering sequence data into a network.
Activation	These layers are responsible for executing a threshold operation on every element of the input. It assigns any value less than zero to the value zero. Another feature is its effective training and fast speed since it maintains positive values only. Only the activate features are forwarded hence, called as activation layer.
Normalization	Every input channel is normalized in this layer across a mini-batch. These layers fasten the training of CNN.
Dropout	The function of this layer is to randomly assign input elements to 0 with a given probability.
Cropping	A 2-dimensional cropping layer crops the inputs in two dimensions while the 3-dimensional layer crops a 3-D volume to the size of the input feature map.
Pooling	These layers simplify the output by performing nonlinear down-sampling, reducing the number of parameters that the network is required to learn.

Table 2: Output layer types

Layer name	Description
Softmax	A softmax function is applied to the input in this layer.
Classification	It calculates the cross entropy loss for multi-class classification problems with mutually exclusive classes. This layer infers the number of classes from output size of the previous layer.
Regression	It calculates the half-mean-squared-error loss for regression problems.

The following operations are performed for the execution of the CNN training:

1) Layers

These are the layers that define the architecture of neural networks for deep learning. An array of layers is directly created to specify the architecture of the CNN network. The proposed system uses the following layers that specify the architecture: the input layer, fully connected layer, softmax layer, and classification layer.

2) trainingOptions ( )

MATLAB also provides a trainingOptions () function that defines the options, such as the CPU, MaxEpochs, Plots, GradientThreshold, and ExecutionEnvironment, to train a neural network. The syntax is shown in Eq. (1). This command returns the options of training for the optimizer specified by solverName.

$$\text{options} = \text{trainingOptions}(\text{solverName}), \tag{1}$$

3) trainNetwork ( )

The trainNetwork () function is responsible for training the network for deep learning. The proposed system uses this function to return information on training using any of the input arguments in the format, as shown in Eq. (2).

$$[\text{net}, \text{info}] = \text{trainNetwork}(\_), \tag{2}$$

As soon as the training is completed, a message box appears with a "Training done" message.

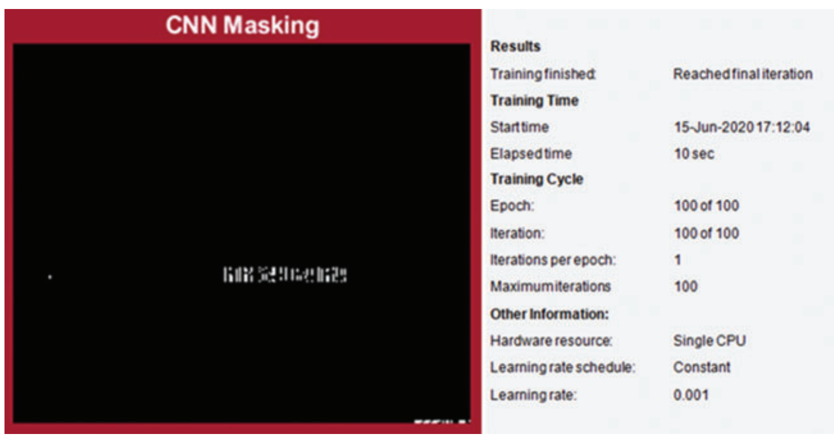


Figure 4: CNN masking and training

3.3.5 Classification Phase

The final step of classifying the license plate is performed using the CNN. The CNN has several layers that help in learning the features as well as classifying the new inputs based on this learning. Various functions of MATLAB are used to classify the region of the license plate when the "Classify" button is pressed, as depicted in Fig. 5.

3.4 Features of the Proposed System

The various features of the proposed system when applied to different kinds of images are described below. The results have been divided into various subsections based on the type of license plate, vehicle images, and vehicle classes.

Figure 5: Classification result

3.4.1 Multi-line Plates

Our system can recognize various types of license plates, even multi-line license plates, which are plates having more than one line of characters as shown in Fig. 6. The precision-recall curve represents the trade-off between these two terms. A high precision is associated with a low false-positive rate, and a high recall is associated with a low false-negative rate. High scores for both precision and recall show that the classifier returns accurate results as well as returns the majority of all positive results as shown in Fig. 6.

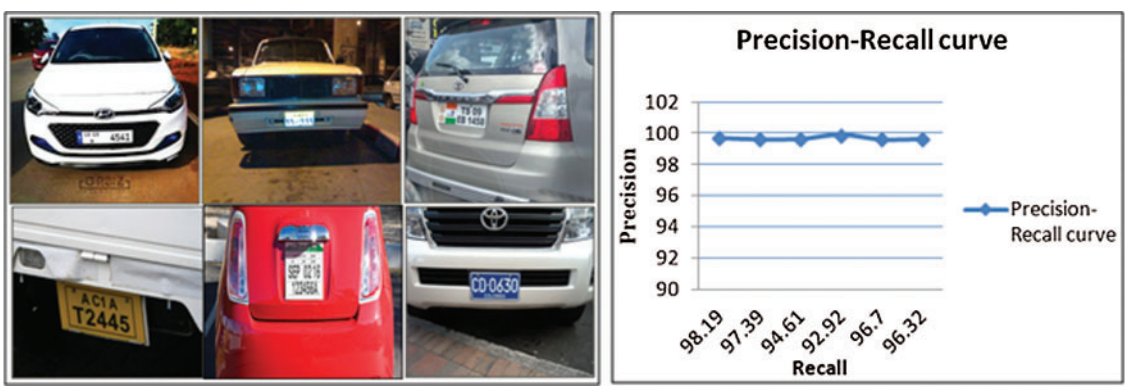


Figure 6: Multi-line license plate images with precision-recall curve

Tab. 3 shows the recognition accuracies of various multi-line license plate images when given as input to the proposed system.

3.4.2 Tilted or Skewed License Plates

Many of the existing systems are unable to recognize and detect tilted or skewed plates captured from varying angles of the camera. The proposed system is capable of addressing this issue and can recognize tilted or skewed plates from an image. Some of the images captured at varying angles are shown in Fig. 7.

Tab. 4 lists the recognition accuracies of five randomly chosen skewed license plates. The system has been tested on 19 multi-angle or skewed license plate images and had an average recognition rate of 96.91%.

Table 3: Recognition accuracies of multi-line images

Image	Recognition accuracy
Test_image1	98.98
Test_image2	98.58
Test_image3	97.14
Test_image4	96.06
Test_image5	98.26
Test_image6	98.04
Average recognition rate = 97.84%	

Figure 7: Multi-angle license plate images

Table 4: Recognition accuracies of multi-angle images

Image	Recognition accuracy
Test_image7	97.18
Test_image8	99.59
Test_image9	97.51
Test_image10	97.52
Test_image11	94.34
Average recognition rate = 96.91%	

3.4.3 Night Mode Recognition

The proposed system is also capable of recognizing license plates in night mode. Recognizing plates at night is difficult due to a lack of illumination and the reflection of headlights on the license plates. Our system was tested at night, and the vehicle images are shown in Fig. 8.

From the dataset, five images were chosen randomly to analyze the results for recognition in night mode. Fig. 8 indicates that night mode recognition accuracy is above 95%, which demonstrates the efficacy of our proposed system. The average recognition accuracy for night mode is 97.52%.

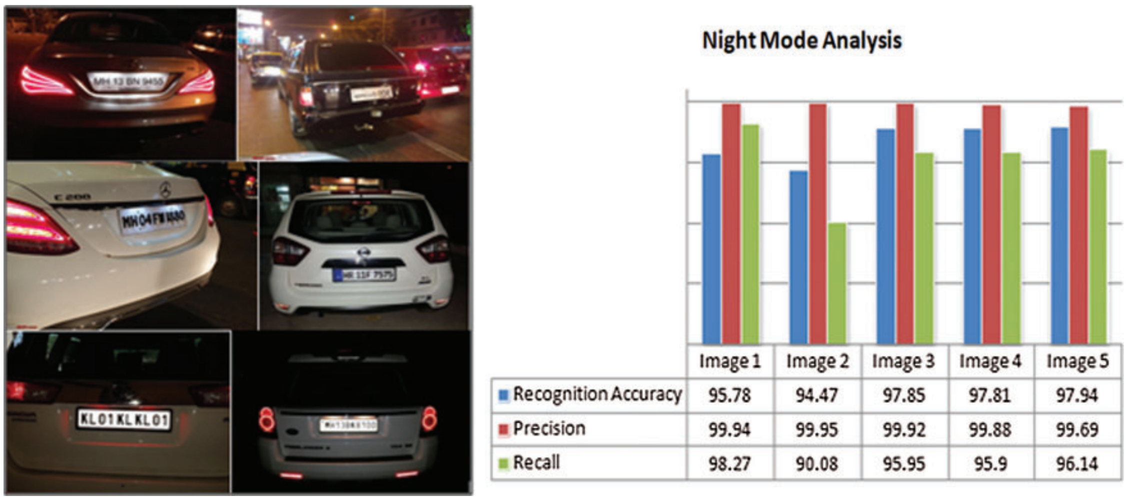


Figure 8: Night mode recognition

3.4.4 Multi-Font License Plates

Another feature of the proposed system is the ability to detect and recognize license plates that have various types of fonts. Some of the vehicle images containing multi-font license plates are shown in Fig. 9.

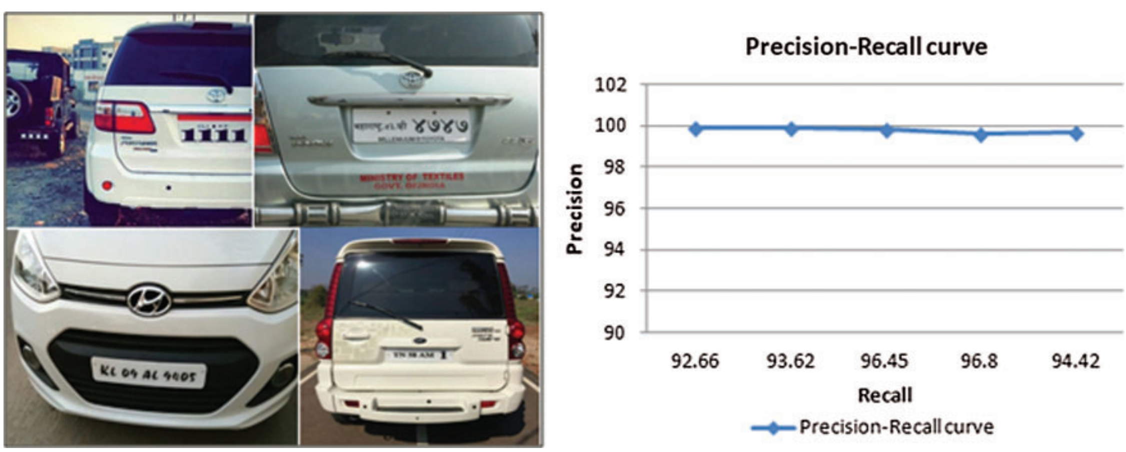


Figure 9: Multi-font license plate images with precision-recall curve

The line chart shown in Fig. 9 is plotted based on the results for precision and recall according to the input images. The precision-recall curve shows a high-rise area underneath the curve representing both high recall and high precision. Since the precision-recall curve is constant around 99%, the system is efficacious for these images because the misclassification rate is low.

Tab. 5 shows the recognition accuracies of various multi-line license plate images when given as input to the proposed system.

Table 5: Recognition accuracies of multi-font images

Image	Recognition accuracy
Img1	96
Img2	96.55
Img3	98.07
Img4	98.25
Img5	96.99
Average recognition rate = 97.17%	

3.4.5 *Different Vehicle Types*

The proposed system is also capable of recognizing the license plates of different types of vehicles, such as bicycles and cars. Some of the bicycle and auto-rickshaw images are shown in Fig. 10.

The recognition accuracies of the bicycle, car, and, auto-rickshaw images are plotted on the bar graph. Fig. 10 shows that the proposed system is equally applicable to various types of vehicles and proves to be highly efficient for all types of vehicles. The recognition accuracy has no significant difference when considering car license plate images, auto-rickshaw, or bicycle license plate images

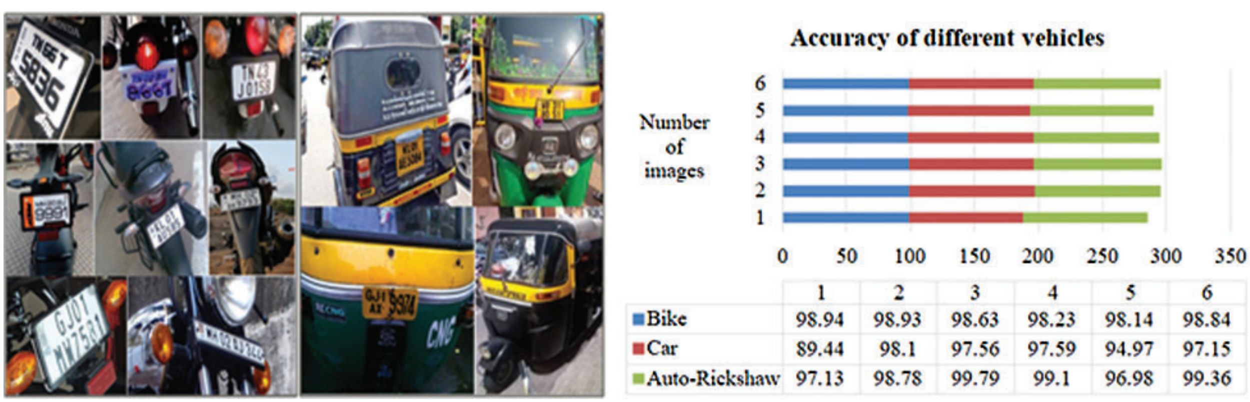


Figure 10: Different vehicle type images and recognition accuracies

The proposed system was tested on 11 bike images that produced a recognition accuracy of 98.47%. The system observed a recognition accuracy of 98.52% when tested on six auto-rickshaw images and 97.41% when tested on 144 car images as shown in Tab. 6.

Table 6: Average recognition accuracy of different vehicles

Vehicle category	Number of images	Accuracy (%)
Bike	11	98.47
Auto–rickshaw	6	98.52
Cars	144	97.41

4 Comparative Analysis

ALPR systems are used because of their effectiveness. There have been many systems proposed in the last decade, and some researchers have also made significant enhancements to the existing systems. There are still many drawbacks to the current ALPR systems. The present section addresses the comparative table as shown in Tab. 7 of the proposed method with existing ones that have used deep learning as the recognition technique.

Table 7: Comparison table

S.no.	Author(s)	Technique(s) used	Recognition accuracy (%)
1.	Abedin et al. [4]	CNN	98
2.	Barreto et al. [18]	Modified YOLO-VOC	92.76
3.	Babu et al. [10]	YOLO network	91
4.	Goncalves et al. [19]	Multi-task deep CNN	88.8
5.	Usmankhujaev et al. [20]	CRNN	95.7
6.	Jagtap et al. [21]	2-layer feed-forward backpropagation ANN	87
7.	Laroca et al. [22]	Fast YOLO network	81.80
8.	**Proposed work [Different vehicle types]**	**CNN**	**98.13**

From the comparison in Tab. 7, it can be inferred that the proposed system is highly efficient and better than the existing systems that use deep learning recognition techniques.

Fig. 11 shows that the system developed in a prior study [4] and our system have 98% and 98.13% recognition accuracies, respectively. The system from the other study uses a CNN for the recognition of the license plate data, but the major drawback is that it is unable to work with more than one vehicle type; it also cannot be applied to images captured at night. Our proposed system overcomes these issues.

4.1 Average Recognition Accuracy

Tab. 8 shows the average recognition accuracies of multi-line and multi-font license plates of different types of vehicles. The average recognition accuracy in night mode and overall recognition accuracy of the proposed system are shown in this table.

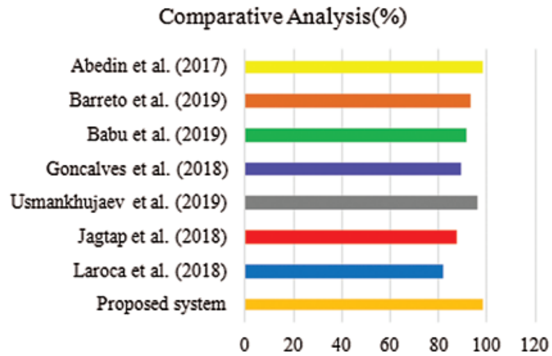


Figure 11: Graphical representation of comparative analysis

Table 8: Average recognition accuracy based on different features

Feature(s) of the system	Average recognition accuracy (%)
Multi-line license plate	97.84
Multi-font license plate	97.17
Tilted license plate images	96.91
Different vehicle types	Bike: 98.47, Auto–rickshaw: 98.52, Cars: 97.41
Night mode	97.52
Overall system accuracy	**98.13**

5 Conclusion

The current era is witnessing a tremendous increase in population growth, which has resulted in many more vehicles being on the roads. This ever-growing number of vehicles has led to some major challenges, such as increased traffic violations, theft, terrorist activities, parking problems, and accidents. It is important to address these issues. To solve these problems, an ALPR system was designed that can effectively identify data on the license plate of a vehicle. This work introduced an ALPR system that utilizes a deep learning technique (CNN) for character recognition. The proposed ALPR system consists of four main phases (Data Acquisition and Pre-processing, Feature Extraction, Region of Interest (ROI) selection, and Classification). Pre-processing techniques are applied (gray scaling, median filter, thresholding, and masking). Then, a CNN masking layer is used to further enhance image quality, which is then followed by the CNN training. Finally, CNN is employed for the recognition of the license plate data.

The presented technique was executed in MATLAB. The system performance was evaluated based on precision, recall rate, and recognition rate. The results demonstrated the efficiency of the ALPR system as it achieved a high recognition rate of 98.13% when tested on 160 images. These tested images contained all types of images, such as multi-line license plates, skewed or multi-angle license plates, and multi-font license plates. The system works efficiently in night mode as well. Although the proposed system is highly efficient and has a high recognition rate, there are also some limitations, such as the recognition of multiple license plates in a single frame. The CNN cannot distinguish between the plate area and the grills of the motor vehicle. Future work will address these issues.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Lin, P. Wang and M. Ma, "Intelligent transportation system (ITS): Concept, challenge and opportunity," in *2017 IEEE 3rd Int. Conf. on Big Data Security on Cloud, IEEE Int. Conf. on High Performance and Smart Computing, and IEEE Int. Conf. on Intelligent Data and Security*, Beijing, China, vol. 2, pp. 167–172, 2017.

[2] Z. Lv, S. Zhang and W. Xiu, "Solving the security problem of intelligent transportation system with deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 1–10, 2020.

[3] M. Arora, A. Jain, S. Rustagi and T. Yadav, "Automatic number plate recognition system using optical character recognition International Journal of Scientific Research in Computer Science," *Engineering and Information Technology*, vol. 5, no. 2, pp. 986–992, 2019.

[4] Y. Kumar, K. Kaur and G. Singh, "Machine learning aspects and its applications towards different research areas," in *2020 Int. Conf. on Computation, Automation and Knowledge Management*, Dubai, United Arab Emirates, vol. 3, pp. 150–156, 2020.

[5] Yuan Jing, B. Youssefi, M. Mirhassani and R. Muscedere, "An efficient FPGA implementation of optical character recognition for license plate recognition," in *2017 IEEE 30th Canadian Conf. on Electrical and Computer Engineering*, Windsor, ON, Canada, vol. 10, pp. 1–4, 2017.

[6] N. Saleem, H. Muazzam, H. M. Tahir and U. Farooq, "Automatic license plate recognition using extracted features," in *2016 4th Int. Symp. on Computational and Business Intelligence*, Olten, Switzerland, vol. 8, pp. 221–225, 2016.

[7] I. Kusumadewi, C. A. Sari, D. R. Moses Setiadi and E. H. Rachmawanto, "License number plate recognition using template matching and bounding box method," *Journal of Physics: Conference Series*, vol. 1201, no. 2, pp. 1–9, 2019.

[8] K. Indira, K. V. Mohan and T. Nikhilashwary, "Automatic license plate recognition," *Recent Trends in Signal and Image Processing*, vol. 3, no. 7, pp. 67–77, 2018.

[9] C. Dias, A. Jagetiya and S. Chaurasia, "Anonymous vehicle detection for secure campuses: A framework for license plate recognition using deep learning," in *2019 2nd Int. Conf. on Intelligent Communication and Computational Techniques*, Jaipur, India, vol. 4, pp. 79–82, 2019.

[10] R. Naren Babu, V. Sowmya and K. P. Soman, "Indian car number plate recognition using deep learning," in *2019 2nd Int. Conf. on Intelligent Computing, Instrumentation and Control Technologies*, Kannur, India, vol. 2, pp. 1269–1272, 2019.

[11] A. Menon and B. Omman, "Detection and recognition of multiple license plate from still images," in *2018 Int. Conf. on Circuits and Systems in Digital Enterprise Technology*, Kottayam, India, vol. 7, pp. 1–5, 2018.

[12] C. Lin and C. Wu, "A lightweight, high-performance multi-angle license plate recognition model," in *2019 Int. Conf. on Advanced Mechatronic Systems*, Kusatsu, Japan, vol. 2, pp. 235–240, 2019.

[13] S. Du, M. Ibrahim, M. Shehata and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, 2013.

[14] Y. Cao, H. Fu and H. Ma, "An end-to-end neural network for multi-line license plate recognition," in *24th Int. Conf. on Pattern Recognition*, Beijing, China, vol. 13, pp. 3698–3703, 2018.

[15] A. Sasi, S. Sharma and A. N. Cheeran, "Automatic car number plate recognition," in *2017 Int. Conf. on Innovations in Information, Embedded and Communication Systems*, Coimbatore, India, vol. 11, pp. 1–6, 2017.

[16] Hendry and R. Chen, "Automatic license plate recognition via sliding-window darknet-YOLO deep learning," *Image and Vision Computing*, vol. 87, no. 14, pp. 47–56, 2019.

[17] M. Paolanti, R. Pierdicca, M. Martini, A. Felicetti, E. S. Malinverni *et al.,* "Deep convolutional neural networks for sentiment analysis of cultural heritage," *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 5, no. 10, pp. 871–878, 2019.

[18] S. C. Barreto, J. A. Lambert and F. Vidal, "Using synthetic images for deep learning recognition process on automatic license plate recognition," *Lecture Notes in Computer Science*, vol. 2, no. 6, pp. 115–126, 2019.

[19] G. R. Goncalves, M. A. Diniz, R. Laroca, D. Menotti and W. R. Schwartz, "Real-time automatic license plate recognition through deep multi-task networks," in *2018 31st SIBGRAPI Conf. on Graphics, Patterns and Images*, Parana, Brazil, vol. 3, pp. 110–117, 2018.

[20] S. Usmankhujaev, S. Lee and J. Kwon, "Korean license plate recognition system using combined neural networks," in *16th Int. Conf. on Distributed Computing and Artificial Intelligence*, Korea, vol. 5, pp. 10–17, 2019.

[21] J. Jagtap and S. Holambe, "Multi-style license plate recognition using artificial neural network for indian vehicles," in *2018 Int. Conf. on Information, Communication, Engineering and Technology*, Pune, India, vol. 3, pp. 1–4, 2018.

[22] R. Laroca, E. Severo, L. Zanlorensi, L. Oliveira, G. Goncalves *et al.,* "A robust real-time automatic license plate recognition based on the YOLO detector," in *2018 Int. Joint Conf. on Neural Networks*, Rio de Janeiro, Brazil, vol. 3, pp. 1–10, 2018.