IEEE - 56998

Vehicle Number Plate Detection System Using Image Processing Approach

NI Tushar
Department of Computer Science and Engineering
Bangladesh University of Business and Technology
Dhaka, Bangladesh
nazrulislamtusar@gmail.com

Abdullah Saikat
Department of Computer Science and Engineering
International University of Business Agriculture and Technology
Dhaka, Bangladesh
abdullahsaikat77@gmail.com

A.A.M Rahat-Bin-Rafique
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
rahat15-10978@diu.edu.bd

Md. Minhajul Abedin
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
minhajul15-11140@diu.edu.bd

K.B.M Tahmiduzzaman
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
tahmiduzzaman15-11150@diu.edu.bd

Md.Thoufiq Zumma
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
thoufiq15-10968@diu.edu.bd

2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 979-8-3503-3509-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCNT56998.2023.10307006

***Abstract*— The primary method of operation for the vehicle authorization system is the reading and processing of license plates. Automatic Number Plate Recognition (ANPR) is the name given to image processing technology that uses a vehicle's license plate to identify it. In this work, we aim to develop a reliable automatic authorized vehicle identification system that makes use of the license plate. This system is intended to be used in highly restricted military areas like the Supreme Court and Parliament as well as government administration and security control systems. The number plate region in an image is found using image segmentation, which is then used to extract the vehicle's number plate from the image. The highly trained neural network serves as the foundation for the recognition system. A database used for vehicle authorization contains the specific vehicle information.**

Keywords — Vehicle, ANPR, Machine learning, Image processing, Neural network,

I. Introduction

To identify the name of a person or thing, use a nameplate. Unlike nameplates, which are typically fixed to an item (such as cars or amplifiers) or physical area, name badges are typically worn on uniforms or clothing. Name plaques and nameplates are two different things. Whenever possible, place a nameplate on the main entrance or a nearby wall because it is believed to increase success, opportunity, and wellbeing. Make sure the nameplate is legible from at least a foot or two away, large enough to write one's name or surname, and the home number.

The procedure for gaining access to a connected smart car is known as vehicle authentication. A driver or passenger would provide one or a combination of authentication elements, such as knowledge, possession, or the discovery that the relationship persists, when authenticating to the car, similar to any online resource. The term "automatic number-plate recognition" (ANPR; also known by other names below) refers to a technology that reads vehicle identification numbers from photos using optical character recognition [8]. Systems that enable the camera to take pictures at any time of day or night frequently employ infrared illumination. A camera that is connected to a computer makes up an ANPR system.

The vehicle registration mark (VRM) is recorded when a car passes in front of the camera, which then takes a picture that is automatically read by the computer. In England, Wales, and Scotland, LEA use ANPR to provide lines of inquiry and evidence in criminal investigations. In general, a number plate recognition system consists of a "frame-grabbing instrument" that can take a picture, look for the location of a number plate within the picture, and then use optical character reading (OCR) tools to translate the image's pixels to identify the relevant symbols. The standardized license plates will include features like RFID (radio frequency identification), which will help law enforcement locate a stolen car. Gedam stated, "We will issue a notification for HSNP to be installed on all cars registered in the state following the tender process." In today's hectic environment, license plate recognition (LPR) is crucial due to the growing number of vehicles on the road. Registration code recognition is being used to stop this behavior as a result of an increase in vehicle theft, traffic infractions, and entering restricted areas.

In this research, we develop a model to automatically decode plate patterns and extract the number. Because there are various types of license plates, the online datasets are not entirely accurate. This is the first architecture for detecting license plates that we have ever seen.

II. Literature Review

In [1] Support Vector Machine (SVM) has been their main focus when it comes to number plate recognition. For character segmentation and recognition in their LPR systems, they have utilized the SVM concept. They tested each character on the license plate individually using their trained model. Their SVM-based model fails in every situation, including dim lighting, the motion blur effect, and faded characters. To identify license plates in the aforementioned situations, such as those involving poor image quality, they have used multiclass classifiers. For the purpose of detecting number plates on their work, they have created a seven-step algorithm. In [2] They have suggested a design that Mat-Lab will use in an embedded system. Their system was created with the goal of securing the parliament building, military bases, toll booths, parking lots, etc.

By incorporating the RTO database, their proposed system can extract the vehicle owner's information. They employed an embedded system, which is utilized in home automation, industrial data acquisition, and environmental monitoring systems. And it's entirely wireless. They implemented their system using a variety of sensors and IoT-based hardware. In [3] They used a pixel-based segmentation algorithm to separate the license plate's alphanumeric characters. 150 different license plates from different nations were used to

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:44:54 UTC from IEEE Xplore. Restrictions apply.

test their system. And their accuracy rate is 91%. In their paper, they demonstrate in figure 6 how they improved accuracy by increasing the number of license plates. They used component labeling, adaptive thresholding, medium filtering, and region growing in their segmentation work. In [4] they talk about the vehicle's pattern-based license plate recognition system. Capture an image first, then process it to identify license plates. The proposed system uses a clever algorithm to recognize the letters and numbers on Alberta license plates when it comes to pattern matching. In license plates, there are two different character types: letters and numbers. It compares the characters with the pattern of letters from A to Z for letter characters and the pattern of digits from 0 to 9 for number characters.

In [5] The work on an algorithm that localizes the license plate area, extracts and segments characters, and finally recognizes and decodes the registration number from a vehicle image is presented in this paper. The procedures based on the MATLAB R2012b software were followed for the VNPDR's overall design processes. For VNPDR operations, some function, script, and command were needed. When working with images, the Image Processing toolbox in MATLAB R2012b Simulation software was essential. In [6] They demonstrated software intended for applications that recognize vehicle license plates. They first extracted the plate location, then used segmentation to individually separate the plate characters, and finally used template matching in conjunction with correlation to recognize the plate characters. This system is intended to identify Turkish license plates, and it has undergone extensive testing using numerous images. Finally, it has been demonstrated to be 92.57% accurate for the extraction of the plate region, 96% accurate for character segmentation, and 98% accurate for recognition unit accuracy.

In [7] This algorithm extracts, segments, and recognizes characters from vehicle number plates. Images with various size, background, illumination, camera angle, distance, etc. are included in the image database. According to the experimental findings, number plates can be faithfully extracted with an 85% success rate using vertical edge detection and connected component algorithms. A success rate of 80% is achieved during the character segmentation phase using connected component analysis and vertical projection analysis. The character recognition success rate is 79.84%.

III. PROPOSED METHODOLOGY

The technique of the suggested architecture is detailed in this section. Four segments make up this section's subdivisions. With thorough justifications, the subsections are sequentially arranged from the model's input to output phase. Figure 3.1 also shows the architecture's entire process.

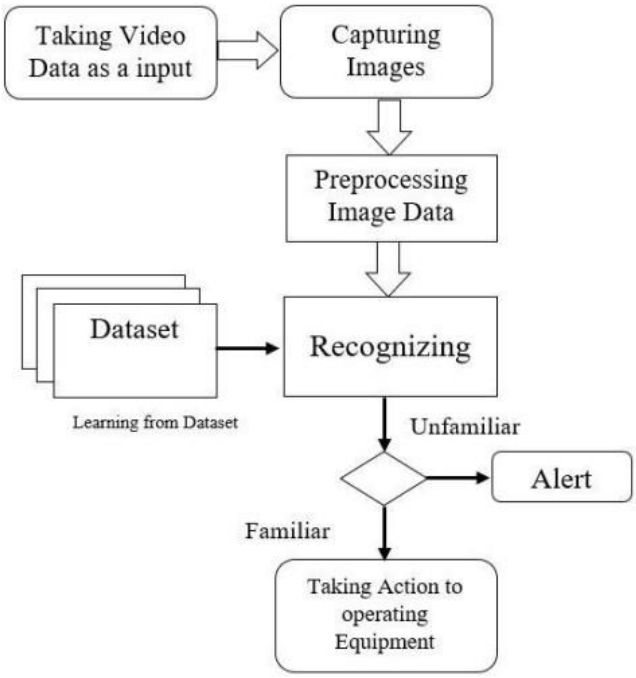

Fig. 1. The figure illustrates the workflow of the proposed system

Image normalization is a crucial pre-processing technique. It is regarded as intensity offsets and lessens the gap in inner-class features. The local area has preset intensity offsets. Gaussian normalization and the standard deviation are therefore helpful when normalizing. Equation computes the final picture after normalization.

$$\psi(\pi,\theta) = \frac{\varepsilon(\pi,\theta)\,\mu(\pi,\theta)}{6\sigma(\Pi,\theta)}$$

$$\mu(\pi,\theta) = \frac{1}{M^2}\sum_{k=-\alpha}^{\alpha}\sum_{n=-\alpha}^{\alpha}[\varepsilon(k_+u, n+\theta)]$$

$$\dot{\mu}(\pi,\theta) = \sqrt{\frac{1}{M^2}\sum_{k=-\alpha}^{\alpha}\sum_{\mu=-\alpha}^{\alpha}[\textstyle\sum(k+u, n+\theta)]^2}$$

A. *Dataset*

We note that the majority of vehicle-included picture datasets on the web are developed for identifying whether or not automobiles are present in an image. However, we require a dataset that is split into the classes vehicle and non-vehicle in order to evaluate a number plate. Therefore, to construct our vehicle-based picture dataset, we employed web gathering techniques to collect photos from the web that either included or did not include vehicles. As previously said, we add the class vehicle and not vehicle picture to our dataset. Along with the class name, provide each of the two classes respecting keywords (vehicle and not vehicle). Image 4.1 The example photos of cars and other objects from the dataset we utilized for this study are shown below, from top to right. The online collecting method was used to collect these photographs from the internet. We have amassed a big collection of photographs from the same category for Criminal. 1500 photos were finally put for the four classes. During assessments, we utilized 1200 photos, or 80%, for training and 300 images, or 20%, for testing. Additionally, there are 484 examples in each class, and the photos of the two classes are spread equally. Figure 4.1 displays a few examples of photos from the Kaggle source

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:44:54 UTC from IEEE Xplore. Restrictions apply.

. Fig 2: Sample Dataset

Fig 4: Vehicle Detection

```
1/1 [==============================] - 0s 59ms/step
This is Vehicle
```

Fig 5: Object detection

B. *Vehicle Detections*

Utilizing technology, automobiles may be recognized and located in photos or video feeds. It is a crucial component of autonomous systems, including traffic monitoring, intelligent transportation systems, and self-driving automobiles. Accurately detecting and locating cars in a scene is the aim of vehicle detection. Making predictions regarding the presence and position of cars entails analyzing visual data using computer vision and machine learning techniques.

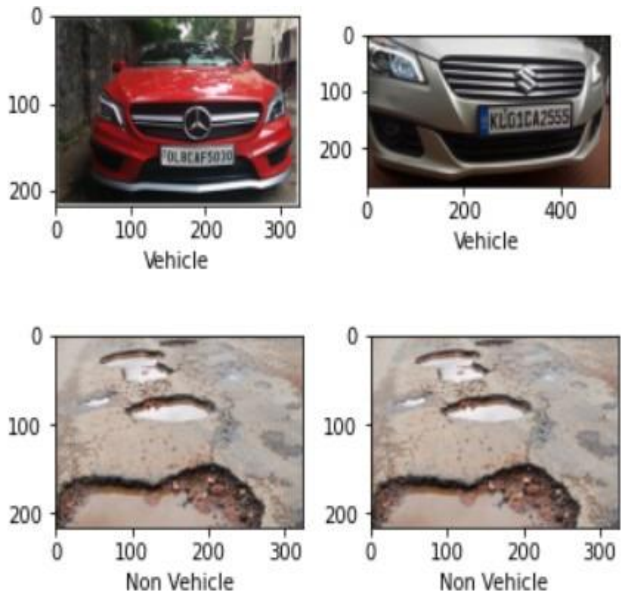

Fig 3: Vehicle detection

After using coding section fit the model, finally get vehicle detection. The dataset has two types of objects. Vehicle and non-vehicle. In fig 3 are showing detection which objects are vehicle and which are non-vehicle.

We use a random picture in the coding sections, the model detects vehicle perfectly and fig 5 is showing which type object it detects. And model detect perfectly. The random image was a vehicle and the detection image is also a vehicle.

C. *Template Detection*

A computer vision method called template detection is used to locate a certain pattern or template inside an image or video stream. It entails locating the area where the template and source picture match by comparing a target template image to a source image. The method is often used to pattern matching, object identification, and face recognition.

Fig 5: template detection

A threshold is chosen to assess if a template has been detected in the source picture, and a correlation metric is used to calculate how similar the source and target images are. The required degree of accuracy and the particular

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:44:54 UTC from IEEE Xplore. Restrictions apply.

needs of the application are often taken into account when determining the threshold value.

Due to the need to compare every potential point in the source picture to the template image, this approach might be computationally costly. However, if the target item is well-known and well-defined and the source picture is relatively tiny, it may be useful for simple object identification tasks.

```
Number Plate Position: [[238, 230],
[570, 230], [570, 304], [238, 304]]

Vehicle Number: 'DL7C N 5617'

Text Detection Accuracy:
0.7414511048497275
```

Fig 6. Template Detection

Easy OCR was developed using Python and the PyTorch deep learning framework; a GPU might hasten the whole detecting procedure. The CRAFT method is used for the detection portion, while the CRNN model is used for recognition.
In fig 6 showing detect template number detection. It finds out by using easy OCR. First it detects text with text position in photos, easy OCR will extract text from images.

D. User interfaces

In this research project using tinker for user interface. All common Python distributions contain Tkinter, which is the de facto method for creating Graphical User Interfaces (GUIs) in Python. It's the only framework that is included in the Python standard library.

IV. Results and Discussion

For the implementation of our combined CNN-RNN architecture, we employ the NAHFE dataset. The combined CNN-RNN [10] approach's prediction accuracy, precision, and recall for diagnosing human anomalies are displayed in Table 4.1. This table compares the projected outcome of the proposed architecture with the standard CNN [9] architecture and shows how well the suggested architecture performed. Our research revealed that classifying human anomalies accurately requires more than just CNN architecture. While the CNN-RNN combined technique provides 0.895 accuracy, the basic CNN architecture only provides 0.732 accuracy.

Model	Accuracy	Precision	Recall
CNN	74.32	72.94	73.38
CNN and RNN	88.46	88.96	87.13

Table 1 validation Accuracy

Table 4.1. The suggested CNN-RNN combined approach's validation accuracy, precision, and recall are displayed in the table along with the fundamental CNN architecture. The suggested CNN-RNN hybrid architecture was assessed using several hyper parameters to considerably fine-tune the model and explore how the model may be improved under various conditions. The suggested CNN-RNN combination approach's ultimate outcome is the result of experiments utilizing various numbers of hidden units and hidden layers

.

No. of hidden units	**Accuracy**	**Precision**	**Recall**
25	88	87	86
50	92	89	87
75	95	91	86
100	98	90	88

Table 2: Accuracy of hidden units

Table 4.2. The validation Accuracy, Precision, and Recall of the proposed CNN-RNN combined technique based on various hidden units are displayed in the table.

The output of the suggested model with a varying number of hidden units was shown in Table 4.2. According to the study, the model produces the greatest results when 150 hidden units are used. The scores for recall, accuracy, and precision rise as the number of concealed units rises. However, the model starts acting unfavorably once there are more than 150 hidden units.

No. of hidden layers	**Accuracy**	**Precision**	**Recall**
3	88.03	87.88	86.09
4	87.67	87.64	87.97
5	88.15	87.22	86.20
6	89.37	86.16	88.22

Table 3: Accuracy of hidden layers

A Table 4.3. The validation Accuracy, Precision, and Recall of the proposed CNN-RNN combined technique based on various hidden layers are displayed in the table.

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:44:54 UTC from IEEE Xplore. Restrictions apply.

Similar to Table 4.3, which displays the results of using various hidden layers in the model, it is discovered that using six hidden layers produces the best classification results. The outcome of the architecture is decreased when there are seven hidden layers added. The experiment concludes that utilizing 150 hidden units and 6 hidden layers improves the performance of the suggested classification model.

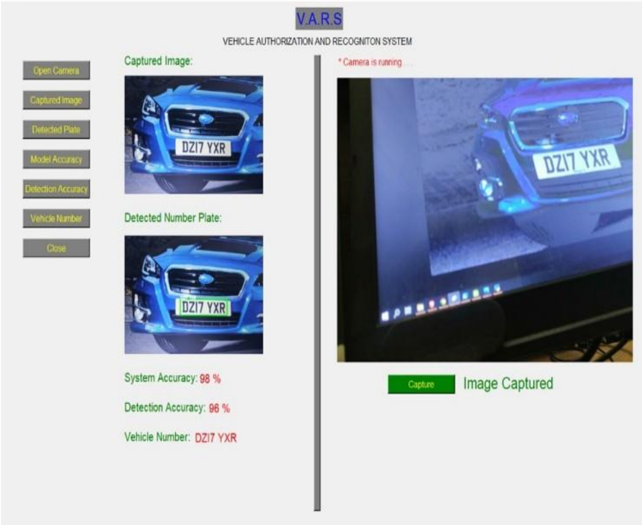

Fig:4.2: Front-end view

The CNN-RNN combination strategy to classify human anomalies was thoroughly explored in this work, and it was found to operate satisfactorily with 150 hidden units and 6 hidden layers.

V. Conclusion

Using our Kaggle source dataset, this study evaluates and tests a Vehicle Number Plate Acknowledgment Framework approach. To create and thoroughly test our method, we used a mixed CNN and RNN technique. We observe that thecombined CNN-RNN strategy provides improved execution for acknowledging vehicle number plates. To the best of our knowledge, the engineering described in this research is the RST architecture, which uses a cutting-edge CNN-RNN combined technique to classify vehicle number plates. Additionally, the proposed design's execution is contrasted with several CNN standard designs. From still images to the best of our knowledge, our proposal is the RST to ask about work on Vehicle Number Plate Acknowledgement frameworks. Our projects exclusively use image-based processes.In the future, we'll be working with high-level algorithms and conducting vehicle number plate detections.

A significant field study may be Vehicle Number Plate Location from Video and Sound Using Deep Learning Structures. In addition, CNN-RNN was used to develop the engineering, however more contemporaryengineering, such as GRU and LSTM, can be tried for the task. We wholeheartedly acknowledge that "Vehicle NumberPlate Recognition Utilizing Combined CNN-RNNApproach" may be a research topic that opens the door for critical investigation into vehicle number plate recognition and enhances the insights and applicability in further work.

References.

[1] Kumar Parasuraman, Member IEEE and Subin P.S. “SVM Based License Plate Recognition System” 2010 IEEE International Conference on Computational Intelligence and Computing Research

[2] P. Patil, C. Kanagasabapathi, and S. S. Yellampalli, “Automatic Number Plate Recognition System for vehicle identification,” *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2017

[3] A. Roy and D. P. Ghoshal, “Number Plate recognition for use in different countries using an improved segmentation,” *2011 2nd National Conference on Emerging Trends and Applications in Computer Science*, 2011.

[4] C. A. Rahman, W. Badawy, and A. Radmanesh, “A real time vehicle's license plate recognition system,” *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, 2003

[5] C. H. Munuo, K. Michae, and N. Nerey H., “Vehicle Plate Number Detection and Recognition Using Improved Algorithm,” *Computer Engineering and Intelligent Systems* , vol. Vol.5, No.10, 2014 , no. ISSN 2222-1719, 2014.

[6] S. Ozbay and E. Ercelebi , “Automatic Vehicle Identification by Plate Recognition,” *World Academy of Science, Engineering and Technology*, 2007.

[7] K. Deb, H.-U. Chae, and K.-H. o, “Vehicle License Plate Detection Method Based on Sliding Concentric Windows and Histogram,” *JOURNAL OF COMPUTERS,* vol. VOL. 4, NO. 8, no. AUGUST 2009.

[8] “Automatic number-plate recognition,” *Wikipedia*, 07-Jan-2023. [Online]. Available: https://en.wikipedia.org/wiki/Automatic_number-plate_recognition. [Accessed: 17-Feb-2023].

[9] A. Das, “Convolution neural network for image processing-using Keras,” *Medium*,11-Jan-2021.[Online].Available: https://towardsdatascience.com/convolution-neural-network-for-image-processing-using-keras-dc3429056306. [Accessed: 17-Feb-2023].

[10] “Recurrent neural networks - combination of RNN and CNN,” *Recurrent Neural Networks - Combination of RNN and CNN - Convolutional Neural Networks for Image and Video Processing - TUM Wiki*. [Online]. Available: https://wiki.tum.de/display/lfdv/Recurrent+Neural+Networks+-+Combination+of+RNN+and+CNN. [Accessed: 17-Feb-2023].

[11] P. Prabhakar and P. Anupama, “A novel design for Vehicle License Plate Detection And Recognition,” *Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014*, 2014.

[12] S.-L. Chen, C. Yang, J.-W. Ma, F. Chen, and X.-C. Yin, “Simultaneous end-to-end vehicle and license plate detection with multi-branch attention neural network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3686–3695, 2020.

[13] V. Mallikalava, S. Yuvaraj, K. Vengatesan, A. Kumar, S. Punjabi, and S. Samee, “Theft vehicle detection using image processing integrated digital signature based ECU,” *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020.

[14] J. I. Zong Chen, “Automatic vehicle license plate detection using K-means clustering algorithm and CNN,” *March 2021*, vol. 3, no. 1, pp. 15–23, 2021.

[15] “Automatic Number Plate Recognition (ANPR) | police.uk.” [Online]. Available: https://www.police.uk/advice/advice-and-information/rs/road-safety/automatic-number-plate-recognition-anpr/. [Accessed: 17-Feb-2023].

Authorized licensed use limited to: Dublin Business School. Downloaded on July 11,2024 at 13:44:54 UTC from IEEE Xplore. Restrictions apply.