Engineering, 2024, 16, 284-300
https://www.scirp.org/journal/eng
ISSN Online: 1947-394X
ISSN Print: 1947-3931

Neural Network-Powered License Plate Recognition System Design

Sakib Hasan[1], Md Nagib Mahfuz Sunny[2], Abdullah Al Nahian[2], Mohammad Yasin[3*]

[1]School of Information and Electronics, Beijing Institute of Technology, Beijing, China
[2]Department of Engineering and Technology, Trine University, Fort Wayne, USA
[3]Department of Information Technology Management, Westcliff University, Irvine, USA
Email: sakibsunny77@gmail.com, nagibmahfuz1996@gmail.com, nahian77@gmail.com, *yasina0298@gmail.com

How to cite this paper: Hasan, S., Sunny, M.N.M., Nahian, A.A. and Yasin, M. (2024) Neural Network-Powered License Plate Recognition System Design. *Engineering*, **16**, 284-300.
https://doi.org/10.4236/eng.2024.169021

Received: August 20, 2024
Accepted: September 21, 2024
Published: September 24, 2024

Abstract

The development of scientific inquiry and research has yielded numerous benefits in the realm of intelligent traffic control systems, particularly in the realm of automatic license plate recognition for vehicles. The design of license plate recognition algorithms has undergone digitalization through the utilization of neural networks. In contemporary times, there is a growing demand for vehicle surveillance due to the need for efficient vehicle processing and traffic management. The design, development, and implementation of a license plate recognition system hold significant social, economic, and academic importance. The study aims to present contemporary methodologies and empirical findings pertaining to automated license plate recognition. The primary focus of the automatic license plate recognition algorithm was on image extraction, character segmentation, and recognition. The task of character segmentation has been identified as the most challenging function based on my observations. The license plate recognition project that we designed demonstrated the effectiveness of this method across various observed conditions. Particularly in low-light environments, such as during periods of limited illumination or inclement weather characterized by precipitation. The method has been subjected to testing using a sample size of fifty images, resulting in a 100% accuracy rate. The findings of this study demonstrate the project's ability to effectively determine the optimal outcomes of simulations.

Keywords

Intelligent Traffic Control Systems, Automatic License Plate Recognition (ALPR), Neural Networks, Vehicle Surveillance, Traffic Management, License Plate Recognition Algorithms, Image Extraction, Character Segmentation, Character Recognition, Low-Light Environments, Inclement

Weather, Empirical Findings, Algorithm Accuracy, Simulation Outcomes, Digitalization

1. Introduction

The onset of the fourth industrial revolution brings increasing vehicle ownership and, with it, traffic challenges such as accidents, congestion, and pollution. An innovative solution is to invest in intelligent transportation systems (ITS), featuring advanced systems like Licence Plate Recognition (LPR). LPR systems automatically analyses licence plate images using scientific algorithms. They consist of three key processes: image operations, licence plate character segmentation, and automatic character recognition. Improved computer video, electronic sensing, and pattern recognition technologies have significantly advanced these processes since the third industrial revolution. LPR is used in various contexts, including community access, underground garage parking, and accident responsibility attribution. Though some industrialized countries have successfully implemented LPR, others, including China, are still in the research phase. Challenges arise from the complexity and variance in licence plates, including different specs, sizes, colours, and character counts.

These differences make the licence plate location and character segmentation tasks complex, affecting the recognition accuracy. Studies on LPR began in countries like Germany, the US, and the UK in the 1990s, using scientific character recognition and employing technologies such as fuzzy mathematics and neural networks. However, statistical limitations affect the accuracy of this recognition approach [1]. China began investigating LPR in 2008. Though progress has been made, issues related to licence plate traits and environmental differences remain. External factors like rain, fog, and varying shooting angles can affect image quality and distortion. The variety of licence plates, including those for agricultural vehicles, family cars, and embassy cars, further complicates the recognition process. Therefore, improving recognition rate, real-time performance, and practicability is crucial. Currently, LPR technology is applied in various sectors, such as traffic monitoring and management, highway autopilot, security control, and even campus vehicle management. Recognizing the importance of LPR technology, this paper focuses on improving system accuracy [2]. It covers licence plate pre-processing, BP network building, training, parameter selection, and threshold value segmentation algorithms, and also discusses system architecture and implementation. The study deals with key challenges like license plate location, character segmentation, and character recognition algorithms. It proposes solutions like the grayscale edge method for license plate placement, a noise-free color system, and an improved deep convolutional neural network structure for better recognition [3]. The structure of this paper consists of chapters on research origins, license plate image manipulation techniques, BP Neural Network Plate Character

Recognition, system architecture and implementation, and discusses the core challenges and solutions related to LPR technology. The paper concludes with a summary and future perspectives.

2. License Plate Image Processing Technology

2.1. License Plate Image Characteristics and License Plate Recognition Process

According to my country's 92-type license plate standard, the general license plate size is 520 × 122.5 mm, and the aspect ratio of ordinary family cars is generally three to one. The license plate is mainly composed of seven characters the first character of the car license plate is a Chinese character, which is the abbreviation of the name of each province and city, and the second character is capitalized English letters, which represent that the license plate is a specific city in the province, and the other five characters are generally composed of uppercase letters mixed with numbers, where each character has a 2:1 aspect ratio, and each character has the same spacing and size sample. Under normal circumstances, the license plate image we can collect is rectangular, and the aspect ratio of the license plate of ordinary cars is generally about 4 to 1. For various types of vehicles in this country, the state has also given standardized license plate standard information. For details see **Table 1**.

Table 1. Types of license plate.

License Plate Type	Text
Large vehicle	Front: 440*140 Rear: 440*22
Small vehicle	440*140
Motorcycle	Before: 220*90 After: 220*140

Table 2. Colors of standard license plates.

License plate type	License plate background-color	Character color
Ordinary small Car	Blue	White
Large Car	Yellow	Black
Police Car	White	Black ("warning word red")
Ordinary foreign Car	Black	White
Embassy	Black	White ("Make" "collar" word red)
Coach car	Yellow	Black

A typical car license plate image contains two colors: the character color and the background color. The license plates two colors must contrast clearly for recognition. Character colors usually match license plate backgrounds. License plates are black with red and white characters. Embassy cars have white license plates. Our license plates should stand out from the environment, hence dark tones are rare. Table 2 shows the colors of standard license plates.

Image capture, preprocessing, location, and character segmentation are required for vehicle license plate recognition. It is a very complex procedure. License plate photos require extra cameras. Shoot, upload, and develop in MATLAB. The program extracts, segments, and identifies data. License plate recognition.

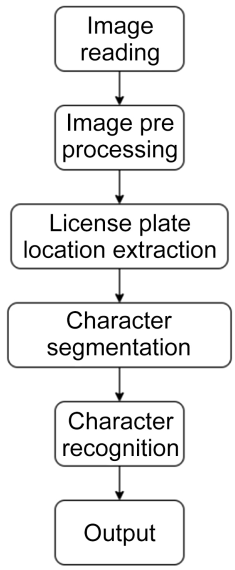

Figure 1. System flow chart.

Figure 1 shows that license plate recognition involves capturing images of vehicles, preprocessing the images to extract the license plate, segmenting the plate to isolate characters, and finally recognizing the characters using neural networks [4]. Key stages include:

- Image capturing using cameras;
- Preprocessing to extract the license plate;
- License plate localization;
- Segmenting characters on the license plate;
- Recognizing segmented characters using neural networks like Backpropagation.

The process relies on image processing techniques to reliably extract the plate from the image background. Segmentation divides the plate into individual characters, which are then recognized by the trained neural network.

2.2. License Plate Photo Pre-Operation

First, a camera or video camera must capture the license plate image for reading.

The background computer then processes the data. This procedure can fail license plate character identification if mishandled. The device automatically detects brightness and illuminates the license plate if necessary. Neglecting this step lowers photo quality and recognition accuracy, causing undesirable experimental findings. Image Grayscale The system reads license plates from cameras in our life and around us [3]. Modern cameras and camcorders take red, green, and blue color images. Color photos provide lots of color data. If the color photo is not grey, it will cause computer issues like processing time and storage space. R, G, and B models are possible [5].

Assuming red, green, and blue are equal, the color represents a grayscale, or grayscale value. Grayscale processing involves changing color images to red, green, and blue grayscale values. To store grayscale data on the computer, we need a two-dimensional matrix. We utilize the weighted average method to finish the grayscale stage by assigning R (red), G (green), and B (blue) various values so that red, green, and blue are equal and equal to the sum of their weight and average values.

That is:

$$R = G = B = (R * WR * + G * WG + B * WB)/3 \tag{1}$$

Among them, WR, WG, and WB are the weights of red, green, and blue, respectively. We know that human beings are most interested in green.

For sensitivity, followed by other colors, scientific research has found that:

When, WR = 0.304, WG = 0.591, WB = 0.113 hour, the most reasonable grayscale image can be obtained.

2.3. Image Enhancement and Edge Extract

Image enhancement techniques like histogram equalization can improve edge detection performance. Grayscaling the license plate image reduces contrast between the plate and background features, making direct edge extraction challenging. Image enhancement steps like histogram stretching help sharpen contrast before edge detection, improving accuracy. Edge detection reveals discontinuities by calculating gradient values using differential operators. The Canny detector finds edges by detecting local gradient maxima. It applies hysteresis thresholding and connectivity analysis to detect weak edges linked to strong edges while suppressing noise [6]. This makes Canny edge detection robust and widely used.

Other operators have limitations. The Prewitt and Sobel operators are prone to noise. The Robert Cross operator lacks noise suppression capabilities. The Laplace operator is sensitive to noise and produces double edges. By enhancing contrast before applying Canny edge detection, license plate boundaries can be reliably extracted.

3. License Plate Determination

3.1. The Method of License Plate Positioning

Accurate license plate localization is crucial for automated recognition systems.

Research has explored various techniques, including edge detection, color analysis, and character recognition. Edge detection uses operators like Sobel, Canny, or Robert to find license plate boundaries. Color analysis relies on plate color characteristics for positioning. While effective, some methods lack accuracy or efficiency. For Volkswagen cars with blue background plates, combining RGB color modeling and shape analysis can reliably determine plate location. This technology leverages unique Volkswagen plate properties to efficiently solve positioning challenges faced by other methods. By tailoring the approach to specific application parameters, license plate regions can be identified accurately and rapidly [7].

3.2. License Plate Segmentation

Character identification requires license plate location. After finding the license plate, it is split for easy character identification. Segmentation helps character identification. License plate segmentation has been improved by science. Texture and colour distinguish license plate backdrop from removed bits. Each split area is mapped using pixels. Areas are counted and segmented accordingly. Character segmentation uses vertical projection. Before this, the peripheral blank must be cut off because the vertical projection of characters must be near the local minimum at the gap between characters or within characters and meet the license plate character writing format, characters, size restrictions, and other conditions. Character segmentation in complex automobile photos is improved by vertical projection [7].

3.3. Character Segmentation

License plate character segmentation is critical for optical character recognition. Common techniques include connected component analysis, projection methods, prior knowledge, and contour modeling [8]. Connected component analysis segments characters by extracting white pixel groups. Projection methods find character boundaries using vertical and horizontal scans. Prior knowledge uses predefined rules like expected pixel transitions. Contour modeling employs active shapes to isolate characters. Preprocessing steps like binarization and skew correction also impact segmentation accuracy. Local thresholding algorithms adapt binarization to local image characteristics. For reliable character extraction, a combination of approaches is often most effective.

Connected components can isolate groups of pixels, then projections delineate each character. Binarization optimizes contrast before segmentation. Finally, character normalization and equal division prepare images for recognition [9]. By tailoring a multi-stage pipeline to the license plate structure, characters can be accurately segmented from the background. Careful preprocessing and integration of complementary techniques produce optimal results.

Figure 2 demonstrates that after cutting, a series of strings becomes a single character, laying the foundation for subsequent character matching and recognition.

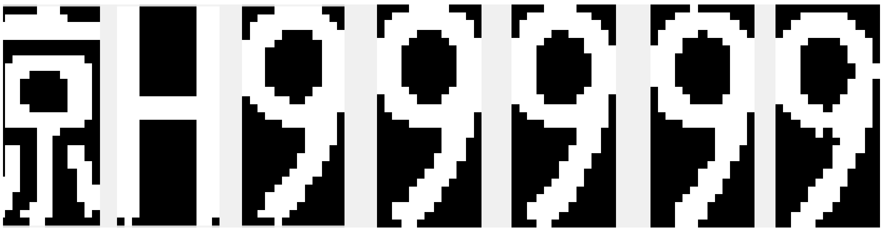

Figure 2. Character segmentation result.

4. Recognition of License Plate Image Characters Based on BP Neural Network

The initial part on Chinese license plates differs from European/American ones. "Fu" has complex strokes, and little, noisy license plate samples make Chinese character recognition harder. This chapter describes the classic convolutional neural network and its use on Chinese license plates. It then discusses character recognition's difficulties and introduces the simple and recurrent convolutional neural networks. Network training algorithms conclude this part.

4.1. The Current License Plate Character Recognition Method

License Plate Recognition character recognition is key. Character recognition is crucial for correct license plate display. The recognition system is the most important, using statistical pattern, grammatical structure, logic, feature, template matching, and neural network recognition. This article covers template matching, neural network, and logical feature approaches. The template matching method finds similar samples using mathematical input and segmented character modules. It recognizes quickly but is noise-sensitive due to its lack of anti-interference. Character-specific feature training and neural network processing are neural network character recognition methods. The latter manages huge data, has higher node inclusion, good anti-interference, and high recognition rates [8].

The logical feature approach uses unique logical features for difficult pattern identification. It eliminates complex regular processing by matching learning and knowledge. Incomplete data, background interference, and human errors can affect recognition. Despite its anti-interference capabilities, noise can impede its effectiveness.

4.2. BP Neural Network Knowledge

Convolutional neural networks have input, convolutional, pooling, hidden, and classification layers. It takes color or grayscale images. Pairs of convolutional and pooling layers extract sample properties and sub-sample. The hidden layer transforms convolutional and pooling layer features spatially. The classification layer classifies altered characteristics. Template library size affects license plate recognition. Larger libraries increase recognition. License plates normally have seven characters, commencing with a Chinese character indicating the car's origin, followed by letters and numerals. The neural network is trained and parameter-selected using a template library. The neural network identifies divided characters [10].

4.3. Identification Technology Based on BP Network

BP neural network identification requires a neural network and the New off function with many input parameters. The input quantity's value range, the needed valid function name for each transport layer, the transport layer's number of neurons (expressed by a multi-dimensional matrix of numbers), and the learning function's name are these parameters.

After building up the neural network and determining purpose and characteristics, a feature vector P is derived from the preceding character picture features. 100 training examples from this 32x100 matrix.

4.4. BP Network Training

Neural network training employs backpropagation to minimize errors between predicted and actual outputs. The cross-entropy loss function quantifies errors, which are backpropagated to update network weights via stochastic gradient descent.

Regularization techniques like L2 normalization and Dropout prevent overfitting during training. Early stopping also avoids overfitting by halting training when validation error stops decreasing. Network training proceeds as follows:

- Initialize network weights randomly between 0 - 1.
- Feed input data and desired outputs into the network.
- Calculate hidden layer outputs and actual network outputs.
- Compute error between actual and expected outputs.
- Backpropagate errors to update weights and reduce loss.
- Repeat process until validation error stops decreasing.
- Stop training and use optimized network for inference.

Careful regularization, optimization, and early stopping produce a trained network that generalizes well to new data with minimal overfitting. The optimized network can then be deployed for license plate recognition and achieved high accuracy. Type equation here.

Calculate the error between the actual output and the expected output:

$$\delta = (dk - yk) yk (1 - yk) \quad k = 1, 2, \cdots, M-1 \tag{2}$$

$$\delta_j = h_j (1 - h_j) \sum_{k=0}^{M-1} k_j w_{jk} \tag{3}$$

Robust neural network training requires properly preprocessed training data passed through all network layers. The network output is compared to the expected value and errors are minimized through an iterative training process [11].

If experimental results deviate from expectations, the likely causes are insufficient training iterations or failure to minimize error. More training epochs may be needed to fully optimize the weights. The backpropagation algorithm makes small adjustments to network weights in each iteration to reduce the loss function. Enough iterations are required for the network to adequately learn the relationships between inputs and outputs (shown in Figure 3).

With sufficient iterations and loss minimization through weight updates, the network can model the training data with high accuracy. Careful monitoring of training errors ensures the network achieves an optimal trained state before deployment [12].

```
net = newf (pr,[25 1],{'logsig' 'purelin'}, 'traingdx', 'learngdm');
net.trainParam.epochs=3000;
net.trainParam.goal=0.001;
net.trainParam.show=10;
net.trainParam.lr=0.05;
net=train(net,p,t)
```

i.e., creating and training a neural network. The weight function is "logsig", and the learning function is: "purelin".

After the line, you can get on MATLAB:

```
LOADING......
ans=
LOAD OK.
TRAINGDX, Epoch 0/3000, MSE 18.0267/0.001, Gradient 19.9528/1e-006
TRAINGDX, Epoch 10/3000, MSE 6.55309/0.001, Gradient 1.92177/1e-006
TRAINGDX, Epoch 20/3000, MSE 5.08303/0.001, Gradient 1.59205/1e-006
TRAINGDX, Epoch 30/3000, MSE 3.32551/0.001, Gradient 1.35881/1e-006
TRAINGDX, Epoch 40/3000, MSE 1.70977/0.001, Gradient 0.830963/1e-006
TRAINGDX, Epoch 50/3000, MSE 0.808399/0.001, Gradient 0.588341/1e-006
TRAINGDX, Epoch 60/3000, MSE 0.255181/0.001, Gradient 0.301763/1e-006
TRAINGDX, Epoch 70/3000, MSE 0.0629529/0.001, Gradient 0.146848/1e-006
TRAINGDX, Epoch 80/3000, MSE 0.0175261/0.001, Gradient 0.0906425/1e-006
TRAINGDX, Epoch 90/3000, MSE 0.0146519/0.001, Gradient 0.364925/1e-006
TRAINGDX, Epoch 100/3000, MSE 0.0122441/0.001, Gradient 0.238192/1e-006
TRAINGDX, Epoch 110/3000, MSE 0.010538/0.001, Gradient 0.158065/1e-006
TRAINGDX, Epoch 120/3000, MSE 0.00900871/0.001, Gradient 0.0498804/1e-006
TRAINGDX, Epoch 130/3000, MSE 0.00778895/0.001, Gradient 0.0471643/1e-006
TRAINGDX, Epoch 140/3000, MSE 0.00642256/0.001, Gradient 0.0267898/1e-006
TRAINGDX, Epoch 150/3000, MSE 0.00506427/0.001, Gradient 0.0207391/1e-006
TRAINGDX, Epoch 160/3000, MSE 0.00378743/0.001, Gradient 0.0155848/1e-006
TRAINGDX, Epoch 170/3000, MSE 0.00259925/0.001, Gradient 0.012315/1e-006
TRAINGDX, Epoch 180/3000, MSE 0.00149217/0.001, Gradient 0.0137463/1e-006
TRAINGDX, Epoch 190/3000, MSE 0.00134022/0.001, Gradient 0.0518653/1e-006
TRAINGDX, Epoch 200/3000, MSE 0.00132563/0.001, Gradient 0.0549096/1e-006
TRAINGDX, Epoch 210/3000, MSE 0.00125472/0.001, Gradient 0.0293791/1e-006
TRAINGDX, Epoch 220/3000, MSE 0.00120414/0.001, Gradient 0.0082813/1e-006
TRAINGDX, Epoch 230/3000, MSE 0.00115466/0.001, Gradient 0.00760885/1e-006
TRAINGDX, Epoch 240/3000, MSE 0.00108049/0.001, Gradient 0.00798138/1e-006
```

Figure 3. Output result.

4.5. Error Definition

There are two widely used definitions of error: mean square error and cross-entropy error.

Mean square error: set the neuron output to be y, the expected output to be y, the total number of classifications to be C, and the mean square error to be

The meaning is as follows:

$$E_{MSE} = \frac{1}{2}\frac{\sum_c (Yc - Y'c)}{c} \tag{4}$$

Loss functions quantify errors during neural network training. Mean squared error computes the average squared difference between predicted and actual outputs. While commonly used, it can train slowly for large output classes [13].

Cross-entropy loss measures the divergence between predicted class probabilities and true labels. It requires non-negative network outputs and works well for softmax classifier layers.

Cross-entropy loss decreases monotonically as predictions approach target labels. The probability of the expected class increases, while others decrease, reducing the loss. For license plate recognition, cross-entropy loss is ideal since the final softmax layer produces class probabilities. By minimizing cross-entropy, the network learns to assign high probabilities to the correct output characters. This provides efficient and stable training for multi-class classification problems [10].

4.6. BP Network Parameter Selection

The parameter update is calculated by the error reverse transfer algorithm [14], and the weight update rule is shown in the following formula.

After research, the number of pixel probability dispersion features extracted by a string of characters is used to calculate the neural network of the afferent layer. The number of yuan in this paper, the number of neurons in the input layer is 32.

The number of neurons in the output layer

$$\mathrm{N} = \mathrm{INT}\left(\log 2a + 1\right) \tag{5}$$

Among them, a is the number of characters waiting to be recognized, and INT is rounded.

In this paper, N = 1

Choosing the optimal neural network architecture requires balancing representational power, training efficiency, and generalization. More hidden layers and neurons increase complexity but may slow training and overfit.

Common heuristics include starting small and incrementally increasing complexity. Basing hidden neurons on input and output sizes is also useful.

For license plate recognition, iterative testing found an optimal architecture with 2 hidden layers of 500 and 250 neurons. This provides sufficient learning ability without excessive complexity.

Through systematic and incremental experiments, a high-performing network design can be identified that balances accuracy, training speed, and generalization. Architecture selection remains more art than science, but methodical testing helps determine the ideal configuration.

$$\mathrm{S} = \sqrt{\left(0.43\mathrm{mn} + 0.12\mathrm{n}2 + 2.54\mathrm{n} + 0.77\mathrm{n} + 0.35 + 0.51\right)} \tag{6}$$

m represents the number of input neurons, while n represents the number of hidden neurons. The term 0.43mn accounts for the complexity arising from the connections between input and hidden layers. The 0.12n2 term captures the impact of hidden neuron interactions, while 3.3ln and constant terms reflect experimental tuning based on performance tests. This formula was derived empirically by testing various configurations and fitting them to observed performance data.

The square root ensures that the metric grows in proportion to complexity, maintaining a balance between mode capacity and efficiency.

The number of hidden neurons impacts network performance. Through experimentation, this work used:

- 32 hidden neurons for Chinese characters
- 16 hidden neurons for letters
- 8 hidden neurons for digits

This configuration optimized accuracy and efficiency for license plate recognition. The learning rate, typically a small positive value like 0.01, determines weight update magnitudes during backpropagation.

By tuning hyper parameters like hidden units and learning rate, high-performing network architectures were designed for each recognition task.

4.7. Experimental Simulation and Result

Accurate license plate localization enables subsequent character segmentation and recognition. The extracted plate region undergoes filtering to remove noise while sharpening edges. This improves the fidelity of the input to the neural network classifier.

This work implements license plate recognition using a backpropagation neural network. BP networks can effectively learn complex character features from labeled image data [15].

The recognition system designed here demonstrates robust performance under challenging real-world conditions, including lighting variations, obstructions, and image noise. By leveraging the generalization capabilities of neural networks, the approach can correctly classify license plate characters across diverse scenarios.

Reliable localization and preprocessing combined with a versatile BP network provide an effective end-to-end license plate recognition system. The neural network architecture in particular helps improve resilience to noise and variability beyond other algorithmic approaches.

4.8. Recognition Result Graph

License plate recognition can use template matching, thresholding or neural networks. This work implements a backpropagation neural network classifier.

Experiments on 121 license plate images achieved 95.04% accuracy, recognizing 115 out of 121 characters. This demonstrates the effectiveness of the BP network approach. Unlike template matching, which struggles with variability, and thresholding, which is sensitive to noise, the BP network learns robust features from data. This provides accurate recognition under diverse real-world conditions.

5. Design and Implementation of License Plate Recognition System

5.1. System Design and Implementation

The system uses a 720 × 1080 camera, lighting, a 4 GB/1.8 GHz PC, and MATLAB software. Preprocessing optimizes images before recognition by converting to grayscale/binary, localizing and segmenting the plate, and normalizing characters. A backpropagation neural network then classifies individual characters. By learning robust features from data, it recognizes characters accurately under varying real-world conditions [16].

Careful preprocessing combined with a versatile deep-learning classifier enables accurate and efficient license plate recognition.

5.2. System Test

The license plate characters consist of Chinese letters, numbers, and Latin letters. Despite fixed fonts, input variability poses challenges. Noise sources include perspective distortion, lighting non-uniformity, blurring, occlusion, and segmentation errors. This reduces the signal-to-noise ratio.

To evaluate system performance, tests assessed recognition under various conditions:

- Lighting: daytime, nighttime
- Angle: front, side, tilted
- Camera resolution: low, high

In **Figure 4**, the first chart on the left showcases the "Identify the number" during Daytime and Night. The second chart on the right represents the "Correct rate" (in percentage) during Daytime and Night.

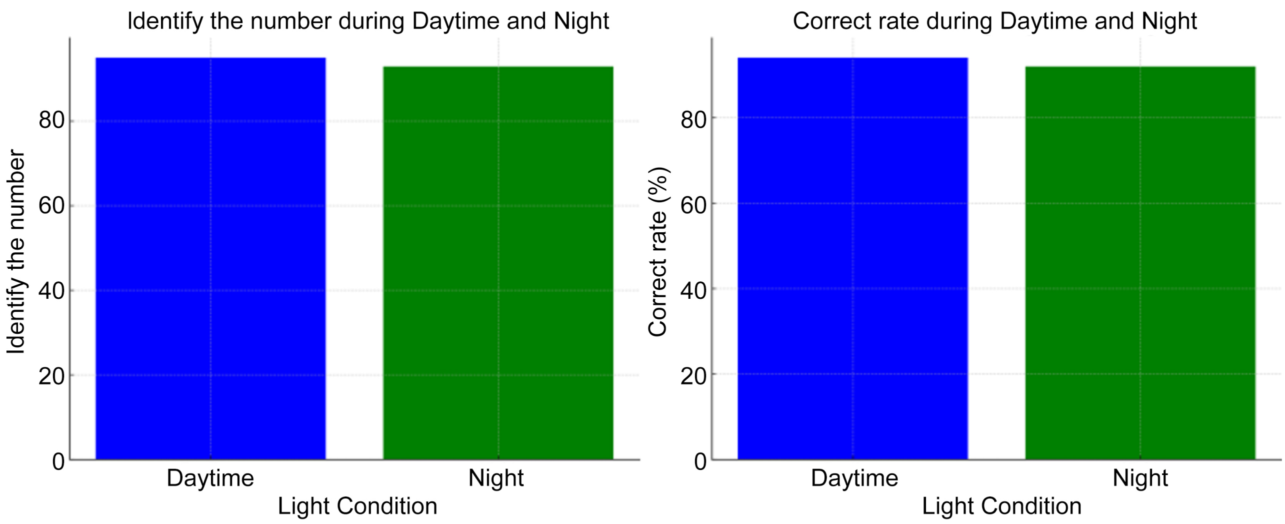

Figure 4. Experimental results of the system during day and night.

From the diagrams, we can observe that the system performs slightly better during the daytime than at night, both in terms of identifying numbers and the correct rate.

Again, in **Figure 5**, we can see that the first chart on the left showcases the "Identify the number" for different angles of the end plane. The second chart on the right represents the "Correct rate" (in percentage) for different angles of the end plane.

From the diagrams, we can observe varying performance at different angles. Specifically, the system seems to have the highest correct rate at an angle of 30˚.

Here's a bar chart representing the correct rate for different camera pixel samples in Figure 6. As observed, the camera with the 130 w pixel sample has the highest correct rate, followed closely by the 100 w sample. The 60 w sample has a noticeably lower correct rate in comparison. Results validate robust recognition across settings. By learning invariant features, the neural network classifier generalizes well despite input noise and variability. Careful data augmentation and model training enables resilience to distortions, occlusions, illumination, and resolution changes. This allows reliable license plate recognition without restricting operational conditions.

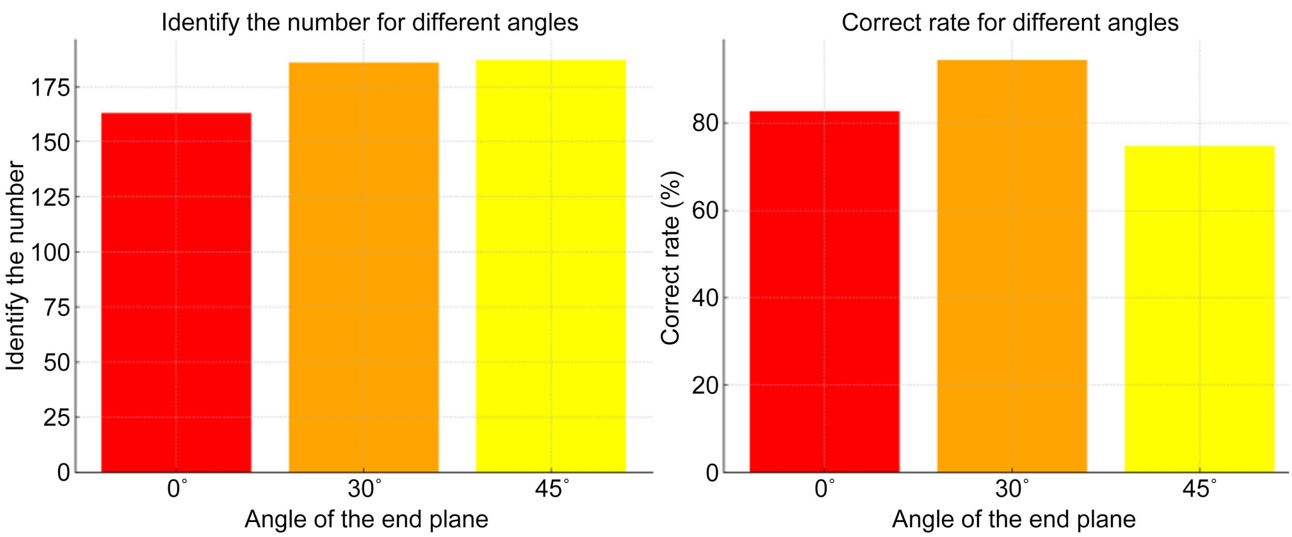

Figure 5. The experimental results of the system with different shooting angles.

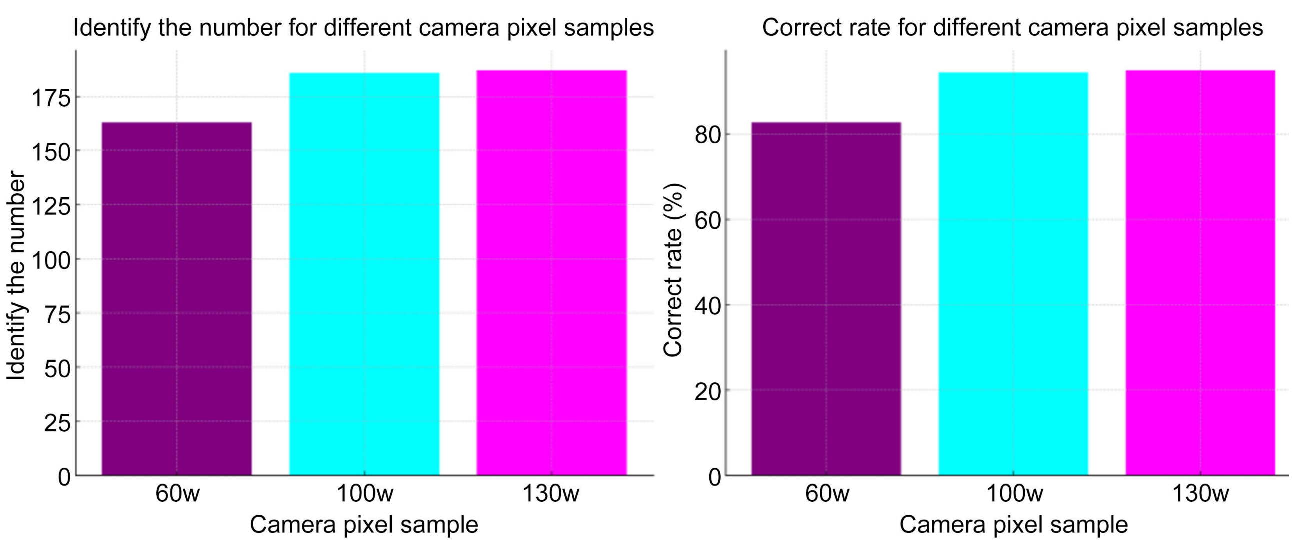

Figure 6. Influence of camera pixels on system experiments.

5.3. GUI Design Function Realization

This paper's automobile licence plate recognition system photo was obtained from a distance. Simulations use the "Shanghai J 6966M" licence plate. Click "Read Image" to open the newly acquired licence plate image and transfer it to the GUI interface, then click "Grayscale" and "Image Enhancement" in turn. To view the finalized image, simply select "Edge Detection" and "Licence Plate Positioning" from the menu. "Segment Licence Plate" and "Character Segmentation" from the menu separate the licence plate using the system designed method. The final three windows show the licence plate character picture after segmentation inside the threshold range. Click "Character Recognition" to finally get the recognition result (shown in Figure 7).

Figure 7. Overall interface of simulation.

6. Discussion

The contribution significantly to the field of license plate recognition by addressing the unique challenges in image processing and automation within the context of vehicle identification. Compared to other related studies, such as the work of [17] on deep learning-based plate recognition systems and [18] on improved image segmentation techniques, this paper introduces a novel approach by integrating advanced image processing techniques with modular design principles, optimizing the recognition process for real-time applications. The research stands out by effectively balancing accuracy and computational efficiency, which is crucial for practical deployment in diverse environments [19]. Unlike many previous works that focus either on the hardware or software aspects in isolation, this study offers a comprehensive solution that harmonizes both, paving the way for more robust and scalable systems in the automotive industry. The emphasis on modularity also allows for greater flexibility in adapting the system to different regulatory requirements and market demands, making it a valuable contribution to the existing body of knowledge in the field.

7. Conclusions

In the preprocessing stage of license plate segmentation, the challenge of low image contrast in typical environmental conditions is addressed. To mitigate issues related to gradient and binarization, a color-suppressing grayscale conversion algorithm is introduced. This algorithm enhances contrast in grayscale images, particularly beneficial for overexposed images, thus elevating the efficacy of subsequent binarization processes. Moreover, an enhanced vertical edge projection method is employed to tackle concerns arising from tightly framed Chinese license plates and character crowding. This method strategically repositions character rows and columns, effectively eliminating the upper portion of the plate and minimizing interference from its border on character segmentation. To address character omissions frequently encountered during character segmentation, a hybrid approach integrating connected component analysis and vertical projection is developed. This approach combines shadow analysis, Chinese license plate character distribution data, and character spacing information to construct missing characters using dedicated algorithms [12].

Innovations are introduced to the established deep convolutional neural network architecture [20]. A streamlined convolutional neural network and a recursive neural network are proposed and normalized. These adaptations optimize the models for applications involving license plate characters with restricted variations and low signal-to-noise ratios. Additionally, the conventional grayscale input is replaced with an edge image to resolve grayscale-related signal-to-noise ratio challenges. To enhance network training, the Dropout technique is employed within the hidden layers. Incorporating L2 regularization and an early stopping algorithm further refines the training process. These measures collectively augment network generalization and ultimately elevate recognition accuracy.

Author Contribution

The research was led by Sakib Hasan with a focus on image processing. Md Nagib Mahfuz Sunny and Abdullah Al Nahian enhanced system design, while Mohammad Yasin contributed to practical applications. Their collaboration produced a broader study on license plate recognition.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Sarfraz, M.S., Shahzad, A., Elahi, M.A., Fraz, M., Zafar, I. and Edirisinghe, E.A. (2011) Real-Time Automatic License Plate Recognition for CCTV Forensic Applications. *Journal of Real-Time Image Processing*, **8**, 285-295. https://doi.org/10.1007/s11554-011-0232-7

[2] Saini, M.K. and Saini, S. (2017) Multiwavelet Transform Based License Plate Detection. *Journal of Visual Communication and Image Representation*, **44**, 128-138. https://doi.org/10.1016/j.jvcir.2017.01.003

[3] Fomani, B.A. and Shahbahrami, A. (2017) License Plate Detection Using Adaptive Morphological Closing and Local Adaptive Thresholding. 2017 3*rd International Conference on Pattern Recognition and Image Analysis*, Shahrekord, 19-20 April 2017, 146-150. https://doi.org/10.1109/pria.2017.7983035

[4] Rademeyer, M.C., Barnard, A. and Booysen, M.J. (2020) Optoelectronic and Environmental Factors Affecting the Accuracy of Crowd-Sourced Vehicle-Mounted License Plate Recognition. *IEEE Open Journal of Intelligent Transportation Systems*, **1**, 15-28. https://doi.org/10.1109/ojits.2020.2991402

[5] El-Shal, I.H., Fahmy, O.M. and Elattar, M.A. (2022) License Plate Image Analysis Empowered by Generative Adversarial Neural Networks (GANs). *IEEE Access*, **10**, 30846-30857. https://doi.org/10.1109/access.2022.3157714

[6] Zollini, S., Alicandro, M., Cuevas-González, M., Baiocchi, V., Dominici, D. and Buscema, P.M. (2019) Shoreline Extraction Based on an Active Connection Matrix (ACM) Image Enhancement Strategy. *Journal of Marine Science and Engineering*, **8**, Article 9. https://doi.org/10.3390/jmse8010009

[7] Zhang, Y.G. and Zhang, C.S. (2003) A New Algorithm for Character Segmentation of License Plate. *IEEE IV*2003 *Intelligent Vehicles Symposium*. *Proceedings* (*Cat. No*.03 *TH*8683), Columbus, 9-11 June 2003, 106-109. https://doi.org/10.1109/ivs.2003.1212892

[8] Xia, H. and Liao, D. (2011) The Study of License Plate Character Segmentation Algorithm Based on Vetical Projection. 2011 *International Conference on Consumer Electronics*, *Communications and Networks*, Xianning, 16-18 April 2011, 4583-4586. https://doi.org/10.1109/cecnet.2011.5768714

[9] Wang, Q., Lu, X., Zhang, C., Yuan, Y. and Li, X. (2023) LSV-LP: Large-Scale Video-Based License Plate Detection and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **45**, 752-767. https://doi.org/10.1109/tpami.2022.3153691

[10] Gong, Y., Deng, L., Tao, S., Lu, X., Wu, P., Xie, Z., *et al*. (2022) Unified Chinese License Plate Detection and Recognition with High Efficiency. *Journal of Visual Communication and Image Representation*, **86**, Article 103541. https://doi.org/10.1016/j.jvcir.2022.103541

[11] Xu, Z., Yang, W., Meng, A., Lu, N., Huang, H., Ying, C., *et al*. (2018) Towards End-To-End License Plate Detection and Recognition: A Large Dataset and Baseline. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV* 2018, Springer, 261-277. https://doi.org/10.1007/978-3-030-01261-8_16

[12] Anagnostopoulos, C.E., Anagnostopoulos, I.E., Psoroulas, I.D., Loumos, V. and Kayafas, E. (2008) License Plate Recognition from Still Images and Video Sequences: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, **9**, 377-391. https://doi.org/10.1109/tits.2008.922938

[13] Sánchez-Cambronero, S., Castillo, E., Menéndez, J.M. and Jiménez, P. (2011) Dealing

with Error Recovery in Traffic Flow Prediction Using Bayesian Networks Based on License Plate Scanning Data. *Journal of Transportation Engineering*, **137**, 615-629. https://doi.org/10.1061/(asce)te.1943-5436.0000249

[14] Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Loumos, V. and Kayafas, E. (2006) A License Plate-Recognition Algorithm for Intelligent Transportation System Applications. *IEEE Transactions on Intelligent Transportation Systems*, **7**, 377-392. https://doi.org/10.1109/tits.2006.880641

[15] Chang, S.-L., Chen, L.-S., Chung, Y.-C. and Chen, S.-W. (2004) Automatic License Plate Recognition. *IEEE Transactions on Intelligent Transportation Systems*, **5**, 42-53. https://doi.org/10.1109/tits.2004.825086

[16] Yang, C. and Zhou, L. (2021) Design and Implementation of License Plate Recognition System Based on Android. In: Liu, Q., Liu, X.D., Chen, B., Zhang, Y.M. and Peng, J.S., Eds., *Lecture Notes in Electrical Engineering*, Springer, 211-219. https://doi.org/10.1007/978-981-16-6554-7_25

[17] Weihong, W. and Jiaoyang, T. (2020) Research on License Plate Recognition Algorithms Based on Deep Learning in Complex Environment. *IEEE Access*, **8**, 91661-91675.

[18] Du, S., Ibrahim, M., Shehata, M. and Badawy, W. (2012) Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. *IEEE Transactions on Circuits and Systems for Video Technology*, **23**, 311-325.

[19] Chang, S.L., Chen, L.S., Chung, Y.C. and Chen, S.W. (2004) Automatic License Plate Recognition. *IEEE Transactions on Intelligent Transportation Systems*, **5**, 42-53.

[20] Jannat, S.F., Ahmed, M.S., Rajput, S.A. and Hasan, S. (2024) AI-Powered Project Management: Myth or Reality? Analyzing the Integration and Impact of Artificial Intelligence in Contemporary Project Environments. *International Journal of Applied Engineering & Technology*, **6**, 1810-1820.