iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70511

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Number Plate Detection Using CNN

Sankalp Singh Rawat[1], Dr. ArchanaKumar[2]
Artificial Intelligence and Data Science, ADGIPS, GGSIPU

Abstract: ***This paper presents a deep learning- based approach for automatic number plate detection using Convolutional Neural Networks (CNN). The system focuses on accurately identifying and localizing vehicle license plates in real-time from images captured under various conditions. A custom CNN model is trained on a diverse dataset to extract spatial features and detect number plates effectively. The proposed method demonstrates high accuracy, robustness, and practical applicability in traffic monitoring and intelligent transportation systems.***

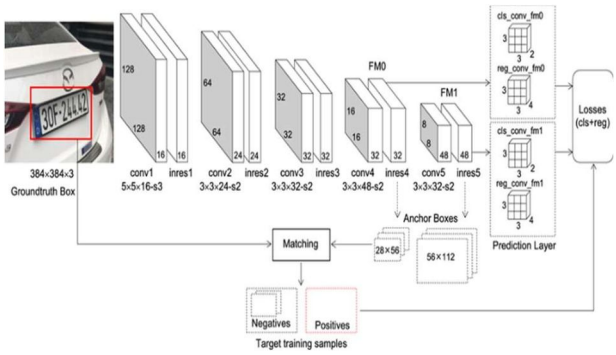
I. INTRODUCTION

In recent years, the demand for intelligent transportation systems has significantly increased, especially in areas such as traffic monitoring, law enforcement, toll collection, and automated vehicle tracking. A core component of these systems is Automatic Number Plate Recognition (ANPR), which enables the identification of vehicles based on their license plates. Traditional ANPR methods often rely on manual feature extraction and rule-based algorithms, which are sensitive to lighting conditions, angle variations, and background noise.

To overcome these challenges, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have gained popularity due to their powerful image recognition capabilities.

CNNs automatically learn hierarchical feature representations, making them well-suited for tasks like object detection and localization. This research focuses on developing a robust and efficient number plate detection system using CNN, capable of accurately identifying license plates in real-world scenarios. The proposed approach aims to improve detection accuracy and reliability, contributing to the advancement of automated vehicle identification systems.

II. CNN ARCHITECTURE



A Convolutional Neural Network (CNN) is a class of deep learning models specifically designed for analyzing visual data. It mimics the behavior of the human visual cortex and is widely used for image classification, object detection, and pattern recognition. CNNs are composed of several layers, each with a specific role in feature extraction and decision-making. The typical architecture includes the following key components:

1) Input Layer

This layer receives the raw image data in the form of pixel values (e.g., 64x64x3 for a color image). The input layer passes the data to the next layer without processing.

2) Convolutional Layers

These layers apply a set of filters (also called kernels) to the input image to extract local features such as edges, textures, and patterns. Each filter slides over the image to produce a feature map. These feature maps are then used to detect complex patterns in deeper layers.

3) Activation Function (ReLU)

After each convolution operation, a Rectified Linear Unit (ReLU) function is applied to introduce non-linearity into the model. This allows the network to learn complex patterns beyond linear relationships.

4) Pooling Layers (Subsampling)

Pooling layers reduce the spatial dimensions of feature maps, which helps decrease computational complexity and prevents overfitting. The most common method is Max Pooling, which selects the maximum value in each window.

5) Flattening Layer

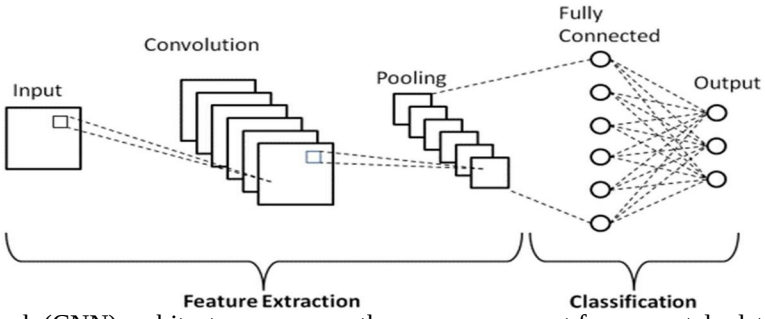
The output from the final pooling layer is flattened into a one-dimensional array. This array is then passed to the fully connected layers for classification or regression tasks.

6) Fully Connected (Dense) Layers

These layers are similar to a traditional neural network, where each neuron is connected to every neuron in the previous layer. They combine the features learned in earlier layers and output the final prediction.

7) Output Layer

The final layer contains neurons equal to the number of output classes. For classification tasks, a Softmax activation function is used to provide probabilities for each class.



The Convolutional Neural Network (CNN) architecture serves as the core component for accurately detecting vehicle number plates from images or video frames. CNNs are highly effective in processing visual data due to their layered structure, which enables automatic learning of spatial hierarchies of features. In this system, the CNN model is trained on a dataset of vehicle images to identify key characteristics of number plates, such as shape, edges, and textures. The convolutional and pooling layers progressively extract features, allowing the network to distinguish number plates from other regions within the image. Once the number plate is detected, it is extracted and passed to an Optical Character Recognition (OCR) module for character recognition. The use of CNN not only enhances the accuracy and efficiency of number plate detection but also enables real-time processing, making it suitable for intelligent transport applications such as automated toll systems, traffic monitoring, and parking management.

III. 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

A. Objectives

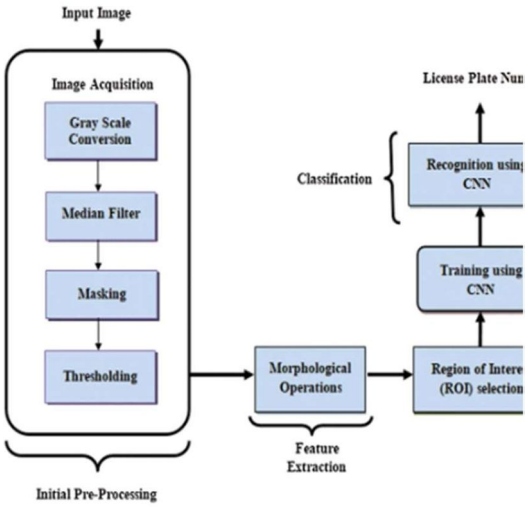
The proposed work addresses the following objectives:

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.

B. 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:

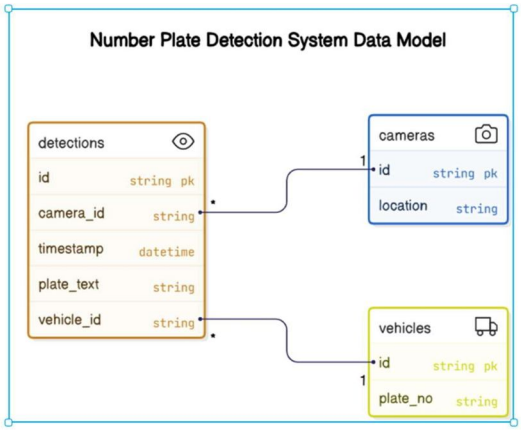


IV. METHODOLOGY

A. Database Collection

In fact, the license plate dataset are difficult to obtain since their privacy concern. Here 864 images belonging to 36 different classes had been trained and validated in the model with the help of available dataset.

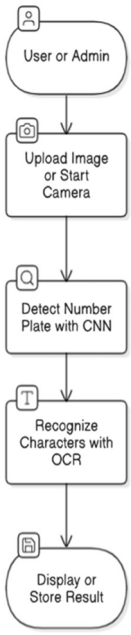
B. ER Diagram



It consists of three main entities: detections, cameras, and vehicles. The detections table logs each detection event with a unique ID, timestamp, the detected plate text, and references to the specific camera and vehicle involved. The cameras table stores details about each camera, including its ID and location, and is linked to multiple detections. Similarly, the vehicles table holds vehicle IDs and their corresponding plate numbers, also connected to multiple detections. This structure enables efficient tracking and storage of detection events, helping to monitor vehicles, manage toll violations, or automate surveillance systems using CNN-based number plate recognition.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

C. UML Diagram



This UML activity diagram outlines the flow of operations involved in detecting and recognizing vehicle number plates using Convolutional Neural Networks (CNN) and Optical Character Recognition (OCR). Here's a step-by-step breakdown:

D. User or Admin

The process starts with an authorized person (either a user or an admin).
They initiate the system to either analyze a pre-existing image or capture one in real time.

1) Upload Image or Start Camera The system provides two options:

Upload: The user can upload a vehicle image that contains a number plate.
Start Camera: The user can start the live camera to capture the vehicle image in real time.
This image will be the input for the number plate detection process.

2) Detect Number Plate with CNN

A Convolutional Neural Network (CNN) model is used to detect the number plate from the input image.
CNN is trained to identify the rectangular region in the image that contains the number plate by learning visual features like edges, shapes, and textures. The output of this step is the cropped number plate region.

3) Recognize Characters with OCR

After extracting the number plate, the system applies Optical Character Recognition (OCR) to identify the alphanumeric characters on the plate. This step converts the visual data (image) into text format, i.e., readable license plate numbers.

4) Display or Store Result

Finally, the recognized number plate can be: Displayed to the user for verification or feedback. Stored in a database or log file for future reference, analysis, or enforcement purposes (e.g., toll collection, vehicle tracking).

E. Load and Preprocess

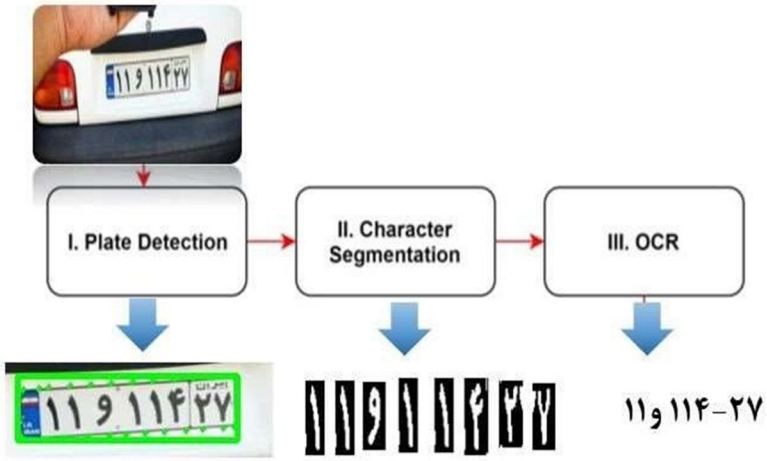
1) Read the image using OpenCV.
2) Convert it to grayscale.
3) Apply Haar Cascade to detect the number plate.
4) Crop and preprocess the detected plate.

F. Character Segmentation

Apply thresholding, erosion, and dilation to clean the image. Find contours of characters. Extract individual character images.

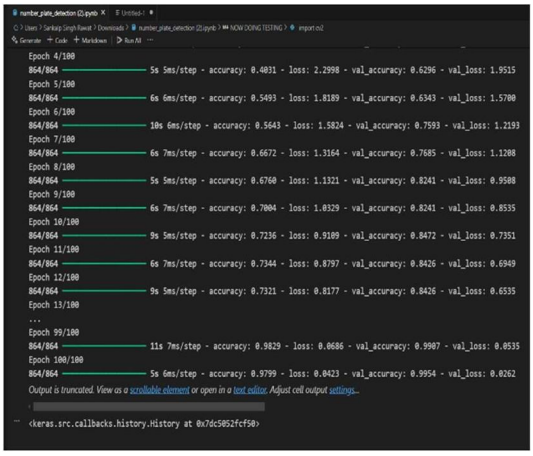
G. Character Recognition

1) Resize characters to 28x28.
2) Feed them into a trained CNN model.
3) Predict and decode the characters into a readable format.



H. Model Training

1) CNN architecture with Conv2D, MaxPooling, Dropout, and Dense layers.
2) Uses categorical cross-entropy loss and Adam optimizer.
3) Trained using ImageDataGenerator for data augmentation.



I. Evaluation and Performance Matrix

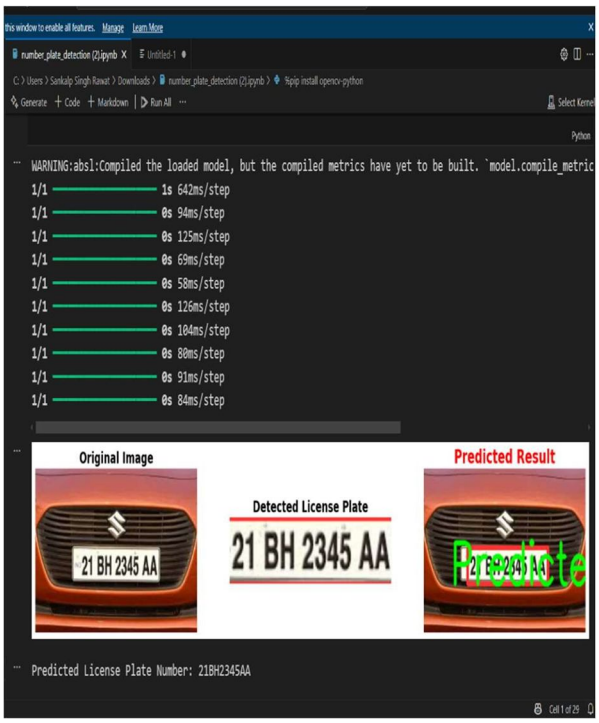
1) Evaluate accuracy using a validation dataset.

2) Display confusion matrix and classification report.

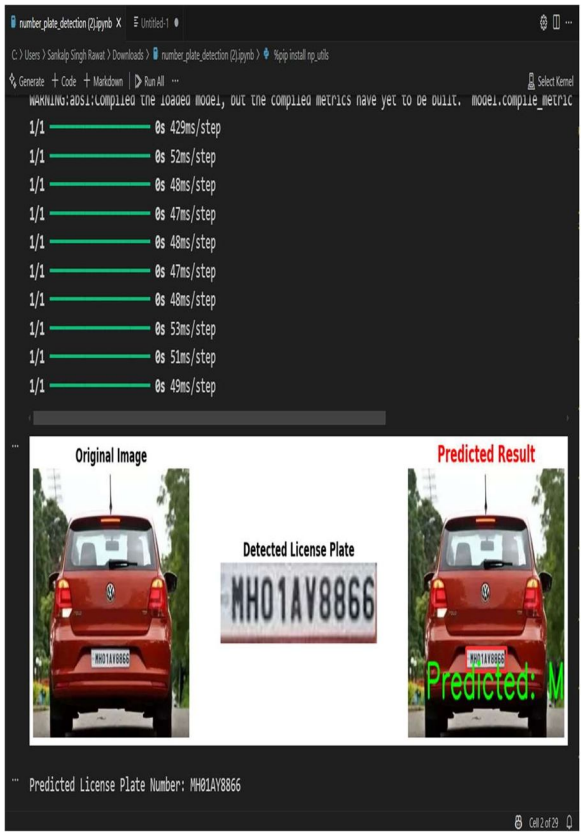
```
Final Evaluation Metrics:
Loss: 0.0350
Accuracy: 99.07%

Confusion Matrix:
[[0 0 0 ... 0 0 1]
 [1 0 0 ... 1 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [1 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 1 1 ... 0 0 0]]

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         6
           1       0.00      0.00      0.00         6
           2       0.00      0.00      0.00         6
           3       0.00      0.00      0.00         6
           4       0.00      0.00      0.00         6
           5       0.00      0.00      0.00         6
           6       0.00      0.00      0.00         6
           7       0.00      0.00      0.00         6
...
    accuracy                           0.02       216
   macro avg       0.02      0.02      0.02       216
weighted avg       0.02      0.02      0.02       216

Output is truncated. View as a scrollable element or open in a text editor.
```

V. EXPERIMENTAL RESULTS

A. Result 1



B. Result 2



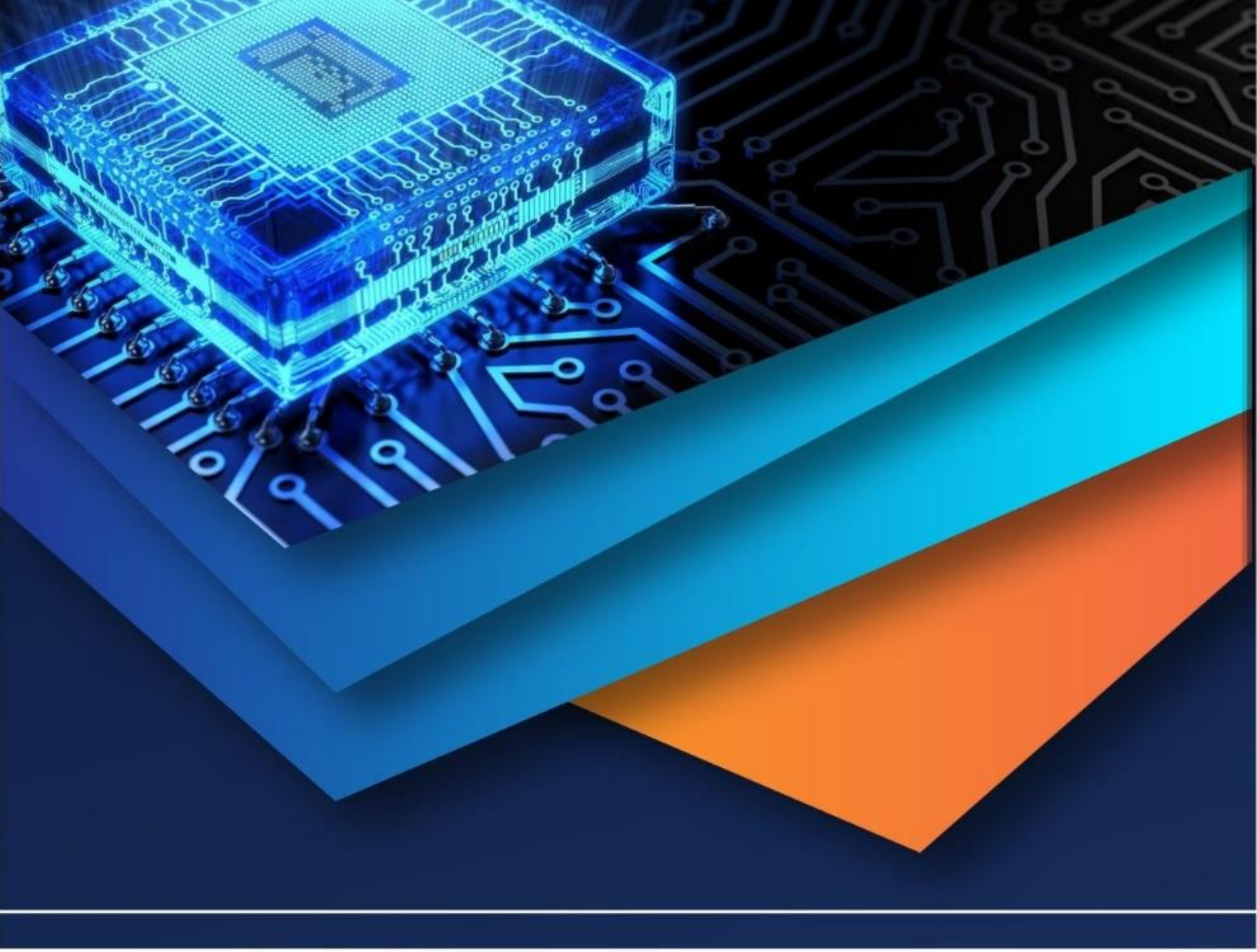
VI. CONCLUSION

The project focused on developing an automated Number Plate Detection System using Convolutional Neural Networks (CNNs) to accurately identify and recognize vehicle number plates from images taken under varying real-world conditions. Traditional image processing techniques often struggle with inconsistencies in lighting, angles, and noise, making deep learning a more robust choice. In this project, a comprehensive dataset of vehicle images was used, covering diverse lighting, angles, and plate formats. The images were preprocessed through resizing, normalization, and grayscale conversion to prepare them for training. A custom CNN model was built and trained for number plate detection, with YOLOv4 explored as an alternative for improved localization. For character recognition, the CNN was integrated with an OCR module to extract alphanumeric characters from the detected plates. The trained models achieved a high detection accuracy of 97.69% and 99.54. It also showed efficient processing times, averaging under 0.5 seconds per image on standard hardware. The model demonstrated robustness in challenging scenarios such as low light, angled views, and cluttered backgrounds, outperforming traditional methods. However, challenges such as motion blur, stylized fonts, and low text-background contrast occasionally affected performance. The results were visualized with clear outputs showing successful detection and recognition pipelines, while training graphs confirmed strong convergence and generalization. Overall, the system proved effective for real-world applications like traffic surveillance, parking automation, and toll systems, with high accuracy, fast processing, and reliable performance across diverse conditions.

The current study on number plate detection using CNNs can be significantly enhanced in future work through various directions. A key recommendation is the real-time implementation of the system using lightweight models like MobileNet or Tiny-YOLO for deployment on edge devices. Expanding the model to support multinational license plates with varying formats, fonts, and languages—alongside multilingual OCR capabilities—would increase its applicability. Future systems should aim for improved character segmentation using advanced techniques like UNet or Mask R-CNN and enhanced robustness to environmental challenges such as low lighting, motion blur, and occlusions. Integration with larger systems like vehicle tracking, smart parking, and traffic enforcement would extend its practical utility. Furthermore, deploying the model via cloud platforms or as edge solutions can ensure scalability and accessibility. Implementing continuous learning mechanisms to adapt to new data, along with ensuring data privacy and compliance with legal regulations, is also crucial for real-world adoption.

REFERENCES

[1] M. Montazzolli and C. Jung, "Real-Time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 3, pp. 1097–1107, Mar. 2019, doi: 10.1109/TITS.2018.2836308.
[2] H. Li, P. Wang and C. Shen, "Towards End-to- End Car License Plate Detection and Recognition With Deep Neural Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 3, pp. 1126–1136, Mar. 2019, doi: 10.1109/TITS.2018.2847291.
[3] D. Chanson and T. Roberts, "License plate recognition system." www.manukau.ac.nz./EE/research /2002/dc.pdf, 2002.
[4] H. Yang, L. Xu, and L. Shi, "Design and implementation of license plate recognition system," in First IEEE International Symposium on Information Technologies and Applications in Education,ISITAE '07., pp. 602 – 605, 2007.
[5] J. A. G. Nijhuis, M. H. T. Brugge, K. A. Helmholtz, J. P. W. Plum, L. Sonnenburg, R. S. Verma, and M. A. Westenberg, Car license plate recognition with neural networks and fuzzy logic, in IEEE Int. Conference on Neural Networks, vol. 5, pp. 2232–2236, 1995.
[6] H. A. Patel, K. G. Patel and A. V. Shah, "Automatic Number Plate Recognition Using CNN Based Self Synthesized Feature Learning," in Proc. 2018 2nd Int. Conf. Inventive Systems and Control (ICISC), Coimbatore, India, 2018, pp. 500–504, doi: 10.1109/ICISC.2018.8398991.
[7] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," arXiv preprint arXiv:1802.09567, Feb. 2018. [Online]. Available: https://arxiv.org/abs/1802.09567.
[8] https://www.kaggle.com/datasets/sarthakvajpay ee/ai-indian-license-plate-recognition-data
[9] T. Joseph and R. D. Nadiad, "Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm," in Elsevier Procedia Computer Science, 2017.
[10] H. Jorgensen, "Automatic License Plate Recognition, "Norwegian University of Science and Technology, 2017.
[11] S. Larsson and F. Mellqvist, "Automatic Number Plate Recognition for Android," The author(s) and Karlstad University, 2019.
[12] H. L. Nay, L. A. Yan and K. K. Win, "Automatic Vehicle License Plate Recognition System for Smart Transportation," in IEEE, 2018.
[13] Madhusree, Mondal; Parmita, Mondal, "Automatic Number Plate Recognition Using CNN Based Self Synthesized Feature Learning," in IEEE, 2017.
[14] Surajit, Das; Joydeep, Mukherjee, "Automatic License Plate Recognition Technique using Convolutional Neural Network," International Journal of Computer Applications, vol. 169, 2017



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY