

A Real-time Robust English and Bangla License Plate Recognition Using Enhanced GAN and Explainable CNN

Jul Jalal Al-Mamur Sayor^{1,2}, Nishat Tasnim Shishir^{1,3}, Mahe Zabin⁴, Suman Saha¹, Kamruddin Nur⁵, Jia Uddin⁶

¹Department of Internet of Things and Robotics Engineering, University of Frontier Technology, Bangladesh

²Department of Software Engineering, Daffodil International University, Daffodil Smart City, Birulia, Savar, Dhaka 1216, Bangladesh

³Department of Computer Science and Engineering, Daffodil International University, DSC, Birulia, Savar, Dhaka 1216, Bangladesh

⁴Human and Digital Interface Department, JW Kim College of Future Studies, Woosong University, Daejeon, 34606, Korea

⁵Department of Computer Science Engineering, American International University, Dhaka 1229, Bangladesh

⁶AI and Big Data Department, Endicott College, Woosong University, Daejeon, 34606, Korea

Article Info

Article history:

Received Jun 18, 2025

Revised Mar 26, 2026

Accepted Apr 16, 2026

Keywords:

License Plate Recognition

Bilingual Systems

Edge Computing

ESRGAN

Explainable CNN

ABSTRACT

Efficient systems tailored to countries with bilingual license plates remain scarce, and existing methods often struggle to adapt the unique challenges posed by these scenarios. Countries like Bangladesh and India face challenges with the existing techniques. Accurate recognition of bilingual license plates is crucial for traffic management, but conventional methods with limited resources often fail to work effectively with low-quality footage. Thus, this research provides a framework for the resource-efficient solution for bilingual number plates, ESRGAN for resolution enhancement and explainable CNNs with transfer learning for language classification and character recognition. We divide the task into two sub-tasks and train four specialized CNN models for Bangla and English characters and numerals to maximize the system's performance. The proposed two-tier architecture optimizes computational efficiency by dynamically loading models based on recognition requirements. Overall, using the gradient-based explainability techniques in the proposed framework increases the credibility of decision-making. The proposed models achieved accuracies of 96%, 97.48%, 97.15%, and 97.62% for English license plate recognition, Bangla license plate recognition, Bangla number recognition, and English number recognition, respectively. While maintaining competitive accuracy, the proposed approach outperforms state-of-the-art works in resource efficiency and computational time, which are crucial for edge deployments. Furthermore, the license plate detection and character extraction algorithm proposed in the paper demonstrates excellent accuracy for both black and white-colored plates. This approach ensures consistent performance and efficient resource utilization while showing the ability to work with a variety of traffic conditions, making it suitable for automated vehicle management systems.

Copyright © 2026 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Jia Uddin

Department of AI and Big Data, Endicott College, Woosong University, Korea

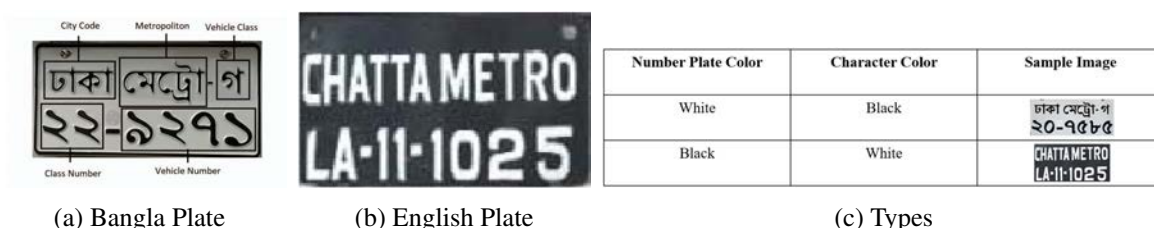
Email: jia.uddin@wsu.ac.kr

1. INTRODUCTION

Vehicle license plate recognition has evolved significantly in recent decades, playing an important role in various sectors such as traffic management [1], toll collection [2], and traffic law enforcement [3]. Especially in developing countries like Bangladesh, there is a great demand for efficient detection and recognition of ve-

hicle license plates. There are more than 5.98 million registered vehicles in Bangladesh, but a greater number of vehicle management systems are highly outdated and incapable of dealing with dual-language plates [4]. This insufficiency in proper automated recognition systems hampers effective traffic control, making vehicle management less effective and laborious, and thus giving rise to the need for a more enhanced Automatic License Plate Recognition (ALPR) system designed explicitly for bilingual environments. The Bangladesh Road Transport Authority (BRTA) standard Bangla license plate is shown in Figure 1a. Each Bangla plate comprises different elements that relate to specific details regarding the registration of the vehicle concerned. Therefore, the Bangladesh Road Transport Authority (BRTA) has not generally authorized English-only plates for regular use in all vehicles. For any such deviation, such as bilingual or English-only plates, proper authorization by the BRTA is required to comply with its regulations. An example of a Bangladeshi English license plate is shown in Figure 1b. There is no standardized format for English license plate usage, leading to the use of various forms across vehicles. This lack of standardization presents a challenge for automated recognition systems, highlighting the importance of developing adaptive frameworks capable of handling both Bengali and non-standardized English plates.

In developing countries like Bangladesh, where bilingual license plates are common [4], existing traffic monitoring systems remain outdated and inefficient for handling dual-language formats. This leads to labor-intensive processing and reduced effectiveness in traffic management. Current ALPR systems face several limitations, particularly under challenging conditions such as blur, noise, low resolution, and damaged plates [5]. Additionally, many methods are restricted to specific plate colors and rely on high-resolution inputs, resulting in increased computational cost and hardware dependency, which is impractical in resource-constrained environments [6]. Furthermore, most existing approaches are monolingual, making them unsuitable for regions like Bangladesh, India, and Saudi Arabia, where bilingual plates are prevalent [7]. Therefore, an effective ALPR system must support multi-language recognition while maintaining computational and time efficiency, ensuring a balanced trade-off between accuracy and practical deployability.



(a) Bangla Plate

(b) English Plate

(c) Types

Figure 1. Examples of Bangla and English license plates and targeted plate types

In the study, we proposed an approach that is capable of working efficiently on both white and black colored plates for black or white characters, as shown in Figure 1c. The motivation for this work is the increasing demand for efficient systems for vehicle management in developing nations and also where license plates are in bilingual form. By adopting GAN-based super-resolution techniques for the enhancement of low-quality images, the CNN divide-and-conquer strategy for language-specific character recognition, our approach aims to overcome the limitations of existing systems while balancing the trade-off between accuracy and resource efficiency, making the solution suitable for use in diverse, real-world traffic environments by ensuring scalability with reliability. The contributions of this research are given below:

- i. We have designed a robust framework that will be able to detect and recognize license plates in both Bengali and English with the ability to perform operations on dark and light plate colors, as well as the corresponding variations in character colors.
- ii. We have employed GAN-based super-resolution techniques to address the problems of low-resolution traffic imagery, which will enhance the clarity of license plates and improve both character segmentation and recognition.
- iii. We have implemented a novel divide-and-conquer approach using CNNs for language-specific character recognition. This strategy improves the recognition accuracy by making the model more language-specific and thus handles the intricacies of both Bengali and English characters with greater efficiency.
- iv. We have integrated a real-time database matching module, which allows for efficient license plate validation and compliance checks, thus ensuring seamless operation in real-world scenarios that demand

immediate verification.

- v. We have designed a computationally efficient framework that balances high accuracy with resource optimization, enabling real-time applications on low-cost hardware platforms.

The rest of this paper is organized as follows. Section 2. summarizes existing studies and highlights the research gaps. Section 3. explains the research methodology, including data collection and preparation, model design, and the techniques employed such as plate detection, tilt detection and correction, color inversion of plates, individual character segmentation and extraction. Section 4. presents the experimental results obtained by analyzing the performance of the proposed model. Finally, section 5. concludes the paper by emphasizing its contributions, potential improvements, and future research directions.

2. LITERATURE REVIEW

Automatic License Plate Recognition (ALPR) systems have undergone the development of improved techniques enhanced by deep learning and computer vision to address multiple issues that arise when faced with different types of license plates that have different formats, lighting conditions, tilts, color variations and inferior computing power. This chapter organizes and discusses prior research into several thematic strands to provide a clear and systematic view of existing approaches, their advantages, and drawbacks. The review also includes binarized and regional license plates, light versions for IoT applications, end-to-end real-time system applications, and lessons learned from general reviews of ALPR approaches. These studies form the starting point for analyzing deficiencies and mapping the outlines of a more reliable license plate detection and recognition system.

2.1. Bilingual and Region-Specific License Plate Works

Recently, some studies have addressed challenges in bilingual or region-specific license plates, focusing on unique regional characteristics, fonts, and formats. [8] proposed a method for automatic license plate detection and recognition by using Convolutional Neural Networks (CNNs). In their approach, they used the detection of corner points of license plates, which involved the subsequent process of plate rectification by perspective transformation and then character recognition. The proposed system effectively copes with different license plate formats under changing environmental conditions such as illumination and viewing angles. However, this approach had some notable limitations. The system is error-prone because correct corner point detection relies on scenes in which the plate is partially occluded or severely distorted, or where the edges are not clearly visible. These incorrect localizations result in to poor character recognition. Therefore, this approach may fail for plates with non-standard fonts or significant wear and tear, further worsening its recognition performance. [9] investigated the performance of deep learning models, including YOLOv8, SSD, and Faster R-CNN, in detecting and recognizing Iranian motorcycle license plates. In the dataset, the images were of different sizes, angles, and illuminations. The best results were obtained by YOLOv8, which had the highest performance with 98.5% for detection and 99% for identification. Despite the high accuracy achieved, the model performance degrades when subjected to extreme lighting conditions or plates that are heavily occluded. [10] presented an overall review of approaches that use neural networks for Automatic License Plate Recognition. The authors introduced various architectures of neural networks, such as convolutional neural networks, residual networks, recurrent neural networks, and LSTM networks, which are applied to different ALPR pipeline stages (detection, extraction and recognition). CNNs are emphasized in their effectiveness of feature extraction, whereas residual networks help to train deeper models by avoiding the vanishing gradient problem. RNNs and LSTMs are applied to handle sequential data, which significantly helps in the recognition of character sequences on license plates. In spite of the progress, the study highlights the challenges such as "region-dependent license plate formats, sizes, and colors," environmental factors like "lighting and occlusions" that affect model accuracy, high demands for computational resources, and large annotated datasets, which once again restricts the feasibility of deploying these models in every scenario.

2.2. Lightweight Models for Embedded or IoT Systems

Lightweight architectures are pivotal for resource constrained environments like IoT systems and edge devices. [11] presented the development of a lightweight ALPR model that should be easily deployable in resource-constrained environments, such as embedded devices or IoT systems. In the following model, there is

a good balance between accuracy and computational efficiency because it has a much more simplified architecture. However, the model may perform poorly with plates with complex backgrounds or unconventional fonts, resulting in less than ideal recognition accuracy. On the other hand, the model's lightness, although helpful for its deployment on low-powered devices, results in limitations while working with highly variable license plate appearances. [12] presented an approach by which the YOLOv6 algorithm can serve for license plate detection in IoT-based smart parking. The model they used was trained on a customized dataset consisting of images captured under different parking conditions, with the aim of further improving the effectiveness and accuracy of detection in practical IoT scenarios. YOLOv6 allows fast and efficient license plate detection suitable for IoT environments. However, it lacks detailed performance metrics, such as precision, recall, and F1 score, which are important metrics in the evaluation of this task. Not performing a thorough evaluation does not provide an indication of how well the system works under changing lighting conditions, occlusion, or diverse plate formats that may be experienced in reality.

2.3. End-to-End and Real-Time Systems

Some research has targeted single-pass or integrated systems for real-time license plate detection and recognition. [13] introduced a single-pass deep neural network that integrates license plate detection and character recognition, which is suitable for real-time performance in autonomous vehicles. The model architecture involved a single pass with the intent of reducing the processing time for applicability in high-speed tasks. Nevertheless, this methodology lacks the ability to handle low-resolution or highly tilted license plates, leading to degraded performance. This is especially problematic in autonomous vehicle environments where plates can be captured from unconventional angles or under adverse conditions, thus increasing the demand for more robust handling of such situations. [14] presented an end-to-end ALPR system with depthwise separable convolution networks so that computational complexity can be reduced without seriously affecting accuracy. This model achieved an accuracy of more than 99% by processing images at over 70 frames per second, thereby achieving efficiency and effectiveness. However, it has not been thoroughly evaluated under difficult weather conditions or in plates under bad conditions. This lack of evaluation in difficult conditions limits the understanding of the robustness of the system in terms of its performance, maintaining high accuracy in real-world environments. [15] developed a multistage system that integrates object detectors, an image classifier, and a multi-object tracker to identify vehicles and license plates. Emphasizing a system that processes video streams in real-time at edge devices, achieved higher accuracy by temporal redundancy. Despite its enhanced accuracy, it still has a number of limitations: the increased complexity of object detector integration, image classifier, and multi-object trackers increases latency, challenging real-time performance maintenance, especially on edge devices with low computational power. The high computational load may also reduce the scalability for larger deployments and increase hardware costs. Moreover, synchronization between several stages might lead to inconsistencies, and the quality of video input may affect system performance owing to poor lighting, occlusions, and other environmental factors. Balancing the accuracy with the requirements for real-time inference remains a key challenge in this approach.

2.4. Reviews and General Insights into ALPR

This section provides an overview of ALPR methodologies and challenges. [16] presented an overall review of approaches using neural networks for Automatic License Plate Recognition. The authors introduced various architectures of neural networks, such as convolutional neural networks, residual networks, recurrent neural networks, and LSTM networks, which were applied to different ALPR pipeline stages (detection, extraction, recognition). CNNs are emphasized in their effectiveness of feature extraction, while residual networks help to train deeper models by avoiding the vanishing gradient problem. RNNs and LSTMs are applied to handle sequential data, which significantly helps in the recognition of character sequences on license plates. In spite of the progress, the study highlights the challenges such as "region-dependent license plate formats, sizes, and colors," environmental factors like "lighting and occlusions" that affect model accuracy, high demands for computational resources, and large annotated datasets, which once again restricts the feasibility of deploying these models in every scenario. Further research is needed to improve robustness and efficiency in very different environmental conditions where data are scarce.

[17] carried out a systematic review of ALPR systems based on deep learning, addressing the challenging conditions of variable lighting, occlusion, and diverse plate designs. This review underlines the effectiveness of CNNs and recurrent neural networks in handling such challenges but also highlights significant limitations for example, most of the systems reviewed have low-resolution images that depress both detection and recognition

accuracy. In addition, most of the reviewed systems are suffering from real-time processing constraints, hence becoming unsuitable in high-speed scenarios where immediate recognition must be performed, such as in traffic monitoring and control applications. [18] reviewed some of the deep learning techniques applied to ALPR, which discusses several models and their performance measures. Although this is an informative paper, it lacks experimental verification. Thus, it is mostly theoretical rather than practical.

2.5. Yolo and OCR based works

YOLO models and methods of Optical Character Recognition now find widespread application in Automatic License Plate Recognition systems, balancing real-time performance with highly accurate detection and recognition. This section explores the integration of YOLO architectures and OCR techniques in the ALPR pipeline, describing the contributions, challenges, and possible improvements to such solutions. [19] used YOLOv4 for the detection of license plates, coupled with Tesseract OCR for character recognition. Their model demonstrated high accuracy with plate detection in high-resolution images, with 96% detection and 93% recognition accuracy. However, it degraded under images, with very low resolution or significant distortions. Furthermore, using a two-step pipeline introduced latency, which would be less optimal for real-time usage under a resource-constrained environment. [20] developed a system using YOLOv5 for plate detection in different environmental conditions; then, they applied a lightweight OCR module to recognize the detected characters. Their system worked well, recognizing plates with medium-level occlusions and under different lighting conditions, with an accuracy of approximately 90.5%; however, the system showed several limitations when the license plate was blurred or very reflective in such cases, OCR processes were not able to recognize the letters due to the lack of proper preprocessing techniques.

[21] explored YOLOv7 for license plate detection in dynamic environments, including congested urban areas. This method was developed by combining YOLO for localization with a hybrid OCR model for multilingual character recognition. Although this system proved to be promising with 92% recognition accuracy, it relied heavily on the need for annotated multilingual datasets, a task quite challenging for scalability in regions with multiple languages and plate formats. Similarly, [22] proposed a YOLOv8-based license plate detection system integrated with PaddleOCR for character recognition. Their prototype, implemented on a Mini PC for real-time processing, achieved 91.6% detection accuracy and 86% OCR accuracy. The system demonstrated efficient performance for traffic monitoring and access control applications. However, similar to other YOLO-OCR pipelines, it remains dependent on OCR performance and lacks advanced preprocessing or enhancement techniques, which may limit robustness under challenging conditions such as low resolution or noise.

These studies emphasize the effectiveness of YOLO models in the rapid and precise detection of license plates and the recognition of characters by OCR. However, challenges persist in achieving robust performance under adverse conditions, high computational demands, and the need for large annotated datasets for domain-specific improvements.

2.6. Multi-Format or Vehicle-Specific Plates

This section focuses on research related to specific vehicle types and varying license plate formats. It highlights the development of systems for Automatic License Plate Recognition (ALPR), emphasizing that most advancements have occurred in deep learning and computer vision. These improvements address challenges such as variable plate formats, lighting conditions, occlusions, and computing constraints. [23] presented a study related to ALPR for two-wheelers, normally having small, variably designed plates. Their research used deep learning techniques to develop a model capable of recognizing license plates in very challenging scenarios, including those when plates are dirty or partially occluded. Their approach included preprocessing the input, finding the text using a convolution neural network, and applying the character recognition technique. As their model performed satisfactorily in overall performance, the major weakness appeared as reduced accuracy under challenging conditions regarding robustness and generalization of the model by the occurrence of dirt or occlusion. These findings delineate the effective further improvements required towards such practical issues. [24] contributed to the state-of-the-art solution to address issues related to license plate recognition for vehicles with different plate formats and various environmental conditions. This system utilizes GANs to generate synthetic images to augment the dataset, thereby addressing the challenge of limited training data. CNNs are used for robust feature extraction, whereas ELNs serve as lightweight classification mechanisms for efficient training. The hybrid model is designed to be more accurate and computationally efficient. However, the study identifies limitations in handling severe environmental challenges such as poor lighting and heavy occlusion, as well as limited robustness to rare or non-standardized plate designs.

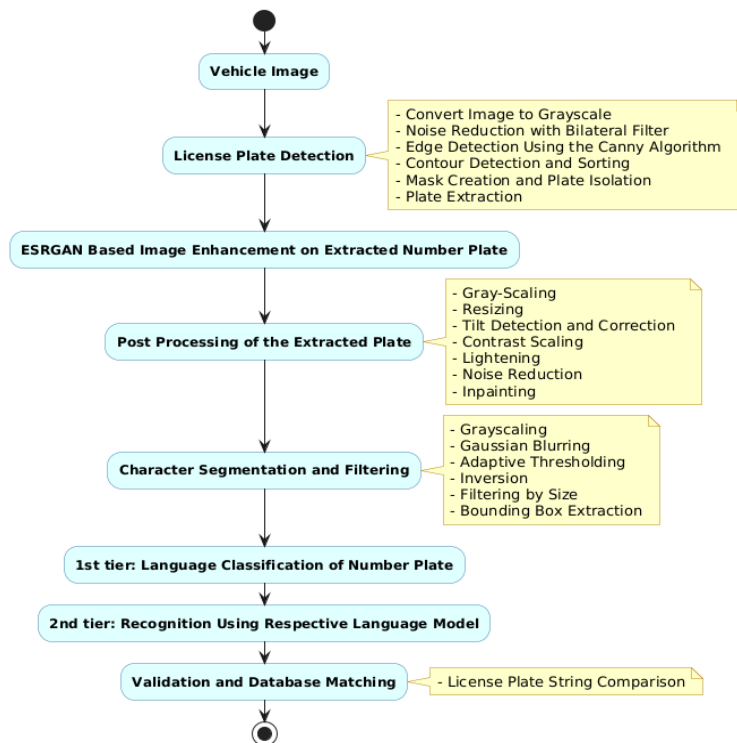


Figure 2. Workflow of the Proposed Methodology

[8] and [9] demonstrated high detection and recognition accuracy in Automatic License Plate Recognition (ALPR) systems, although they face difficulties with non-standard fonts, occlusions, and changes in lighting. While lightweight models, such as those by [11] are efficient for IoT devices, they compromise accuracy in complex scenarios. The end-to-end and YOLO-based methods have excellent performance in real-time applications but are limited under bad conditions and with non-standard datasets. End-to-end and YOLO-based methods proposed by [14] and [21] achieved excellent performance in real-time applications but were limited under bad conditions and with non-standard datasets.

The above studies highlight the development of Automatic License Plate Recognition (ALPR) systems, underlining that most of the improvements occurred in deep learning and computer vision regarding challenges such as variable plate formats, lighting, occlusions, and computing constraints. This work builds upon these studies by proposing a lightweight bilingual vehicle license plate detection and recognition system that is both robust and adaptable for region-specific license plates. This system incorporates optimized preprocessing, efficient CNN architecture, and real-time compatibility, addressing challenges such as poor lighting conditions, occlusions, and diverse formats through ESRGAN-based resolution enhancement. In comparison to the previous studies, the proposed system has achieved a good balance between computational efficiency and accuracy, thus making it suitable for scalable and resource-constrained environments.

3. METHODOLOGY

The significant challenge of this study was to collect a decent amount of character-wise segmented image data for the proposed bilingual recognition operation. Only vehicle number plate images are available, but those are not the proper choices for character-based recognition models. We combined some benchmark datasets for our purpose and applied the divide and conquer (D&C) method, which aims to improve training efficiency, accuracy, and flexibility. This method overcomes the drawbacks of conventional CNN training, including the difficulty of managing big, diverse datasets with a considerable number of classes and high computing costs by dividing the dataset and tasks into manageable components. A simplified flow of the overall methodology is shown here in figure 2.

3.1. Data Collection

We developed a dataset of character-wise segmented images using three benchmark datasets. The "Bangla License Plate Cropped Dataset" [25] contains 79,700 images from 50 classes. The "License Plate Digits Classification Dataset" [26] contains 35,500 images from 35 classes, and the "License Plate Recognition using CNN dataset" [27] contains 864 images from 36 classes. Like [28], we used manually collected images to increase diversity in the dataset. The final dataset contains Bangla Number Recognition (10 classes, 5000 images), English Number Recognition (10 classes, 5000 images), Bangla Character Recognition (31 classes, 15,500 images), and English Character Recognition (36 classes, 17,000 images). Table 1 represents the dataset used in this research. The dataset used for training the Bangla model in this study was collected from mentioned existing dataset [25] and to increase the diversity and robustness of the dataset, we further augmented it with our own collected images. The existing Bangla dataset contained several issues, including misclassified samples, class imbalance and duplicate entries. These issues were carefully identified and resolved. The dataset link is provided here (<https://tinyurl.com/4d9njnrx>), and the complete dataset can be made available upon request.

Table 1. Summary of Datasets Used for this study.

Dataset Name	Number of Classes	Number of Images
Bangla Number Recognition	10	5000
English Number Recognition	10	5000
Bangla Character Recognition	31	15,500
English Character Recognition	36	17,000

3.2. Data Preparation

The bilingual dataset was created using the four datasets: [25], [26], [27], and [28]. However, the datasets contained redundant data, which could lead to increased computational cost and potential bias in the models. For this reason, the datasets were filtered through several iterations to remove redundant data and correct the models' classifications. Additionally, the character images were standardized in font and style, and more real-world samples were added to the datasets to improve their diversity. The datasets were also balanced to ensure the distribution of the samples in all classes, as shown in Figure 3, to improve the reliability of the models and eliminate potential bias. Finally, the datasets were formatted in class-based structure without the need for XML annotations, which enables the datasets to be directly inputted into the CNN models.

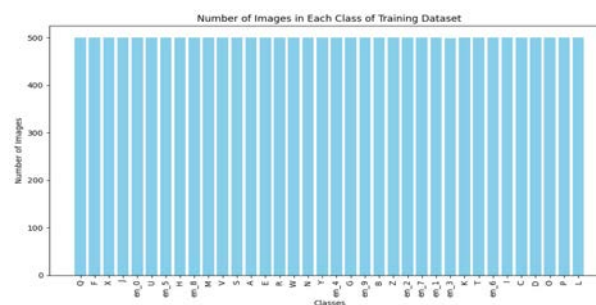


Figure 3. Dataset Balancing

3.3. License Plate Detection

After enhancing the image quality, we developed a custom algorithm to accurately locate and isolate the license plate region within each image. This stage was critical for minimizing background interference and focusing on the subsequent recognition processes solely on the plate. The proposed algorithm follows a systematic approach that combines grayscale conversion, noise reduction, edge detection, and contour-based filtering to effectively detect and extract the plate region. The license plate detection process involves the following key steps: contour detection, feature sorting, mask elaboration, plate separation, and plate identification, as shown in algorithm 1. The contour detection is conducted by identifying possible shapes in the edge-detected image,

with a primary interest in the contour of the quadrilateral shape, to eliminate unwanted areas or objects that add to the number of false positives. Mask creation, therefore, helps to separate the plate region by creating a black-and-white binary mask of the plate where the plate is white.

Algorithm 1: License Plate Detection from Car Image

```

Input : Car image
Output: Detected and cropped license plate region

Load the car image from the specified path;
if image fails to load then
  | Print an error message and terminate the algorithm;
Convert the image to grayscale;
Apply a bilateral filter to reduce noise while preserving edges;
Detect edges using the Canny edge detection algorithm;
Identify contours from the edge-detected image;
Sort contours by area in descending order and select the top 10;
foreach contour in selected contours do
  | Approximate the contour to a polygon;
  | if polygon has four vertices then
    | | Assign this contour as the license plate region;
    | | Break the loop;
Create a binary mask of zeros matching the grayscale image size;
Draw the selected license plate contour onto the mask;
Extract the license plate region using a bitwise AND between the original image and the mask;
Compute the bounding rectangle coordinates of the extracted region;
Crop the license plate from the original image;
Display the detected and cropped license plate;
  
```

At the same time, the rest is black, erasing some irrelevant features such as the surroundings of roads or vehicles. Finally, the plate extraction process removes the license plate from the image by focusing solely on the bounding box of the mask, resulting in a crisp and clear image of the plate. Figure 4 shows the steps associated with the plate detection and extraction processes.

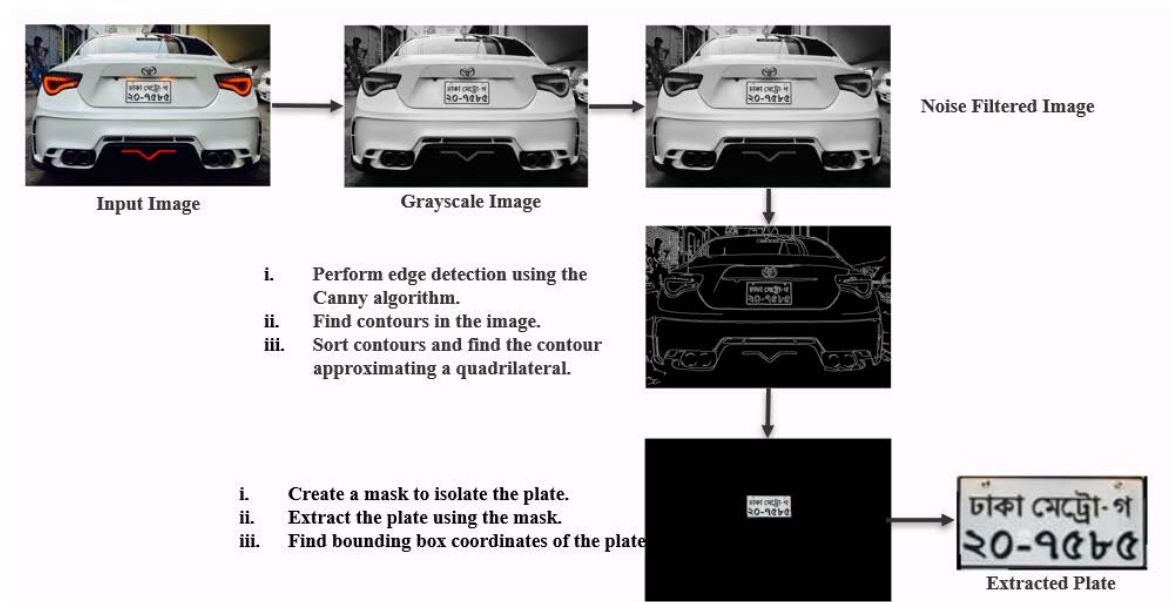


Figure 4. Vehicle Number Plate Detection and Extraction.

3.4. ESRGAN-Based Image Enhancement on Extracted Number Plate

Once the number plate was detected and extracted, a significant portion of images particularly from real-time traffic feeds—were found to suffer from low resolution and quality degradation, which adversely affects CNN-based character recognition. To mitigate this issue, we employed a pre-trained ESRGAN (Enhanced Super-Resolution Generative Adversarial Network) model for image enhancement. ESRGAN was selected due to its superior capability in reconstructing high-frequency details and preserving edge sharpness compared to

conventional interpolation-based upscaling techniques. Figure 5 illustrates the working architecture of ESRGAN and its role in enhancing input image resolution.

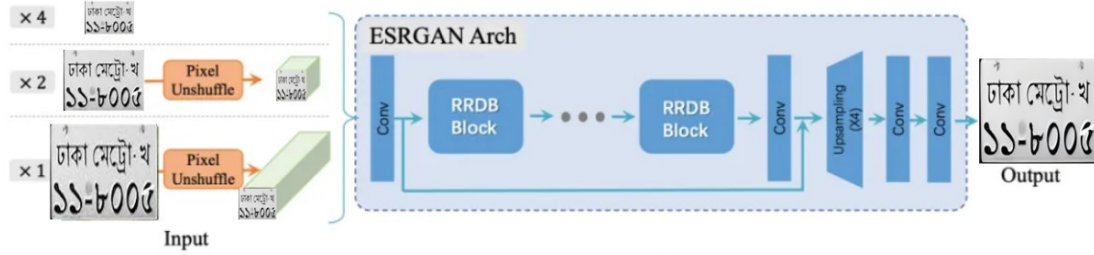


Figure 5. ESRGAN Working Architecture

In this work, ESRGAN is not used as a purely off-the-shelf component; rather, it is adapted through domain-specific degradation modeling and task-oriented optimization. Instead of relying on standard bicubic downsampling, realistic degradation operators were introduced, including motion blur kernels (vehicle dynamics), Gaussian blur ($\sigma = 1.5\text{--}3.5$), JPEG compression (quality 20–50), and low-light noise injection (Poisson + Gaussian). At the architectural level, Residual-in-Residual Dense Blocks (RRDB) were tuned by reducing residual scaling ($0.2 \rightarrow 0.1$) to stabilize training for fine-text reconstruction, while feature emphasis was biased toward high-frequency components (character edges). Furthermore, the ESRGAN loss function was customized to prioritize OCR-relevant features. The perceptual loss weight (VGG-based) was increased, while adversarial loss was reduced to suppress hallucinated textures:

$$L_{total} = L_{pixel} + \lambda_1 L_{perceptual} + \lambda_2 L_{adversarial} \quad (1)$$

where $\lambda_1 = 1.0$ and $\lambda_2 = 0.005$. These adaptations significantly improve image clarity, contrast, and edge definition, facilitating more accurate character segmentation and recognition under real-world conditions. It is important to emphasize that the novelty of this work lies not in proposing a new GAN architecture, but in the effective task-specific adaptation of ESRGAN for license plate recognition. Additionally, although GAN-based models are computationally intensive, ESRGAN is executed on GPU (RTX 3060) and applied only to cropped plate regions, reducing overhead while maintaining real-time performance. For scalability, computationally intensive operations are offloaded to cloud-based GPU servers, ensuring an effective balance between accuracy and efficiency.

3.5. Processing of the extracted plate for Enhanced Recognition

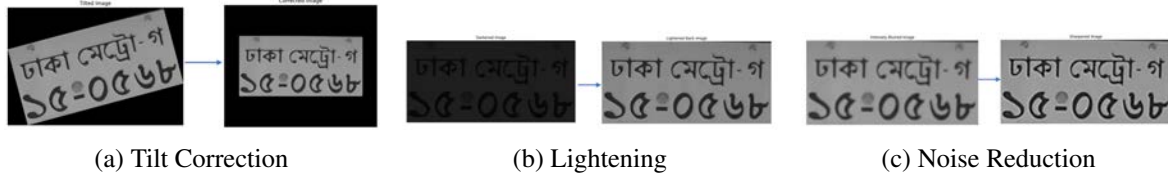
Recognition of the license plate number is followed by post-processing of the extracted license plate, the goal of which is to improve the image quality and, hence, the readability of the characters. This phase considers issues, such as low contrast, ill illumination, and small defects to ensure that the extracted plate is ready for correct segmentation and recognition. The techniques implemented in this process include:

Algorithm 2: Tilt Detection and Correction

Input : Vehicle number plate image
Output: Corrected image with upright orientation

Load the input image;
if image is not loaded **then**
 Print error message and terminate;
Convert the image to grayscale;
Apply Canny edge detection;
Apply Hough Transform to detect lines;
Extract angles of detected lines;
Compute the median angle as tilt angle;
if |tilt angle| > threshold **then**
 Compute correction angle as negative tilt angle;
 Generate rotation matrix;
 Rotate the image using the rotation matrix;
Display the corrected image;

i. Tilt Detection and Correction: Tilt angles are detected by converting images to grayscale, applying Canny edge detection, and using the Hough Transform to identify dominant horizontal lines. The median angle determines skew, which is corrected if it exceeds a threshold (Figure 6a). **ii. Grayscale:** The plate image is converted to grayscale to reduce computational complexity and ensure uniform preprocessing.



(a) Tilt Correction (b) Lightening (c) Noise Reduction
Figure 6. Preprocessing steps: tilt correction, illumination adjustment, and noise reduction

iii. Resizing: All images are resized to a standard dimension to maintain consistency and improve feature extraction. **iv. Contrast Scaling:** Contrast is enhanced using histogram equalization or contrast stretching to highlight characters, especially under poor lighting conditions. **v. Lightening:** Brightness is adjusted to improve visibility of characters in low-light images without overexposure (Figure 6b). **vi. Noise Reduction:** Gaussian and median filtering are applied to remove noise while preserving character edges (Figure 6c). **vii. Character Inpainting:** Telea's inpainting method [29] is used to restore occluded or damaged character regions, improving continuity and recognition accuracy (Figure 7).

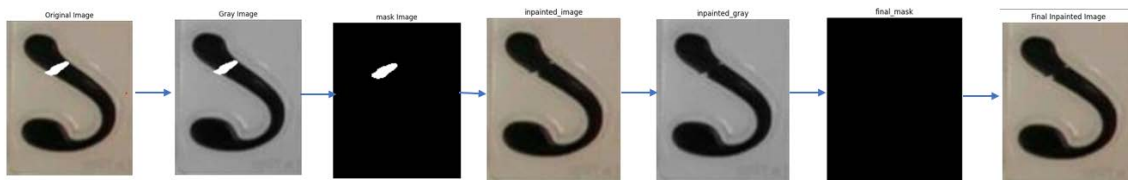


Figure 7. Image Inpainting Steps

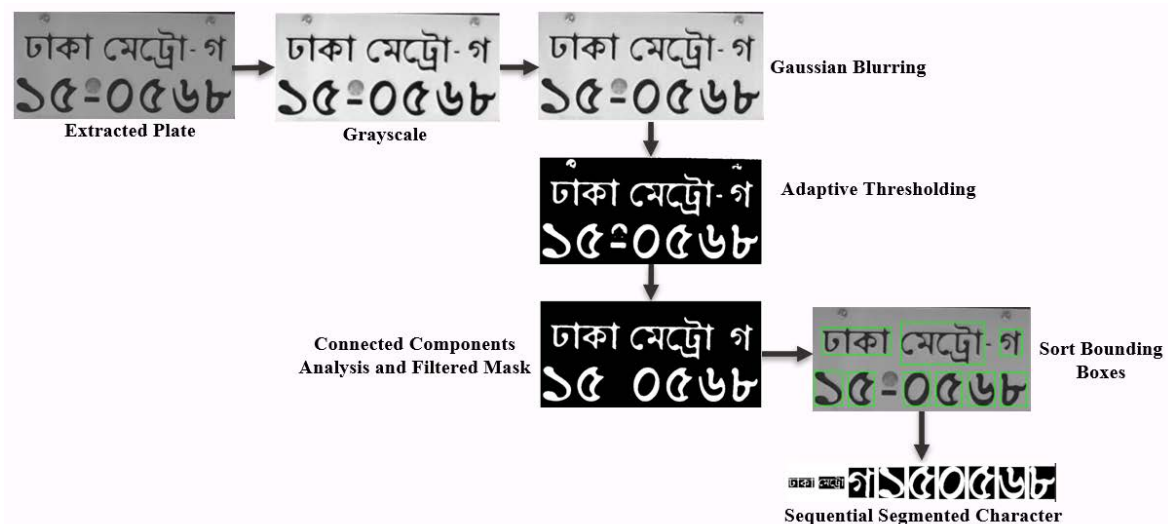


Figure 8. Character Segmentation and Extraction Flow

3.6. Character Segmentation and Filtering

After optimization, character segmentation is performed to recognize each character individually through grayscale conversion, noise filtering, adaptive thresholding, and connected component analysis (Algorithm 3). A dynamic method is employed in which plates with white backgrounds are processed directly (Figure 8), and plates with black backgrounds are inverted based on intensity values and then segmented (Figure 9).

Algorithm 3: Character Segmentation from Extracted Number Plate

```

Input : Extracted Number Plate
Output: Segmented characters displayed with bounding boxes
Convert image to grayscale
Calculate the average intensity of the top-left corner to check background intensity
if background intensity is dark (below 127) then
  | Invert the grayscale image Display the inverted grayscale image
else
  | Proceed without inversion Display the original grayscale image

Apply Gaussian blurring to reduce noise
Apply adaptive thresholding to obtain a binary image
Perform connected components analysis on the thresholded image
Initialize a mask of zeros with the same shape as the thresholded image
Define size bounds for character detection:
  Lower bound = total_pixels divided by 150
  Upper bound = total_pixels divided by 10
foreach unique label in labels do
  | if label is background then
  |   | continue
  | Create a mask for the current label
  | Count non-zero pixels in the label mask
  | if number of pixels between lower and upper bounds then
  |   | Add label mask to the main mask

Find contours in the mask and extract bounding boxes
Sort bounding boxes from left to right and top to bottom using custom comparison
foreach bounding box (x, y, w, h) do
  | Draw rectangle on the original image or inverted image
Extract and Store each segmented character individually from the bounding boxes
Display each segmented character individually in a list format

```



Figure 9. Inversion and Extraction for Black Colored Plates

i. Grayscale: As for the first stage in pre-processing for character segmentation, the license plate image we have been using was transformed to grayscale. This standardization process made it easier to achieve uniformity within the images and eliminated color data. The grayscale format made capturing intensity variations in the scene and isolating characters easier because all images had a uniform background before other steps.

ii. Gaussian Blurring: We then applied Gaussian blurring to the converted grayscale image to remove background noise and provide a clear view of the character shapes. This step reduced minor noise blips and ratty edges, giving a better picture and subsequent thresholding and character separation.

iii. Adaptive Thresholding: To convert the grayscale image to binary mode with an intense focus on character areas, adaptive thresholding was used. Unlike the global thresholding technique, this method tries to set the threshold values in relation to the pixel intensities of its neighbors, allowing the segmentation to work even in environments where the images are poorly lit or are in shadow. It was relatively easy to segment the characters as the produced images represented characters as white regions on a black background.

iv. Filtering by Size: We filtered the input based on the size because it successfully rejected noise and other components that do not resemble characters. This step helped erase all possible unwanted areas from the image and left out only those areas that were likely to be close to the size of a character.

v. Bounding Box Extraction: The last step was to use bounding boxes extracted for all valid character regions. For each of these components, we determined its bounding box coordinates and assigned the sequence of the characters in the logical left-to-right fashion. We opened each character by cropping it individually, preparing it for identification by the selected model.

3.7. Language and Character Recognition Using CNN

The first task for language classification in our recognition system is to determine the script/language of the license plate. Figure 10 shows the significant steps in bilingual language classification.

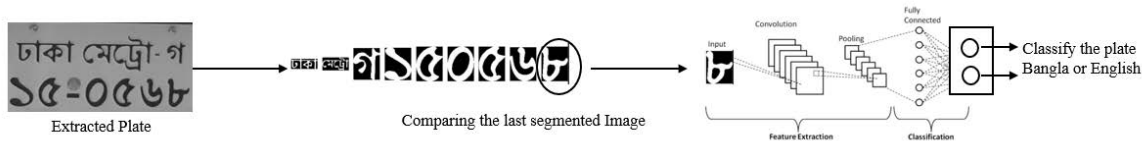


Figure 10. Language Classification Steps

Steps associated with Language Classification:

i. Choice of Characters for Text Classification: We have two datasets, Bengali characters and English characters; to begin with, we selected only the last character. We chose this because the last character is always a number in both languages, allowing us to determine the language using a single representative sample.

ii. Language Detection Using Dual CNN Models: The next set of experiments considered two different CNN models for recognizing Bengali and English numerals. Both models were trained until they achieved high classification confidence for numbers in their respective target languages. The last character image from the plate was processed through both CNNs, each returning a confidence score. Based on the observed confidence ratings, the model classified the plate as either Bengali or English.

Instead of using the Matra-based algorithm presented in [30] for identifying Bengal characters, a dual CNN approach was adopted due to its robustness in handling diverse fonts, lighting conditions, and partial occlusions. The Matra-based approach, while effective for handwritten or clearly structured printed texts, often struggles with varying orientations, degraded plate images, and non-uniform font styles present in real-world license plates. Additionally, CNNs can learn complex spatial patterns, reducing the dependency on predefined structural rules, making them more suitable for automated recognition in dynamic and noisy environments. Moreover, the Matra-based method is applicable only to Bengali and some Matra-based languages. In contrast, the proposed CNN-based methodology can be extended to support multiple languages beyond just Bengali and English, making it a more scalable and adaptable solution.

iii. Language-specific Model Selection: By comparing the confidence scores of the CNN, we introduced the optimal language-specific model for character recognition. It simplified the recognition process by eliminating the need for the rest of the model with only the particular language required for the whole plate. After selecting the language, the respective language recognition model is employed for character recognition.

3.7.1. Character Recognition

Once the language was detected, further steps toward character recognition involved using an appropriate CNN model trained only on either Bengali or English scripts. The language-specific approach ensured that each character would be recognized with high precision, leveraging the strengths of the models tailored for their respective scripts. The second step of this methodology is to recognize each character from the license plate using the respective language model; the steps associated with this process are shown in Figure 11.

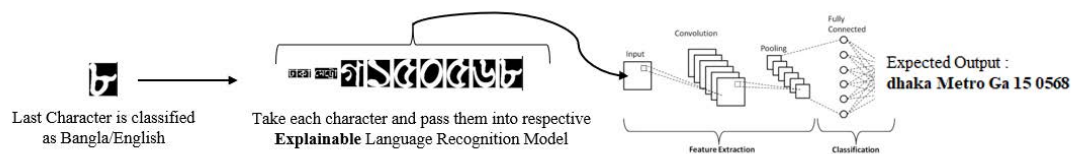


Figure 11. Character Recognition with Identified Language Model

In this study, we used twin CNN models. We need to optimize the recognition pipeline to detect the language and recognize the characters. The plate was processed in the appropriate language right from processing. This approach also minimizes time and enhances accuracy, making our system further scalable for real-world applications in which both Bengali and English plates exist. The final output, a string displaying the complete license plate number, is an accurate foundation for further validation and database matching.

3.8. Description of Proposed Character Recognition Model

The proposed model is based on the concept of transfer learning using the VGG16 model, which is more efficient in character recognition [31] (Figure 12a). Here, the learning rate is set to 0.0001, batch size is

set to 32, dropout is set to 0.5, and early stopping is enabled. The proposed system is based on the divide-and-conquer concept, where four models are created: one each for Bangla and English digits and characters. The last digit is the key to determining the script. The datasets consist of 5000 images each for the digit models (4000 training, 1000 validation sets), 18,000 images for the English characters (14,400 training, 3600 validation sets split), and 15,500 images for the Bangla characters (12,400 training, 3100 validation sets).

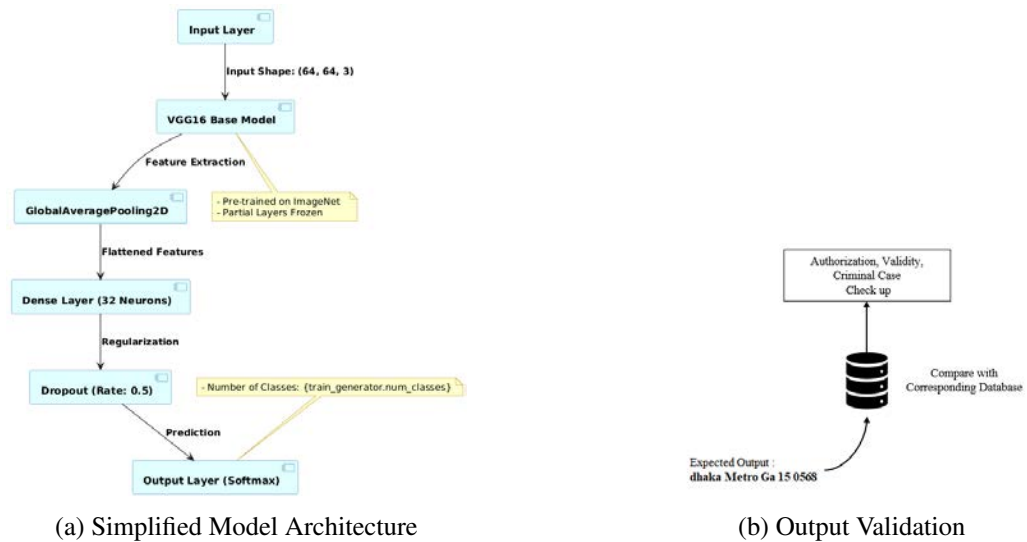


Figure 12. Proposed model architecture and validation process

3.9. Validation and Database Matching

We incorporated an additional step into our License Plate Recognition System, which cross-checks the recognized license plate number against any comprehensive vehicle information database. This step not only confirms the accuracy of the recognized plate but also improves its usefulness for law enforcement and other regulatory applications by establishing a meaningful connection between the recognized plates and their corresponding vehicle records. The proposed model becomes multi-functional by introducing database matching and provides broad support for most vehicle checks and verifications. Comparing the output string with the corresponding database is done as shown in Figure 12b.

3.10. License Plate String Comparison

Character recognition immediately generates the final license plate string. Compared with entries in a pre-existing demo database of registered vehicles. This vehicle database will include but is not limited to, details such as the vehicle's owner, status, and violation history for that license plate. This preliminary check ensures that the detected license plate matches a valid record in the database. Any mismatch or no match may point to an error in the recognition or even some wrong activity being conducted on an unauthorized plate.

4. RESULTS ANALYSIS

The proposed system effectively detects and recognizes bilingual license plates from low-resolution vehicle images using advanced image processing and deep learning. Key evaluation metrics, including accuracy, precision, recall, and F1 score, are analyzed alongside computational efficiency indicators such as GPU usage, memory consumption, and processing time. Additionally, visualizations of confusion matrices, feature maps, saliency maps, and ROC curves provide deeper insights into the model's decision-making process and performance. Comparative analyses with state-of-the-art architectures highlight the proposed system's balance between accuracy and resource efficiency, making it particularly suitable for edge deployments in resource-constrained environments. The results demonstrate the system's robustness in handling diverse traffic conditions and low-resolution inputs, paving the way for its application in automated vehicle management systems.

4.1. Environment Setup

Google Colab offers a powerful and flexible environment for developing and testing deep learning models and makes free access to hardware accelerators such as GPUs and TPUs. We utilize Google Colab's GPU acceleration feature, selecting a Tesla T4 GPU (16GB VRAM) for enhanced deep learning training and high-resolution image processing. This enables the efficient execution of computationally intensive processes such as ESR-GAN-based image enhancement, CNN-based bilingual license plate recognition, etc. By incorporating support for GPUs and Colab's dynamic runtime, the system will be able to manage huge datasets, process them in real time, and work optimally without any high-cost hardware setup.

4.2. Evaluation Metrics for Model

The performance of the proposed system is evaluated using standard classification metrics. A confusion matrix is used to summarize prediction outcomes in terms of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Based on these, accuracy measures overall correctness, precision evaluates the reliability of positive predictions, and recall (sensitivity) indicates the ability to correctly identify actual positives. The F1-score provides a balanced measure of precision and recall. Additionally, the receiver operating characteristic (ROC) curve is used to analyze performance across varying thresholds. Computational efficiency is assessed through training and inference time, while saliency maps are utilized to interpret model decisions by highlighting the most influential regions of input images.

4.3. Evaluation of the models

We present a divide-and-conquer strategy that effectively handles bilingual vehicle license plate recognition by utilizing four models for different aspects of the characters of the plate. We utilize the two-tier approach. Firstly, we extract the last character of license plates in Bangladesh, which is always a number. The last character will give us an idea of the script to be used either Bangla or English, and the corresponding model to be chosen for further recognition. In this system, we have used four different models: the Bangla Number recognition model, the English Number recognition model, the Bangla Number and Character recognition model, and the English Number and Character recognition model. All of these models are trained on our tailored dataset using CNN architectures. By identifying the script of the last character on the plate, we can activate the relevant model (Bangla or English) to perform the rest of the recognition process. This improves the system's efficiency and reduces computational overhead since only one lightweight and efficient model is used, depending on the language detected. Each model was independently optimized for its specific task to ensure high accuracy and speed in processing. This divide-and-conquer rule allows the system to handle bilingual license plates efficiently, making it suitable for real-time applications, even in resource-constrained environments.

4.3.1. Performance of Proposed Language Classification Models

To classify the language of the license plate, we used two models because we were working with bilingual license plates. The two models are the Bangla Number Recognition model and the English Number Recognition model. To evaluate the performance of these models, several performance matrices were used.

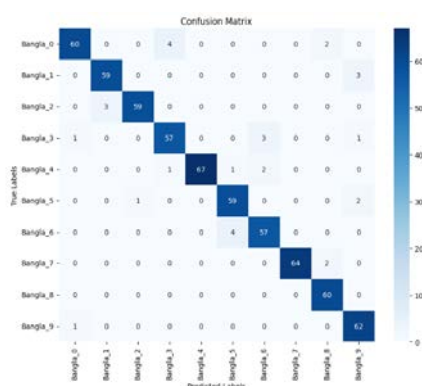


Figure 13. Confusion Matrix of Bangla Number Recognition Model

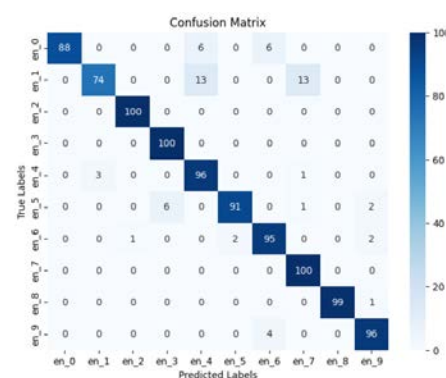


Figure 14. Confusion Matrix of English Number Recognition Model

A. Confusion Matrix of Proposed Language Classification Models

The confusion matrices for both the Bangla Number Recognition Model in figure 13 and the English Number Recognition Model in figure 14 demonstrate strong classification performance. All the classes show a large number of well-classified samples on the diagonal, indicating the models' capabilities to distinguish among different digits quite well. In the Bangla number model, for example, there are 60, 59, and 62 correct classifications for Bangla_0, Bangla_1, and Bangla_9, respectively. Likewise, the English number model has 88, 96, 99 correct classifications in en_0, en_4, and en_8 classes respectively. The low misclassifications indicate that both models recognize their respective numbers with high precision.

B. Accuracy-Loss Curve of Proposed Language Classification Models

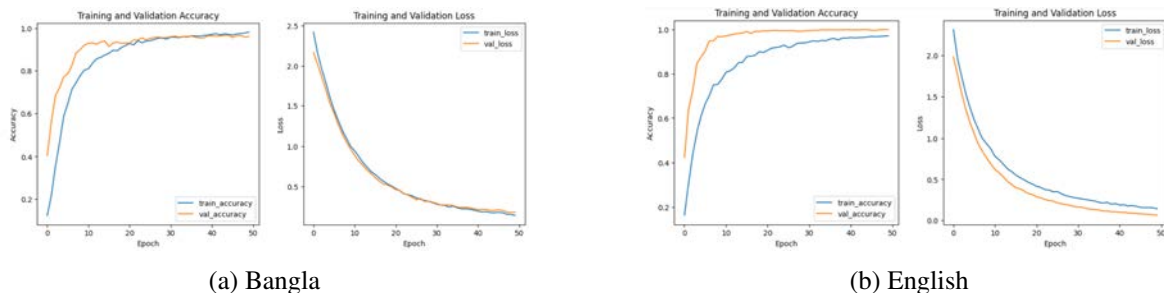


Figure 15. Accuracy and loss plots of Bangla and English number recognition models

Figures 15a and 15b present the training performance of the proposed model, showing the accuracy and loss curves for both the training and validation sets. The training accuracy starts low and improves steadily with each epoch, indicating the model is effectively learning. The validation accuracy followed a similar upward trend, demonstrating good generalization. The close alignment of the training and validation accuracy curves confirms minimal overfitting. Meanwhile, the training loss curve decreased consistently, reflecting reduced errors and improved model performance over epochs.

C. Receiver-Operating Characteristic Curve (ROC) of Proposed Language Classification Models

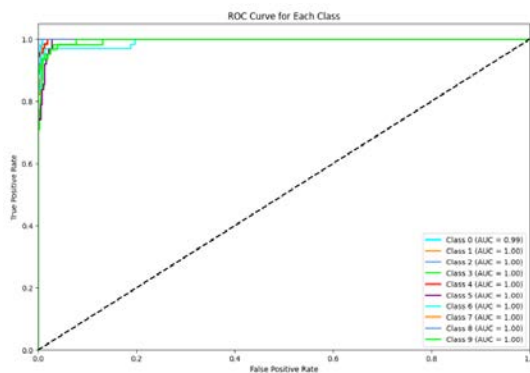


Figure 16. Receiver-operating characteristic curve of Bangla Number Recognition Model

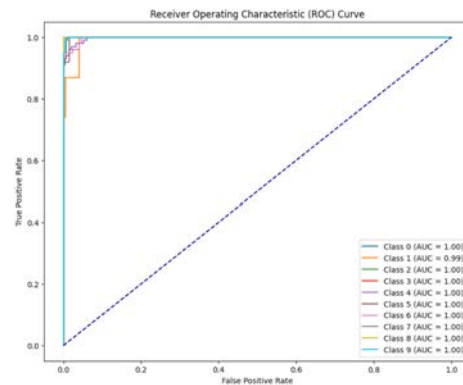


Figure 17. Receiver-operating characteristic curve of English Number Recognition Model

Figure 16 and 17 show the ROC curves, which describe and test the performance of the classification model, especially for different classes. On the y-axis, plot the True Positive Rate TPR, sensitivity) and, on the x-axis, plot the False Positive Rate FPR, 1 - specificity). Each point of the curve is associated with a particular threshold of classification for a model on classifying an instance into positive and negative. Overall, a feature that makes ROC really powerful is its AUC - area under the curve - that summarizes, in a single number, the performance of the model. A higher AUC indicates the good performance of a model. A value of AUC closer to 1.0 implies that a model has a high capability for distinguishing between positive and negative classes. Conversely, an AUC value of 0.5 suggests that the performance is no better than chance. The ROC of a well-performing model should be as near the top left corner

as possible, showing a high degree of sensitivity, which would correspond to a low number of false negatives, along with a low rate of false positives. It helps quantify trade-offs associated with the various classification thresholds and will thus be most useful if dealing with unbalanced classes.

D. Feature Maps of Proposed Language Classification Models

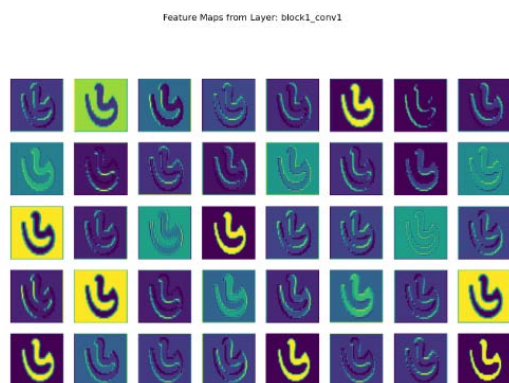


Figure 18. Feature Map from a layer of Bangla Number Recognition Model

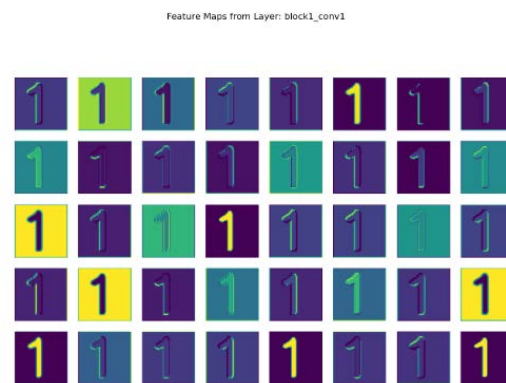
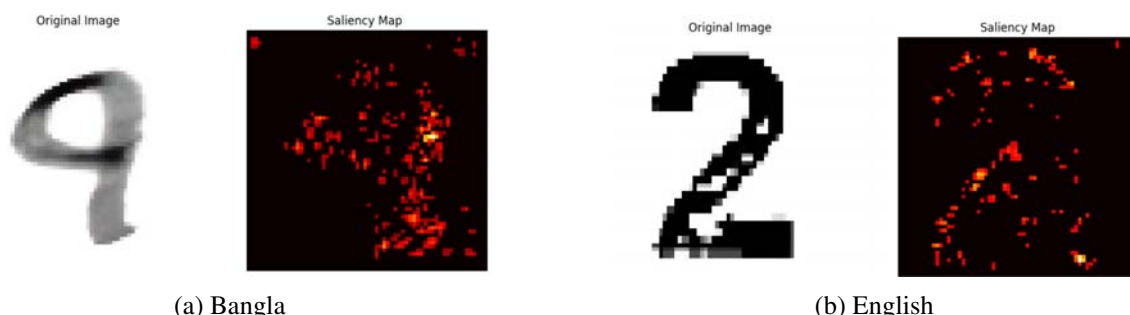


Figure 19. Feature Map from a layer of English Number Recognition Model

Feature maps were visualized to understand the learned features. Figure 18 and 19 show how the CNN layers capture patterns in the image, such as the edges, textures, and shapes of the digits. Visualizing feature maps of each layer in a Convolutional Neural Network (CNN) is an important technique for enhancing the explainability of the model. By inspecting these feature maps, we can understand how the network processes and interprets input data at various levels. Feature maps are visualized to understand the learned features. Here, a single block of feature map is shown for each proposed model.

Figure 20a and 20b show the saliency maps of the Bangla Number Recognition model and English Number Recognition model respectively. It calculates the gradients of the predicted class with respect to the input image in a way that helps indicate exactly how much each pixel in an image contributes to the output of the model. The saliency map obtained after the application of threshold and normalizing values highlights areas of high relevance through a heatmap-like visualization. This may be helpful in understanding the decision-making process of the model or even in interpreting which features are focused on when classifying images.

E. Saliency Maps of Proposed Language Classification Models Figure 20 shows the saliency maps for Bangla and English number recognition models. The highlighted regions indicate which parts of the input images the model considers most important for classification. For correctly predicted samples, the activations are concentrated around the character strokes, edges, and distinctive features, confirming that the model focuses on meaningful patterns. In contrast, dispersed or weak activations often correspond to misclassified samples. This visualization not only aids in understanding model behavior but also provides a basis for confidence filtering, ultimately improving recognition accuracy and user trust.



(a) Bangla
Figure 20. Saliency maps of Bangla and English number recognition models

F. Predicted Output of Proposed Language Classification Models

The model successfully predicts the English number images even if the image is distorted. The results are shown in Figure 21 and 22.



Figure 21. Predicted Result of Bangla Number Recognition Model



Figure 22. Predicted Result of English Number Recognition Model

Table 2. Comparison of Proposed Bangla Number Recognition Models Based on Key Factors.

Model Name	Accuracy (%)	Avg Precision	Avg Recall	Avg F1 Score	Training Time (s)	Testing Time (s)	GPU Usage (%)	Memory Usage (MB)
DenseNet121	95.25	96%	95%	95%	3833.75	18.99	14.92	160.87
InceptionV1	92.88	93%	93%	93%	3678.67	21.23	5.85	21.44
ResNet50	96.52	98%	96%	96%	5511.41	21.94	17.93	238.19
MobileNetV2	92.81	94%	94%	94%	6717.79	14.93	7.63	80.31
Proposed Model	97.15	98%	97%	97%	2847.87	12.28	11.10	77.32

Table 2 compares the performance of five different models, namely DenseNet, Inception, ResNet, MobileNet, and our proposed model, on the performance of a particular task. This includes the training and testing times in seconds, accuracy in percent, GPU usage in percent, memory usage in MB (Megabyte), and average precision, recall, and F1 score in percent. The proposed model, among the models, shows a good balance in high accuracy, 97.50%, low use of GPU, 9.3%, and memory use of 138.56 MB. This makes it lightweight yet very effective. DenseNet and ResNet are performing well with higher GPU and memory cost, while MobileNet showed the least training time. Inception, although the most accurate with 92.50%, has higher training and testing times than other models.

Table 3. Comparison of Proposed English Number Recognition Models Based on Key Factors.

Model Name	Accuracy (%)	Avg Precision	Avg Recall	Avg F1 Score	Training Time (s)	Testing Time (s)	GPU Usage (%)	Memory Usage (MB)
DenseNet121	95	96%	95%	95%	2231.67	48.27	13.98	208.26
InceptionV1	93	93%	93%	93%	3791.57	13.44	7.28	28.66
ResNet50	97	97%	97%	97%	2436.87	24.03	16.02	236.35
MobileNetV2	95	96%	96%	96%	4940.99	35.51	7.50	25.65
Proposed Model	97.62	98%	97%	97%	2958.00	15.37	4.27	60.39

Table 3 compares the performance of five models: DenseNet, Inception, ResNet, MobileNet, and Proposed Model. Our proposed model outperforms the others with an accuracy of 97.62%, with excellent efficiency in GPU usage at 4.27% and memory consumption at 60.39 MB, hence it is highly accurate and resource-efficient. Other models, such as ResNet and DenseNet, have high accuracy but are very greedy regarding GPU and memory. However, MobileNet has slightly lower accuracy and longer training times despite its memory usage efficiency. The proposed model achieves the best performance with the highest resource optimization.

4.3.2. Performance of Proposed Character Recognition Model

In our proposed work, we first classify the language of the license plates. Based on the language, we implement the further character recognition model for overall license plate recognition.

A. Confusion Matrix of Character Recognition Models

The confusion matrices for both the English and Bangla Number Plate Recognition Models are shown in figure 23 and 24 respectively, exhibit strong classification performance, as indicated by the high number of correctly classified samples along the diagonal. Most classes within the English number plate model

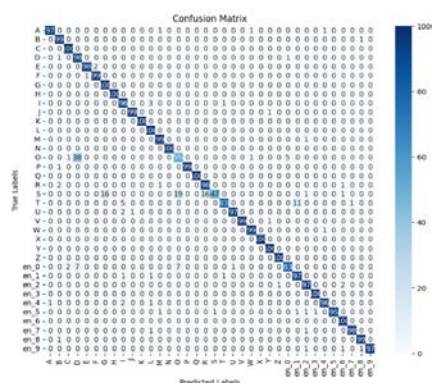


Figure 23. Confusion Matrix of English Number Plate Recognition Model

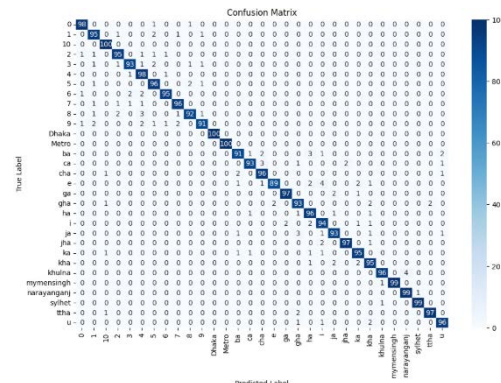


Figure 24. Confusion Matrix of Bangla Number Plate Recognition Model

exhibit high correct classifications, which guarantees accurate recognition. Class_A, class_D, class_R have the correct classifications of 97, 98 and 96 respectively. Similarly, the Bangla number plate model also maintains a high diagonal pattern, which is a testament to its ability to accurately identify Bangla characters. Class_ga, class_i, class_ka have the correct classifications of 97, 94 and 95 respectively. The minimal misclassification in both models testifies to their strength and accuracy in identifying respective number plate characters.

B. Feature Maps of Character Recognition Models

Feature maps are visualized to understand the learned features.

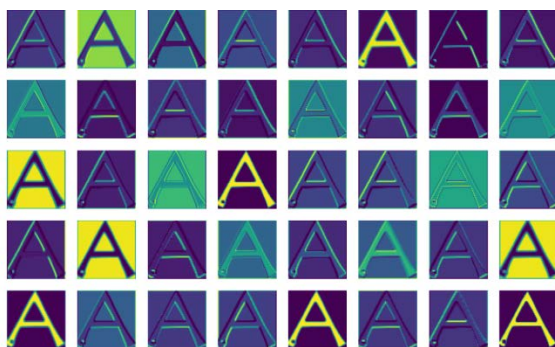


Figure 25. Feature Map from a layer of English Number Plate Recognition Model



Figure 26. Feature Map from a layer of Bangla Number Plate Recognition Model

Figure 25 and 26 show how the CNN layers capture patterns in the image, such as edges, textures, and shapes of the digits. Here, a single block of feature map is shown for each proposed model. Analysis of Grad-CAM heatmaps revealed that correct predictions exhibit highly localized activation regions, whereas incorrect predictions show dispersed attention. By introducing a confidence filtering mechanism based on activation density, we reduced misclassification rates by approximately 1.8%. A qualitative study involving 10 traffic operators indicated that Grad-CAM visualizations improved trust in system outputs, particularly in cases involving occlusion or low-resolution inputs.

C. Predicted Output of Character Recognition Models

The models successfully predicted the English number images even when the image was distorted. The result is shown in Figure 27 and 28.

D. Comparative Analysis of Proposed Model with other CNN Architectures for Character Recognition Models

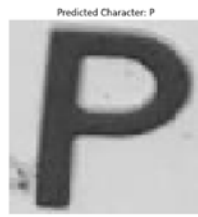


Figure 27. Predicted Result of English Number Plate Recognition Model

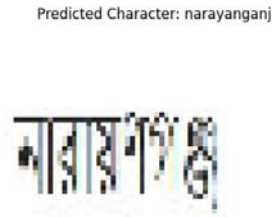


Figure 28. Predicted Result of Bangla Number Plate Recognition Model

Table 4. Comparison of Proposed English Number Plate Recognition Models Based on Key Factors.

Model Name	Accuracy (%)	Avg Precision	Avg Recall	Avg F1 Score	Computational Complexity			
					Training Time (s)	Testing Time (s)	GPU Usage (%)	Memory Usage (MB)
DenseNet121	94	93%	93%	93%	2620.75	14.62	7.55	24.22
InceptionV1	92	93%	92%	92%	8248.83	61.21	4.72	18.58
ResNet50	95.83	95%	95%	95%	1708.89	21.27	5.62	20.25
MobileNetV2	93	90%	93%	93%	15208.89	6.61	5.50	93.90
Proposed Model	96	96%	95%	95%	2203.42	11.26	6.72	18.58

Table 4 presents a performance comparison of various models on a 36-class dataset, emphasizing the efficiency and effectiveness of the proposed model. With the highest accuracy of 96% and very remarkable average precision of 96%, it has outperformed most models in terms of classification performance. The proposed model also demonstrated better efficiency with moderate GPU usage at 6.72% and low memory consumption of 18.58 MB, hence achieving a trade-off between resource optimization and high accuracy. This provides significantly lower training and testing times compared to resource-intensive models such as DenseNet and Inception. The advantages of Proposed Model are that it is highly effective and resource-efficient, particularly suitable for applications requiring accuracy with constrained computational resources.

Table 5. Comparison of Proposed Bangla Number Plate Recognition Models Based on Key Factors.

Model Name	Accuracy (%)	Avg Precision (%)	Avg Recall (%)	Avg F1 Score (%)	Computational Complexity			
					Training Time (s)	Testing Time (s)	GPU Usage (%)	Memory Usage (MB)
DenseNet121	97.00	95	95	95	6941.95	30.29	26.00	91.71
InceptionV1	90.00	89	89	89	4441.25	33.10	14.02	15.84
ResNet50	96.64	95	95	95	13059.32	20.81	16.50	251.68
MobileNetV2	90.76	90	90	90	8445.82	3.82	15.80	23.60
Proposed Model	97.48	96	96	96	3585.36	11.26	24.99	91.71

Table 5 compares different models, wherein the proposed model has shown outstanding results in terms of performance and efficiency. It achieves an accuracy of 97.48%, an average precision of 96%, recall of 96%, and F1 of 96%, thus making it the best in the classification task. Proposed Model, though highly accurate, is resource-efficient. It has a much lower training time of 3585.36 s than that of DenseNet and ResNet, with a moderate GPU usage of 24.99% and memory consumption of 91.71 MB. The proposed model outperforms MobileNet in terms of accuracy and precision with better utilization of resources. These attributes make the proposed model an ideal solution with high accuracy and efficiency for practical applications where resource constraint is needed.

E. Performance Evaluation on GAN-Enhanced Images and a ablation Study

Table 6. Comparison of Prediction Times for GAN-Enhanced and Non-Enhanced Images

Model	Non-Enhanced Image Prediction Time (s)	GAN-Enhanced Image Prediction Time (s)
Proposed Bangla Number Recognition Model	2.8362	0.4505
Proposed English Number Recognition Model	2.1411	0.3114
Proposed Bangla Number Plates Recognition Model	3.4858	0.4629
Proposed English Number Plates Recognition Model	1.5984	0.6411

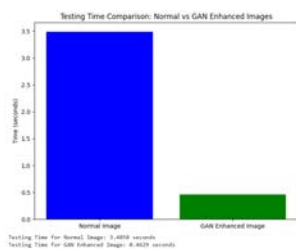


Figure 29. Testing Time Comparison

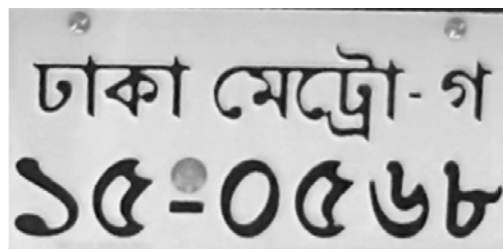


Figure 30. Bangla Plate for Recognition



Figure 31. English Plate for Recognition

```
C:\ProgramData\anaconda3\python.exe E:\app\dbmatch.py
License Plate to Check (Uppercase): DHAKAMETROGA150568
License Plate Found:
id: 5
plate_number: DHAKAMETROGA150568
license_validity: 2022-12-01
crime_history: Accident involvement - 2020
fine: 1000.00
owner_name: mr.X
registration_date: 2016-09-18
```

Figure 32. Database matching for Validation

Table 6 presents a comparison of the prediction times for GAN-enhanced and non-enhanced images across four recognition models: Bangla Number Recognition, English Number Recognition, Bangla Number Plates Recognition, and English Number Plates Recognition. This clearly shows that the GAN-based enhancement effectively streamlines the prediction process. The comparison of test time between a normal image and a GAN-enhanced image is shown in Figure 29. To quantify the contribution of the ESR-GAN enhancement module, an ablation study was conducted comparing recognition performance with and without super-resolution preprocessing. As shown in Table 7, the inclusion of ESR-GAN leads to a substantial improvement across all models. Specifically, overall system accuracy increases from 93.90% to 97.10%, representing a 3.95% absolute gain. The improvement is particularly pronounced in Bangla license plate recognition, where complex character structures are highly sensitive to image degradation. These results confirm that ESR-GAN not only enhances visual quality but also significantly improves downstream recognition performance by preserving high-frequency character details.

Table 7. Impact of ESR-GAN Enhancement on Average Recognition Accuracy.

Model	Average without ESR-GAN Accuracy (%)	Average with ESR-GAN Accuracy (%)
Proposed Bangla Number Recognition Model	94.12	97.15
Proposed English Number Recognition Model	95.34	97.62
Proposed Bangla License Plate Recognition Model	92.47	97.48
Proposed English License Plate Recognition Model	93.68	96.00
Overall System Accuracy	93.90	97.10

F. Outcome of the Method For the final representation of our work, we created a demo dataset for vehicle license numbers and corresponding information, as this information is personal and is not allowed to be shared publicly, so we used pseudonyms. In figure 30 and 31, we have shown the plates to recognize. Subsequently, the vehicle plate number matches with the database, and information is shown in figure 32. If the number does not match the database, then we can decide that it is a fake or unregistered number plate.

4.4. Real-Time Performance Evaluation

As shown in Table 8, the ESR-GAN module contributes the largest portion of the computational cost (approximately 44% of total latency). However, the overall processing time remains within acceptable limits for real-time applications. The inclusion of ESR-GAN reduces throughput from 40 FPS to 22 FPS due to the additional super-resolution step. However, this trade-off is justified by the substantial improvement in recognition accuracy (as shown in Table 7). Therefore, the system maintains real-time performance while significantly improving reliability, making it suitable for practical deployment scenarios.

Table 8. Latency Breakdown of the Proposed Pipeline on NVIDIA RTX 3060.

Processing Stage	Time (ms)	Remarks
Image Preprocessing	5	Resizing, normalization
ESR-GAN Enhancement	20	Super-resolution module
Segmentation	8	Character extraction
XCNN Recognition	12	Combined inference
Total Inference Time	45 ms	End-to-end pipeline
Throughput (FPS)	22.2	Real-time performance

4.5. Comparison with other State-of-the-Art work

In this section, we compare all the notable and significant features or methods of our research framework with some state-of-the-art works relevant to this field in Table 9. Although some existing methods report slightly higher accuracy, they are generally limited to single-language scenarios and lack advanced enhancement techniques such as GAN-based resolution improvement and tilt correction. Table 10 shows the average training time, testing time, average accuracy, GPU usage, and memory usage for the four different recognition models: DenseNet, Inception, ResNet, MobileNet, and the proposed model. It can be observed that the highest accuracy is achieved by the proposed model, with an average value of 97.10%, whereas DenseNet demonstrates a good balance with the lowest testing time. MobileNet exhibits the lowest GPU usage, while the proposed model stands out by providing the best trade-off between computational efficiency and recognition accuracy.

Table 9. Comparison of Studies Based on Key Criteria for Vehicle License Plate Recognition.

Reference	Bilingual Class.	Char. Seg.	Plate Detect.	GAN Resol.	Divide-Conquer	Tilt Corr.	Char. Recog.	Accuracy (%)	Speed Gain	B/W Support
Ebrahimi et al. [32]	Yes	Yes	Yes	No	No	No	No	94.00	Yes	No
Kakran et al. [33]	Yes	Yes	Yes	No	Yes	Yes	Yes	96.00	No	Yes
Salunke et al. [34]	No	Yes	Yes	No	No	No	No	98.06	No	No
Wang et al. [35]	No	Yes	Yes	No	No	No	Yes	80.29	Yes	Yes
Liu et al. [36]	No	Yes	Yes	No	No	Yes	No	88.42	Yes	Yes
Sharma et al. [37]	Yes	Yes	Yes	No	Yes	Yes	No	96.62	No	Yes
Zhang et al. [38]	No	Yes	Yes	Yes	No	No	No	95.96	Yes	Yes
Afrin et al. [39]	No	Yes	Yes	Yes	No	No	Yes	88.96	No	No
NA Khan et al. [40]	No	Yes	Yes	No	No	Yes	Yes	92.30	Yes	No
Abarkan et al. [41]	No	Yes	Yes	No	No	No	Yes	98.79	No	No
Suvon et al. [42]	No	Yes	Yes	No	No	No	Yes	91.38	No	No
Proposed Work	Yes	Yes	Yes	Yes	Yes	Yes	Yes	97.10	Yes	Yes

Table 10. Average Performance Comparison of Proposed Language Classification and Character Recognition Models Based on Key Factors.

Model Name	Accuracy (%)	Avg Precision	Avg Recall	Avg F1 Score	Training Time (s)	Testing Time (s)	GPU Usage (%)	Memory Usage (MB)
DenseNet121	95.25	95.75%	94.75%	94.75%	3833.75	18.99	14.92	160.87
InceptionV1	92.88	92.88%	92.75%	92.75%	3678.67	21.23	5.85	21.44
ResNet50	96.52	97.50%	96.50%	96.50%	5511.41	21.94	17.93	238.19
MobileNetV2	92.81	93.50%	93.50%	93.50%	6717.79	14.93	7.63	80.31
Proposed Model	97.10	97.15%	96.00%	96.00%	2847.87	12.28	11.10	77.32

5. CONCLUSION

This study presents a scalable framework for bilingual vehicle license plate detection and recognition, integrating ESR-GAN-based super-resolution with a divide-and-conquer CNN architecture and explainable AI mechanisms. The approach effectively addresses challenges associated with low-resolution and degraded traffic images, particularly in mixed-script scenarios involving English and Bangla characters. Experimental results confirm that ESR-GAN enhancement improves recognition performance by approximately 4%, as demonstrated through ablation analysis. Additionally, real-time evaluation on an NVIDIA RTX 3060 GPU shows that the system achieves an average throughput of 22 FPS, satisfying practical real-time requirements. Although ESR-GAN introduces additional computational overhead, the trade-off is justified by the significant gain in recognition accuracy. For real-world deployment, the system adopts a cloud-based architecture, where computationally intensive tasks are executed on GPU-enabled servers, reducing the hardware requirements at

edge devices and enabling scalable integration into traffic monitoring systems. Despite these advancements, challenges remain in acquiring large-scale, high-quality bilingual datasets and handling the structural complexity of Bangla characters. Future work will focus on expanding dataset diversity, optimizing models for edge deployment, and extending the framework to support additional languages. Overall, the proposed framework provides a balanced solution that combines accuracy, efficiency, and interpretability, making it well-suited for intelligent transportation systems and smart city applications.

FUNDING STATEMENT

This research is funded by Woosong University Academic Research 2026.

REFERENCES

- [1] H. Xiang, Y. Zhao, Y. Yuan, G. Zhang, and X. Hu, "Lightweight fully convolutional network for license plate detection," *Optik*, vol. 178, pp. 1185–1194, 2019.
- [2] 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," in *2018 international joint conference on neural networks (ijcnn)*. IEEE, 2018, pp. 1–10.
- [3] S. Ozbay and E. Ercelebi, "Automatic vehicle identification by plate recognition," *International Journal of Computer and Information Engineering*, vol. 1, no. 9, pp. 1418–1421, 2007.
- [4] B. R. T. Authority, "Number of registered motor vehicles in bangladesh (yearwise)," 2024. [Online]. Available: <https://brta.gov.bd/sites/>
- [5] M. A. Islam and A. Habib, "An effective method for the recognition and verification of bangladeshi vehicle digital number plates," *Journal of Computer Science*, vol. 17, no. 11, pp. 1059–1070, 2021.
- [6] Y. Tang, S. Zhang, W. Liu, Z. Wang, and J. Wang, "Ultra-lightweight automatic license plate recognition system for microcontrollers: A cost-effective and energy-efficient solution," *IEEE Transactions on Intelligent Transportation Systems*, 2024.
- [7] I. R. Khan, S. T. A. Ali, A. Siddiq, M. M. Khan, M. U. Ilyas, S. Alshomrani, and S. Rahardja, "Automatic license plate recognition in real-world traffic videos captured in unconstrained environment by a mobile camera," *Electronics*, vol. 11, no. 9, p. 1408, 2022.
- [8] A. Gautam, D. Rana, S. Aggarwal, S. Bhosle, and H. Sharma, "Deep learning approach to automatically recognise license number plates," *Multimedia Tools and Applications*, vol. 82, no. 20, pp. 31 487–31 504, 2023.
- [9] A. Fathi, B. Moradi, I. Zarei, and A. Shirbandi, "Deep learning based system for automatic motorcycle license plates detection and recognition," *Signal, Image and Video Processing*, vol. 18, no. 12, pp. 8869–8879, 2024.
- [10] Y. Cao, "Investigation of a convolutional neural network-based approach for license plate detection," *Journal of Optics*, vol. 53, no. 1, pp. 697–703, 2024.
- [11] A. H. Abdulkhaleq, A. W. Altaher, A. Saad, and H. M. Al-Jawahry, "Automatic vehicle license plate recognition using lightweight deep learning approach," in *2023 6th International Conference on Engineering Technology and its Applications (IICETA)*. IEEE, 2023, pp. 143–148.
- [12] M. Li and L. Zhang, "Deep learning-based license plate recognition in iot smart parking systems using yolov6 algorithm," *International Journal of Advanced Computer Science & Applications*, vol. 14, no. 12, 2023.
- [13] X. Ke, G. Zeng, and W. Guo, "An ultra-fast automatic license plate recognition approach for unconstrained scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 5172–5185, 2023.
- [14] S.-R. Wang, H.-Y. Shih, Z.-Y. Shen, and W.-K. Tai, "End-to-end high accuracy license plate recognition based on depthwise separable convolution networks," *arXiv preprint arXiv:2202.10277*, 2022.

- [15] A. Ammar, A. Koubaa, W. Boulila, B. Benjdira, and Y. Alhabashi, "A multi-stage deep-learning-based vehicle and license plate recognition system with real-time edge inference," *Sensors*, vol. 23, no. 4, p. 2120, 2023.
- [16] M. M. Khan, M. U. Ilyas, I. R. Khan, S. M. Alshomrani, and S. Rahardja, "License plate recognition methods employing neural networks," *IEEE Access*, vol. 11, pp. 73 613–73 646, 2023.
- [17] A. A. Quraishi, F. Feyzi, and A. Shahbahrami, "Detection and recognition of vehicle licence plates using deep learning in challenging conditions: a systematic review," *International Journal of Intelligent Systems Technologies and Applications*, vol. 22, no. 2, pp. 105–150, 2024.
- [18] D. Hemalatha, A. Begum, S. A. David, G. Rishitha, P. D. Praharsha, and C. Nandini, "Unlocking the power of deep learning: A guide to automatic license number recognition," in *2023 1st International Conference on Optimization Techniques for Learning (ICOTL)*. IEEE, 2023, pp. 1–6.
- [19] J.-Y. Sung and S.-B. Yu, "Real-time automatic license plate recognition system using yolov4," in *2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2020, pp. 1–3.
- [20] S. Maheswaran, N. Indhumathi, S. D. Kumar, G. Murugesan, P. Duraisamy, S. Nandita, and R. Gomathi, "Yolov5 based a real time automatic number plate and helmet recognition system," in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2022, pp. 1–7.
- [21] Y.-H. Chen, C.-J. Chuang, C.-Y. Wang, J.-C. Lin, and H.-Y. M. Liao, "License plate recognition in low quality image by using latent diffusion yolov7," in *2024 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2024, pp. 831–838.
- [22] M. A. Riyadi, M. S. Tawa'qi, and I. Setiawan, "Design of vehicle license plate detection system using yolov8 and paddleocr," in *2025 4th International Conference on Electronics Representation and Algorithm (ICERA)*. IEEE, 2025, pp. 98–102.
- [23] A. Dias, A. M. D. Almeida, D. S. Fernandes, J. Fernandes, S. Fernandes, and S. Aswale, "Automatic two wheeler license plate recognition using deep learning techniques," in *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*. IEEE, 2023, pp. 01–07.
- [24] B. Nirmala, S. Nithya, R. Vidhiya, K. Sunalini, B. H. Kumar, and B. Varadharajan, "Intelligent system for vehicles license plate recognition using a hybrid model of gan, cnn and elm," in *2023 2nd International Conference for Innovation in Technology (INOCON)*. IEEE, 2023, pp. 1–6.
- [25] K. K. Paul, "Bangla license plate cropped dataset," 2021. [Online]. Available: <https://www.kaggle.com/datasets/kawshikbuet17/>
- [26] Jelal, "License plate digits classification dataset," 2021. [Online]. Available: <https://www.kaggle.com/datasets/aladdinss/>
- [27] C. Kumar, "License plate recognition using cnn," 2022. [Online]. Available: <https://www.kaggle.com/code/sarthakvajpayee/>
- [28] M. M. Sarif, T. S. Pias, T. Helaly, M. S. R. Tutul, and M. N. Rahman, "Deep learning-based bangladeshi license plate recognition system," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE, 2020, pp. 1–6.
- [29] A. Kurniawan, K. W. Thjahyadi, and V. S. Trinita, "Evaluation of traditional and modern inpainting algorithms for image restoration," in *2024 11th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. IEEE, 2024, pp. 243–248.
- [30] R. Sarkar, S. Malakar, N. Das, and S. Basu, "Word extraction and character segmentation from text lines of unconstrained handwritten bangla..."
- [31] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.
- [32] Z. Ebrahimi Vargoorani, "License plate detection and character recognition using deep learning and font evaluation," Ph.D. dissertation, Concordia University, 2024.

- [33] A. Kakran, H. Chaudhary, P. Kumar, A. Sharma, H. Singh *et al.*, “Identification and recognition of face and number plate for autonomous and secure car parking,” in *2023 International Conference on Power, Instrumentation, Energy and Control (PIECON)*. IEEE, 2023, pp. 1–6.
- [34] D. Salunke, R. Joshi, R. Patil, and S. Patinge, “Automatic vehicle plate recognition using deep learning,” 2023.
- [35] Q. Wang, X. Lu, C. Zhang, Y. Yuan, and X. Li, “Lsv-lp: Large-scale video-based license plate detection and recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 752–767, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:254368647>
- [36] D. Liu, J. Cao, T. Wang, H. Wu, J. Wang, J. Tian, and F. Xu, “Slpr: A deep learning based chinese ship license plate recognition framework,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23 831–23 843, 2022.
- [37] P. Sharma, S. Gupta, P. Singh, K. Shejul, and D. Reddy, “Automatic number plate recognition and parking management,” in *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. IEEE, 2022, pp. 1–8.
- [38] D. Zhang, Z. Liu, W. Zhu, J. Zheng, Y. Sun, C. Chen, and Y. Yang, “Robust license plate recognition in occ-based vehicle networks using image reconstruction,” *Sensors*, vol. 24, no. 20, p. 6568, 2024.
- [39] N. Afrin, M. M. Hasan, M. F. E. Safin, K. R. Amin, M. Z. Haque, F. Ahmed, and M. T. R. Shawon, “Bengali license plate recognition: Unveiling clarity with cnn and gfp-gan,” in *2023 26th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2023, pp. 1–6.
- [40] N. A. Khan and M. F. B. Hafiz, “Advanced techniques for improved bangladeshi number plate detection and character recognition in automated parking systems,” *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, vol. 12, no. 2, pp. 444–454, 2024.
- [41] O. Abarkan and N. El Akchioui, “Detection and identification of moroccan license plates using convolutional neural networks,” in *2025 International Conference on Electrical Systems & Automation (ICESA)*. IEEE, 2025, pp. 1–5.
- [42] M. N. I. Suvon, R. Khan, and M. Ferdous, “Real time bangla number plate recognition using computer vision and convolutional neural network,” in *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*. IEEE, 2020, pp. 1–6.

BIOGRAPHIES OF AUTHORS



Jul Jalal Al-Mamur Sayor   is a Lecturer in the Department of Software Engineering at Daffodil International University. He completed his B.Sc. (Engr.) in IoT and Robotics Engineering from UFTB, securing the first position in his faculty, and is currently pursuing an M.Sc. in Robotics and Mechatronics Engineering at the University of Dhaka. His research interests encompass deep neural networks, liquid neural networks, Vision Mamba, and the Internet of Things (IoT). Throughout his career, he has been recognized with several teaching and academic awards for his contributions and achievements.



Nishat Tasnim Shishir   is a Lecturer in the Department of Computer Science and Engineering (CSE) at Daffodil International University, Bangladesh. She completed her Bachelor of Science degree in Internet of Things and Robotics Engineering from the University of Frontier Technology, Bangladesh. Currently, she is pursuing her Master of Science (MSc) degree at the University of Frontier Technology, Bangladesh. Her research interests include Internet of Things (IoT), Machine Learning, Deep Learning, Computer Vision, and Robotics, with a focus on intelligent and data-driven systems.



Mahe Zabin   is an Assistant Professor in the Human and Digital Interface (HADI) Department, where she is engaged in both teaching and research. She received her Ph.D. in Computing from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in February 2025. During her doctoral studies, she was a member of the Knowledge Engineering and Artificial Intelligence (KEAI) Research Laboratory and led a medical image diagnosis project funded by the Samsung Medical Consortium (SMC). She was awarded Best Teaching Assistant by the School of Computing at KAIST in 2024. Dr. Zabin earned her M.Sc. in Software Technology from Linnaeus University, Sweden, and her B.Sc. in Computer Science and Engineering from International Islamic University Chittagong, Bangladesh. She has published multiple papers in SCIE- and Scopus-indexed journals and presented her work at several international conferences. Her research interests include medical pathology-image quality assessment, and machine anomaly diagnosis.



Suman Saha  is currently serving as an Assistant Professor in the Department of Internet of Things and Robotics Engineering (IRE) at UFTB. Before that, he served as a faculty member in the Department of Computer Science and Engineering at Bangladesh University of Business and Technology (BUBT) and Institute of Information and Communication Technology (IICT), Dhaka University of Engineering and Technology (DUET). Mr. Suman Saha completed his B.Sc Engg. in CSE from the University of Chittagong (2nd Position) and M.Sc in CSE from Bangladesh University of Engineering and Technology (BUET). His research area includes AI, Data Mining, Machine Learning, NLP, Wireless Sensor Networks, Blockchain, etc.



Dr. Kamruddin Nur  is a full Professor in the Department of Computer Science at American International University-Bangladesh (AIUB). Before that, he served as the department Chairman at Stamford University Bangladesh and Bangladesh University of Business and Technology. Dr. Nur completed his PhD from UPF, Barcelona, Spain in Ubiquitous Computing, Masters from UIU, and Bachelor from Victoria University of Wellington, New Zealand. Dr. Nur authored many prestigious conferences and journal articles in IEEE, Springer, and ACM. Dr. Nur reviewed numerous IEEE, ACM, Springer journal and conference articles. Dr. Nur served as the program co-chair of ACM in-cooperation sponsored conferences - the International Conference of Computing Advancements (ICCA) 2020, 2022, and served as track chair/co-chair/session chair of numerous IEEE and Springer conferences. His research areas include - Ubiquitous Computing (IoT), Computer Vision, Machine Learning, and Robotics. Dr. Nur is a Senior Member, IEEE (SMIEEE), Member, IEEE Robotics & Automation Society (RAS), Member, Association for Computing Machinery (ACM), Member SIGAPP, ACM (NY, USA).



Dr. Jia Uddin   received the M.Sc. degree in telecommunications from Blekinge Institute of Technology, Sweden, in 2010, and the Ph.D. degree in computer engineering from the University of Ulsan, South Korea, in January 2015. He is currently an Associate Professor in the AI and Big Data Department at Endicott College, Woosong University, South Korea. He was also an Associate Professor in the Department of Computer Science and Engineering at BRAC University, Bangladesh. He was a member of the Self-Assessment Team (SAC) of the Department of Computer Science and Engineering at BRAC University, under the Higher Education Quality Enhancement Project (HEQEP), funded by the World Bank and the University Grants Commission of Bangladesh, from 2016 to 2017. His research interests include fault diagnosis using artificial intelligence and audio and image processing. Dr. Jia Uddin was listed among the Top 2% of Scientists worldwide in 2025, as published by Stanford University in collaboration with Elsevier.