Proceedings of 15th IOE Graduate Conference
Peer Reviewed
Year: 2024 Month: May Volume: 15
ISSN: 2350-8914 (Online), 2350-8906 (Print)

Automatic Number Plate Detection and Recognition Using YOLOv8 and CNN

Kshitiz Gajurel [a], Manoj Kumar Baniya [b], Khushilal Mahato [c], Manish Kathet [d], Suwarna Lingden [e], Pukar Karki [f]

[a, b, c, d, e, f] *Department of Electronics and Computer Engineering, Purwanchal Campus, IOE, Tribhuvan University, Nepal*

[a] 077bct042@ioepc.edu.np, [b] 077bct046@ioepc.edu.np, [c] 077bct039@ioepc.edu.np, [d] 077bct044@ioepc.edu.np, [e] suwarna_mise004@ioepc.edu.np, [f] pukar@ioepc.edu.np

Abstract

This paper introduces a tailored Automatic Number Plate Detection and Recognition (ANPDR) framework designed specifically for the unique challenges of the Nepali context. The framework leverages the YOLOv8 model for efficient plate detection, coupled with Byte-sort Tracking for precise detection and tracking of vehicles and plates within video frames. A preprocessing stage further enhances the system's adaptability by isolating characters from plates, effectively addressing variations in environmental conditions. In character recognition, a pivotal aspect of ANPDR, Convolutional Neural Networks (CNNs) are employed, with a particular emphasis on recognizing Devanagari characters commonly found on Nepali number plates. The CNN model, trained on a customized dataset, ensures accurate recognition even under challenging conditions. Additionally, post-processing techniques are implemented to strengthen the system's robustness and reliability. The results of this study showcase the comprehensive ANPDR solution's effectiveness. The YOLOv8 model achieves high accuracy in plate detection, with precision of 91% demonstrating its robust performance. Similarly, the CNN-based character recognition model achieves outstanding results, with an accuracy of 90%, highlighting its effectiveness in accurately identifying individual characters on detected number plates.

Keywords

YOLOv8, Byte-sort tracking, character segmentation, CNN, ANPDR

1. Introduction

The modernization of transportation management, law enforcement, and surveillance systems has been significantly propelled by the emergence of Automatic Number Plate Detection and Recognition (ANPDR) technologies. However, despite their widespread adoption globally, the applicability of existing ANPDR systems to the unique context of Nepal remains limited. The rapidly increasing traffic volume coupled with the necessity for effective law enforcement necessitates tailored ANPDR solutions that address the specific challenges prevalent in the Nepali environment.

This paper addresses a pressing scientific problem: the lack of robust ANPDR systems customized for the intricacies of Nepali numbered number plates. While vehicle detection and number plate recognition have been extensively researched and implemented in various countries, the specific characteristics of Nepali plates pose unique challenges, particularly in character matching. Traditional template matching [1] techniques, often employed in other contexts, are inadequate for handling the diverse range of Nepali number plate designs.

To bridge this gap, this methodology encompasses a comprehensive approach. Firstly, we employ tracking techniques for efficient plate localization within video frames, without specific values to evaluate the tracking as it depends on the individual implementation. Subsequently, we focus on character segmentation of the detected plate, a crucial step in ANPDR, which allows for accurate identification of individual characters. Finally, Convolutional Neural Networks (CNNs) are leveraged for character recognition, with a particular emphasis on accurately identifying Devanagari characters commonly found on Nepali plates. The CNN model, trained on a customized dataset, ensures accurate recognition even under challenging conditions, thereby overcoming the limitations of previous research efforts.

The results of this study highlight the effectiveness of the proposed ANPDR solution. The YOLOv8 [2] model demonstrates high accuracy in plate detection, with metrics such as Mean Average Precision (mAP), precision ,and recall reaching 68%, 81% and 87% respectively for plate detection. Similarly, the CNN-based character recognition model achieves outstanding results, with an accuracy of 90% for character recognition, underscoring its effectiveness in accurately identifying individual characters on detected number plates.

Beyond the technical aspects, the contributions of this research extend to practical applications in traffic management, law enforcement, and surveillance in Nepal. By providing a tailored ANPDR solution, this framework offers tangible contributions to the development of safer transportation infrastructures in the region. Moreover, the dataset utilized in this study, comprising a diverse range of images, facilitates robust training and validation procedures, further enhancing the applicability and effectiveness of the proposed framework in real-world scenarios. Overall, this research addresses a critical scientific problem and delivers solutions with significant implications for the advancement of ANPDR technology in the Nepali context.

2. Literature Review

Automatic Number Plate Recognition has been a significant area of research with various approaches explored to achieve accurate and efficient recognition systems. One notable research focused on Nepali License Plate Recognition using Support Vector Machines (SVM) [3]. This research developed a machine learning model based on SVM based learning to recognize and prediction on calculated Histograms of Oriented Gradients (HOG) features from each character.

The detection and recognition of Devanagari characters in Nepalese license plates have advanced significantly. One notable study employed the IWPOD-NET model for LP extraction and CNN models for character recognition, achieving high accuracy in both stationary and moving vehicles [4].

Automated Vehicle Number Plate Detection and Recognition has seen significant advancements. A comprehensive study utilized image processing and CNNs for precise number plate detection and character recognition, employing super-resolution methods and bounding box isolation. The system, aimed at law enforcement and parking management, demonstrated high accuracy and efficiency, supported by a developed user interface for practical application [5].

Lastly, a recent study compared the efficacy of different CNN architectures like AlexNet, LeNet-5, modified LeNet-5, and ResNet-50 for ANPR. The research found ResNet-50 and modified LeNet-5 to be particularly promising in terms of accuracy during training, with ResNet-50 showing superior performance in testing. Furthermore, CNN techniques generally surpassed the performance of Freeman chain code (FCC) extraction with SVM. Otsu binarization was also identified as a more efficient method than static threshold binarization in FCC extraction [6].

3. Methodology

An Automatic Number Plate Recognition (ANPR) system requires a series of steps and processes. The processes used in this work are illustrated in the form of a block diagram, as shown in Figure 1.

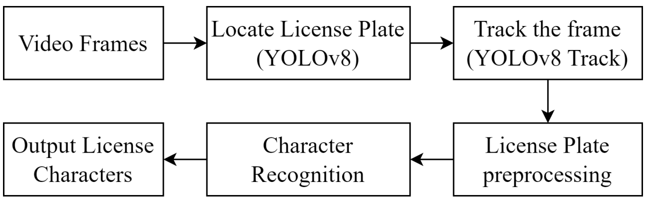

Figure 1: System block diagram

3.1 Data Collection

Data was collected for the training dataset. To collect data for creating a dataset for YOLOv8 to detect vehicles and number plates, we used a mobile phone to capture images of publicly parked vehicles. Additionally, some of the images were collected from publicly available videos on the internet. We positioned the camera at various angles and heights to capture different views of the vehicles. We ensured that each image contained at least one vehicle, and the number plate was visible. This task was particularly challenging due to the high volume of traffic on the roads of Nepal.

Furthermore, for character recognition, we cropped individual characters from the number plate images and assembled a character dataset comprising 17 classes. This dataset was specifically designed for the recognition of characters using Convolutional Neural Network (CNN) [7].

3.2 Data labeling

For image labeling, we employed LabelImg [8], a graphical image annotation tool. LabelImg is written in Python and utilizes Qt for its graphical interface. Annotations are saved as XML files in the PASCAL VOC format, which is the format used by ImageNet. Additionally, it supports YOLO and CreateML formats.

The classified vehicles can be of three categories: bus (heavy 4-wheeler), car (medium 4-wheeler), and bike (small 2-wheeler). Including number plate information, the dataset comprised a total of three classes. Finally, the data was prepared for training with YOLOv8.

3.3 Object Detection

For the detection task, we used YOLOv8. You Only Look Once (YOLO) is one of the most popular modules for real-time object detection, tracking, segmentation, and pose-estimation. YOLOv8 model was trained to detect vehicles and number plates using the custom dataset of 1778 images after proper labeling. The dataset was split in the ratio of 80:20 for train and validation data, respectively. Instead of initializing the weights randomly, pretrained weights on the COCO2017 dataset [9] were used. After training the model for 60 epochs, the process was terminated.

3.4 Object tracking

Object tracking plays a crucial role in video analytics, enabling the identification and classification of objects while maintaining their unique identities as the video progresses. With the advent of deep learning and computer vision technologies, You Only Look Once version 8 (YOLOv8) has emerged as a powerful solution for real-time object tracking.

YOLOv8 utilizes the Bot-SORT tracking method, a robust state-of-the-art tracker that combines motion and appearance information advantages. It also incorporates camera-motion compensation and a more accurate Kalman filter state vector. The BoT-SORT algorithm is used to track the targets detected by the detector to complete the whole detection and tracking process [10].

3.5 Processing Number Plate ROI

After the number plate had been detected and uniquely identified from the YOLOv8 tracking, the number plate Region of Interest (ROI) was extracted, and then further processing was carried out for character recognition. Number Plate Recognition is an image-processing technology used to identify vehicles by their number plate [11].

The bounding boxes of the number plate from the model contained the number plate ROI along with some other parts of the vehicle and were not perfectly horizontal, making them unsuitable for character segmentation. Therefore, the following processes were put into place to transform the image into one that could be processed further and to improve the effectiveness of the segmentation algorithm. Figure 2 depicts the original number plate image used in the process.

Figure 2: Number Plate Image

3.5.1 HSV Color Space Conversion

Color models like RGB, which were defined earlier, are very suitable for specifying color coordinates for display or printing. However, these types of color models are not conducive to representing the typical human perception of color. Instead, human perception of color is best described in terms of Hue, Saturation, and Value (HSV) [12]. The image was converted into the HSV color space. In the HSV color space, colors are described based on their Hue, Saturation, and Value. When accurate color representation is essential, the HSV color model is often preferred over the RGB model. Figure 3 illustrates the HSV color space.

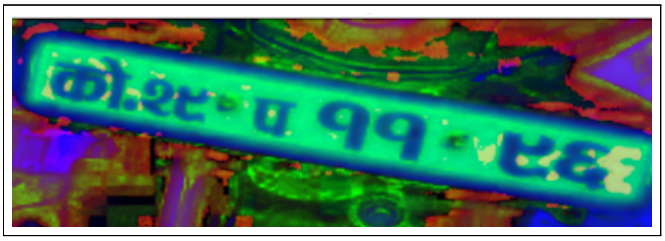

Figure 3: HSV color space

3.5.2 Color masking

Since Nepali number plates were of different color backgrounds (red background with white characters for private vehicles, black background with white characters for public vehicles, and many more), a red and black color mask was applied. This masked the number plate image with either a red or black mask depending on the type of number plate. Hue, saturation, and value (HSV) color space is applied for its good capability of representing the colors of human perception and simplicity of computation [13]. In the HSV model, the Hue value ranging from around 0°-10° and 350°-360° could be approximated as the red color, while for the black color, the Hue could be from 0°-360°, and a value in HSV between 40-80 was chosen based on the desired color intensity for masking. Using these ranges, the red color regions and black color regions were masked from the images or frames. Subsequently, the number of pixels in the red and black masks was calculated, and the dominant mask was selected based on the image with the higher number of pixels.

Figure 4 illustrates the image after applying the red and black mask. The Dominant Mask Image can be seen in Figure 5.

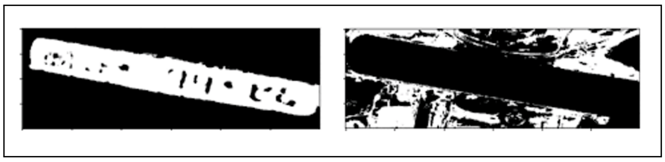

Figure 4: After applying red and black mask

Figure 5: Dominant Mask Image

3.5.3 Perspective transform

The color-masked image was sent for external contour detection, and the contours were sorted by area, with the largest contour being approximated by a minimum rectangle. This rectangle serves as the location of the characters on the number plate. However, due to variations in camera position and angle, the plate could be oriented in any direction, which would hinder segmentation if not corrected. To address this, the four coordinate points of the minimum rectangle (top-left, top-right, bottom-right, and bottom-left) were arranged in order.

A blank image was then created with a width equal to the maximum distance between the bottom-right and bottom-left x-coordinates or the top-right and top-left x-coordinates, and a height equal to the maximum distance between the top-right and bottom-right y-coordinates or top-left and bottom-left y-coordinates. This provided the size of the new image, and a set of destination points was obtained in the same order (top-left, top-right, bottom-right, and bottom-left).

To achieve a "bird's eye view" or top-down perspective of the number plate, a perspective transform matrix was calculated using OpenCV's getPerspectiveTransform() method [14]. This matrix was then used to compute the perspective transform, transforming the image to the desired top-down orientation. The perspective image resulting from the above process is depicted in Figure 6.

Figure 6: Final number plate image after perspective transform

3.6 Character Segmentation

This is the most important part in the Number Plate Recognition System. For the correct recognition of number characters accurate segmentation is very important [15]. There are several steps involved in in the character segmentation process.

3.6.1 RGB to Grayscale Conversion

In [16], True color is the specification of the color of a pixel on a display screen using a 24-bit value, which allows the possibility of up to 16,777,216 possible colors. Many displays today support only an 8-bit color value, allowing up to 256 possible colors. When converting from RGB to grayscale, it is said that specific weights to channels R, G, and B ought to be applied. These weights are: 0.2989, 0.5870, and 0.1140. It is said that the reason for this is different human perception/sensibility towards these three colors. The perspective transformed figure depicted in Figure 7 has been converted from RGB to grayscale.

Figure 7: Grayscale

3.6.2 Gaussian Blurring

Gaussian blur is a technique used in image processing to reduce noise and detail while preserving important features. It involves convolving the image with a Gaussian kernel, which applies a weighted average to each pixel's neighborhood. This process smooths out sharp transitions and reduces high-frequency noise, resulting in a softer appearance.

3.6.3 Otsu Binary Thresholding

Otsu's method looks at every possible value for the threshold between background and foreground, calculates the variance within each of the two clusters, and selects the value for which the weighted sum of these variances is the least [17]. The image after applying thresholding is shown in Figure 8

Figure 8: Threshold Image

3.6.4 Connected Component Analysis

The image still had unwanted small blobs and lines that were not characters. Connected Component Analysis [18] was performed to filter out these unwanted connected pixels. Filtration steps included the connected pixel's area if it was too large or too small to be a character, the aspect ratio of the connected component if it did not resemble that of characters, analyzing the number of white pixels, and the position of each connected component. After applying these steps, an image containing only characters was obtained. Subsequently, the application of these measures resulted in the extraction of an image exclusively comprising characters, as depicted in Figure 9.

Figure 9: Clean Image

3.6.5 Character locating with contour detection

Object contour plays an important role in fields such as semantic segmentation and image classification [19].After finding the contours of each character, a bounding box was obtained, which was then plotted to isolate each character. The bounding boxes were sorted from top to bottom and left to right on the number plate to ensure the correct order of characters. This ordering was essential for the character recognition process to accurately place the recognized characters in sequence, as shown in Figure 10.

Figure 10: Characters Segmented

3.7 Character Recognition

3.7.1 Character preprocessing

Each segmented character was cropped and then further processed before being fed into the classification model. First, the cropped character was resized to 64x64 pixels and then

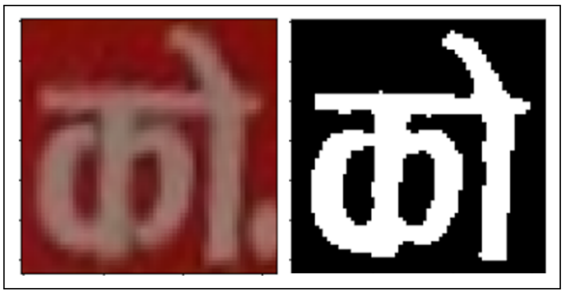

Figure 11: Character Preprocessed

converted to binary using a thresholding technique. Next, unnecessary blobs were removed, and finally, a morphological erosion operation was performed. Now, this image as shown in Figure 11 was ready to be fed into the classification model.

3.7.2 Neural Network (CNN)

A neural network [20] is a type of machine learning model that mimics the human brain. It creates an artificial neural network that, through an algorithm, enables the computer to learn by incorporating new data. An image represents 2-dimensional data, containing spatial information of pixel intensity. Before inputting the data into the ANN, we must extract features from the input image.

In this system, each segmented character was processed and then fed into the CNN model for character classification. After successfully classifying all segmented characters, we obtained a list of number plate characters, completing the character recognition process successfully.

The Neural Network architecture comprises eight layers in total. The initial five layers are convolutional layers, each utilizing a 3x3 filter size. A detailed explanation of these layers is provided in Table 1.

Table 1: Model Architecture Summary

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	64×64×3	-	-	-
1	Convolution	32	62×62×32	3×3	1	relu
	Max Pooling	32	31×31×32	2×2	2	relu
2	Convolution	64	31×31×256	3×3	1	relu
	Max Pooling	64	15×15×256	2×2	2	relu
3	Convolution	128	15×15×384	3×3	1	relu
4	Convolution	128	15×15×384	3×3	1	relu
5	Convolution	64	15×15×256	3×3	1	relu
	Max Pooling	64	7×7×256	2×2	2	relu
6	FC	-	12544	-	-	relu
7	FC	-	4096	-	-	relu
8	FC	-	4096	-	-	relu
Output	FC	-	17	-	-	Softmax

Additionally, Figure 12 illustrates the fully connected layer, indicating that an input image of size 64x64 results in an output layers with 17 classes, representing the classes to be predicted.

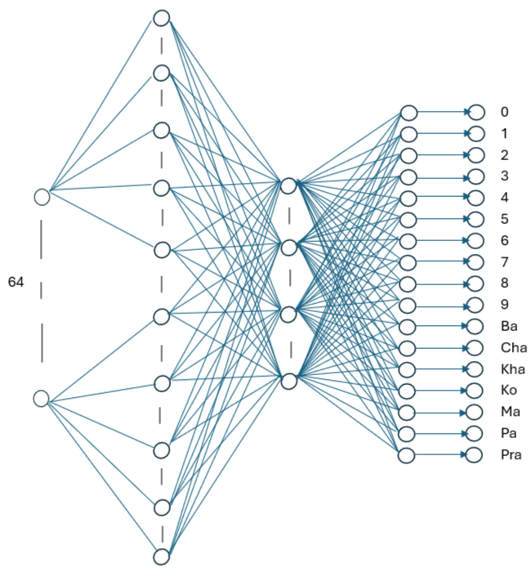

Figure 12: Convolution Neural Network

4. Model Evaluation

In evaluating the performance of the ANPR system, we employed a comprehensive set of metrics tailored to assess both the object detection capabilities using YOLO and the character classification accuracy achieved by the CNN model.

4.1 YOLOv8 detection metrics

The YOLO-based detection system demonstrated a precision of 91% percent, recall of 87% percent and an F-1 Score of 88% percent on the task of localizing and identifying number plates within images. Also, the model achieved average precision of 91% percent and a mean average precision (mAP) of 68% percent with a specific focus on the "number plate" class. The precision, recall and mean average precision for the classes Car, Bike and Bus is depicted in Table 2.

Table 2: Yolo Validation Result

Class	Precision	Recall	mAP0.5
Car	0.91	0.87	0.76
Bike	0.92	0.86	0.66
Bus	0.86	0.87	0.68
Average	**0.89**	**0.86**	**0.7**

4.2 CNN Character Classification Metrics:

The CNN model employed character classification achieved an accuracy of 87% percent, precision of 90% percent , recall of 85% percent and an F1-score of 87% percent.

Table 3: Character Recognition Result

Metrics	Score(%)
Average System Performance	88
System Error	12
Precision	90.97
Recall	85.16
F-score	87.75

Table 3 shows the various metrics of the CNN model. It can be seen that model performs considerably well in classifying 17 distinct classes depicted in Table 5.

5. Results and Discussion

The aim of this research is to Detect and Recognize Nepali number plates. First of all the system is trained against labeled vehicle dataset along with optical character dataset. And then, system is evaluated against new vehicle image samples and whole number plate to measure the accuracy of the proposed Automatic Number Plate Detection and Recognition System .

A. Datasets

1) Vehicle with Number Plate Dataset

There were about 1776 images of the Vehicles captured from different vehicles at different orientation and lighting conditions in which number plates were visible. Some of them were taken when the vehicles were moving and some of them when the vehicles were parked. Also CCTV footages were cropped for collecting images. Detail of the number of the samples for each class is given in the Table 4.

Table 4: Vehicle Dataset

Class	No. of training images	No. of validation images
Bike	397	111
Bus	509	150
Car	576	135
Total	**1482**	**396**

2) Character Dataset

Character dataset contains total 3572 samples for 17 classes. Dataset was created by manually cropping number plate characters. Detail of the number of the samples for each class is given in the Table 5.

Table 5: Character Dataset

Character	No. of Samples	Character	No. of Samples
0	201	9	196
1	218	ba	166
2	217	cha	244
3	222	kha	224
4	240	ko	221
5	241	me	158
6	203	pa	233
7	199	pra	203
8	186	**Total**	**3572**

B. YOLO Result

Proposed system was tested for 80 randomly chosen samples for vehicle and number plate detection and localization in which the images of number plate were clearly visible.

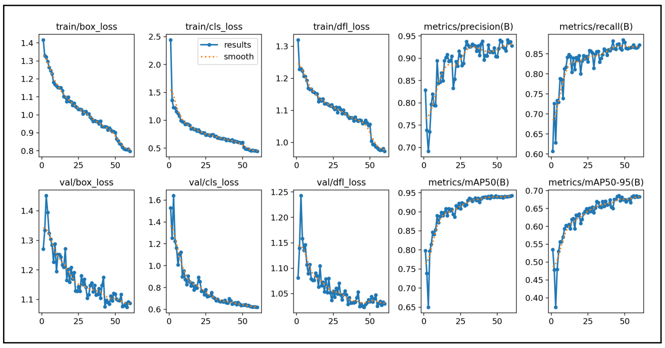

Figure 13: Yolo Validation Curve

C. Real time implementation

When we implemented this model for video processing, where the video was processed frame by frame, the YOLOv8 object tracking maintained accurate tracking at an average frame rate of approximately 15 frames per second (fps) on a low-end graphics card. However, with an Nvidia GeForce GTX1650-TI, the inference speed significantly improved, achieving 25-30 fps alongside the model for character recognition . This difference underscores the impact of hardware acceleration on real-time object detection and tracking tasks, highlighting the potential for even greater speed and efficiency with more advanced GPU setups.

D. Segmentation Result

In clear number plate image with red and black backgrounds (private and public) , the segmentation method demonstrated robust performance, accurately isolating individual characters. However, challenges arose when processing number plates with three rows structure and particularly in instances where plate were dirty or partially obscured.

In addition to the challenges mentioned, the segmentation method encountered difficulties in handling low-resolution images, especially with older number plates. The reduced clarity and distinctiveness of characters in such images posed significant obstacles to accurate segmentation and subsequent classification. Uneven characters, stemming from variations in font styles or degradation over time, further compounded the segmentation challenge, leading to suboptimal results in character isolation and recognition accuracy.

E. Character Recognition Result

Character recognition was performed on each uniquely tracked number plate from the video frames and the segmented characters were then fed into the CNN model for classification. Table 6 shows the final output.

Table 6: Recognition Test Results

License Plate Images	Total Alpha-numeric	Detected Alpha-numeric	Undetected Alpha-numeric	Results
	7	7	0	ko2cha2697
	12	12	0	pra301006kha0912
	6	6	0	me1cha871
	7	7	0	ko1cha6098
	12	9	3	pra###002cha5506
	8	9	0	ko20pa#2078
	8	8	0	ko14pa6283
	8	8	0	ba15cha2031

6. Conclusion

In conclusion, this paper demonstrates the effective utilization of the YOLOv8 model for object detection and tracking, alongside successful implementation of number plate segmentation and character recognition. These advancements highlight the practical applications of deep learning in solving real-world challenges. This research success paves the way for future enhancements in automatic traffic control, electronic toll collection, vehicle tracking, border security, and beyond.

7. Future Enhancements and Current Limitations

Object detection serves as the foundational step in the research, laying the groundwork for further advancements. Enhancing the model's performance requires a meticulously balanced dataset. Introducing new data and incorporating sample variations reflective of real-world deployment environments is imperative to bolstering the model's accuracy and robustness.

For number plate recognition, the current focus is on private and public vehicles in Bagmati, Koshi, and Mechi Zones. Expanding to other vehicle types and provinces is crucial due to the variability of Nepali number plates, particularly provincial plates and character segmentation. Currently, the dataset primarily consists of images of publicly parked vehicles with old standard plates, limiting the model's training. Future enhancements will include more diverse data sources, such as embossed plates, to better represent real-world scenarios.

While the research achieves object tracking at 15 fps, real-time integration is challenging due to the high computational demands of processing video streams at 60 fps or higher. Future development will focus on validating real-time implementation through camera integration and video processing. Efforts to collect additional datasets from road CCTV footage and residential surveillance cameras aim to overcome these limitations and achieve real-time functionality in future iterations.

Acknowledgments

The work on Automatic Number Plate Detection and Recognition is supported by Department of Electronics and Computer Engineering, IOE Purwanchal Campus, Dharan in 2024. The authors are grateful and pay deepest gratitude to Mr. Pravin Sangroula whose sharp observations and suggestions enhanced the quality of the work.

The authors would also like to extend their thanks to the District Traffic Police Office, Dharan for their support in making the availability of CCTV footage.

References

[1] Andrea Gutai, Andras Anderla, Nina Kozma, Anđela Milićević, and Sara Havzi. Number plate recognition using template matching technique. pages 473–483, 11 2022.

[2] Moahaimen Talib, Ahmed Al-Noori, and Jameelah Suad. Yolov8-cab: Improved yolov8 for real-time object detection. *Karbala International Journal of Modern Science*, 10, 01 2024.

[3] Ashok Pant, Prashnna Gyawali, and Shailesh Acharya. Automatic nepali number plate recognition with support vector machines. 12 2015.

[4] Pankaj Raj Dawadi, Bal Krishna Bal, and Manish Pokharel. Devanagari license plate detection, classification and recognition. In *Proceedings of the First International Conference on Advances in Computer Vision and Artificial Intelligence Technologies (ACVAIT 2022)*, pages 304–318. Atlantis Press, 2023.

[5] Gobinda Pandey, Nirajan Lamichhane, Utsav Subedi, and Karun K C. Cnn based system for automatic number plate recognition. *Journal of Soft Computing Paradigm*, 5:347–364, 01 2024.

[6] Firman Adhari, Taufik Abidin, and Ridha Ferdhiana. License plate character recognition using convolutional neural network. *Journal of Information Systems Engineering and Business Intelligence*, 8:51–60, 04 2022.

[7] T. Nukano, M. Fukumi, and M. Khalid. Vehicle license plate character recognition by neural networks. In *Proceedings of 2004 International Symposium on Intelligent Signal Processing and Communication Systems, 2004. ISPACS 2004.*, pages 771–775, 2004.

[8] Tzutalin. Labelimg. Free Software: MIT License, 2015.

[9] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[10] Shangqu Yan, Yaowen Fu, Wenpeng Zhang, Wei Yang, Ruofeng Yu, and Fatong Zhang. Multi-target instance segmentation and tracking using yolov8 and bot-sort for video sar. In *2023 5th International Conference on Electronic Engineering and Informatics (EEI)*, pages 506–510, 2023.

[11] Manjit Sandhu, Sukhdeep Kaur, and Jaipreet Kaur. An application of image processing -license plate recognition. 09 2022.

[12] Kamlesh Lakhwani, P Murarka, and Mr Narendra. Color space transformation for visual enhancement of noisy color image. *IET Image Processing*, 11 2015.

[13] Zhong Liu, Weihai Chen, Yuhua Zou, and Cun Hu. Regions of interest extraction based on hsv color space. In *IEEE 10th International Conference on Industrial Informatics*, pages 481–485, 2012.

[14] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[15] Mei-Sen Pan, Jun-Biao Yan, and Zheng-Hong Xiao. Vehicle license plate character segmentation. *International Journal of Automation and Computing*, 5:425–432, 10 2008.

[16] G. Jyothi, CH. Sushma, and D.S.S. Veeresh. Luminance based conversion of gray scale image to rgb image. *International Journal of Computer Science and Information Technology Research*, 1(1):Page Range, October 2014. Published from October 2013 to March 2014.

[17] Jamileh Yousefi. Image binarization using otsu thresholding algorithm. 05 2015.

[18] Ramalingeswara Rao K V, Bhaskara N., Ramesh Babu, Deeksha Goplani, and Anusha Pai. Connected component analysis for character feature extraction. 09 2013.

[19] Xin-Yi Gong, Hu Su, De Xu, Zhengtao Zhang, Fei Shen, and Hua-Bin Yang. An overview of contour detection approaches. *International Journal of Automation and Computing*, 15:1–17, 06 2018.

[20] Enzo Grossi and Massimo Buscema. Introduction to artificial neural networks. *European journal of gastroenterology & hepatology*, 19:1046–54, 01 2008.