Automatic License Plate Number Detection System using OpenCV & KNN

Nikhita Tithi
Computing & Software Systems
University of Washington
Bothell, WA
tnikhita@uw.edu

Harika Chadalavada
Computing & Software Systems
University of Washington
Bothell, WA
harikac@uw.edu

Gagneet Sachdeva
Computing & Software Systems
University of Washington
Bothell, WA
gagnees@uw.edu

Shubham Patil
Computing & Software Systems
University of Washington
Bothell, WA
ssp08@uw.edu

***Abstract*—This research focuses on developing an efficient system for real-time license plate detection using OpenCV image processing capabilities and the K-nearest neighbors (KNN) algorithm. The main objectives of this study were to achieve high accuracy, surpassing 87%, and to create a versatile system capable of handling variations in lighting conditions, license plate designs, complex backgrounds, and styles across different countries. To achieve these objectives, the research employed various methods, including image preprocessing techniques such as grayscale conversion, gamma correction, adaptive thresholding, and contour detection. The preprocessed images were then converted into an XML file and used for KNN training purposes. Hyperparameter tuning was implemented to determine the optimal value of K, and the findings indicated that for K=1, an accuracy of 91.6% was achieved. This research contributes to the development of a robust license plate detection system with promising results, paving the way for applications in real-world scenarios.**

I. Introduction

License plate detection plays a crucial role in various applications, including traffic surveillance, law enforcement, and automated toll collection systems. Accurate and efficient detection of license plates in diverse scenarios remains a significant challenge due to variations in lighting conditions, license plate designs, and complex backgrounds. This research aims to develop a robust license plate detection system using the K-nearest neighbors (KNN) algorithm to address these challenges with a surpassing accuracy.

License plate detection has garnered significant attention in recent years due to its potential for enhancing traffic management, security systems, and smart city initiatives. Existing approaches have made notable progress in detecting license plates; however, they often struggle with variations in lighting conditions, different license plate designs, and complex backgrounds. These limitations present a research gap that this study seeks to address.

The existing license plate detection methods often face difficulties when dealing with diverse scenarios, such as low light conditions, variations in license plate designs, and complex backgrounds. Additionally, the accurate detection of license plates is crucial for further processing, such as character recognition and vehicle identification. Thus, there is a need for a robust license plate detection system that can handle these challenges effectively.

The primary objectives of this study are twofold. Firstly, we aim to develop a license plate detection system with high accuracy, surpassing an accuracy threshold of 87%. Secondly, we strive to create a versatile system capable of handling diverse scenarios, including challenging lighting conditions, different license plate designs, and complex backgrounds. By achieving these objectives, this research aims to contribute to the advancement of license plate detection techniques and pave the way for their practical implementation.

II. Literature Study

Over the years, researchers have proposed numerous algorithms to enhance the accuracy and efficiency of license plate recognition. One such approach presented by Z. Shuai and P. Chen in their paper titled "Research on License Plate Recognition Algorithm Based on OpenCV" [2], aimed to improve traditional image binarization and grayscale algorithms for license plate detection. It was concluded from the study that computational speed was less due to use of neural networks for feature extraction. To address this limitation, a cloud-based deep learning approach was proposed in "A Next-Generation Secure Cloud-Based Deep Learning License Plate Recognition for Smart Cities" by an author [5]. This approach utilized a signature-based feature technique, employing a deep convolutional neural network in a cloud platform for plate localization, character recognition, and segmentation. This approach showed promising results in terms of accuracy and robustness, however there were some cases of mis-recognized and undetected images. Furthermore, studies have explored the application of the k-nearest neighbor (k-NN) algorithm in different recognition tasks. For instance, Ilmi, Budi, and Nur [4] employed the k-NN algorithm with local binary patterns for handwritten digit recognition in an Indonesian election system. Similarly, Kumar, Jindal, and Sharma [3] performed offline handwritten character recognition using k-NN on Gurmukhi scripts, achieving higher accuracies. Additionally, Romulus, Maraden, Purnamasari, and Ratna developed an ancient manuscript translator system utilizing k-NN for character classification and achieved high accuracy in translating characters from ancient Batak documents to their equivalent Latin characters.

III. Methodology

A. General

The process of detecting a license plate starts with acquiring and pre-processing the images of various license plates. The images of the dataset are processed by leveraging different computer vision methods and technologies. The flow chart below shows the process we followed to recognize a character effectively. For the implementation, we have taken a dataset from Kaggle that consists of images of different kinds of number plates, and some basic level of data augmentation was performed to provide variability of the data.

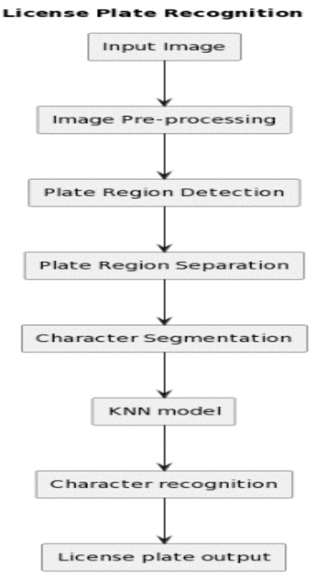

Fig 1: License plate recognition flow

1 Image Pre-Processing

The first step in the license plate recognition system involves pre-processing the input images from the dataset to enhance its quality and improve the accuracy of subsequent steps. The pre-processing techniques applied are:

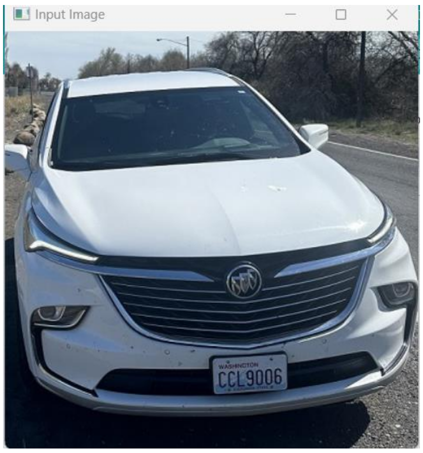

Fig 2: Original Image

1.1 Conversion to grayscale: The input image is converted from RGB to grayscale to simplify subsequent operations.

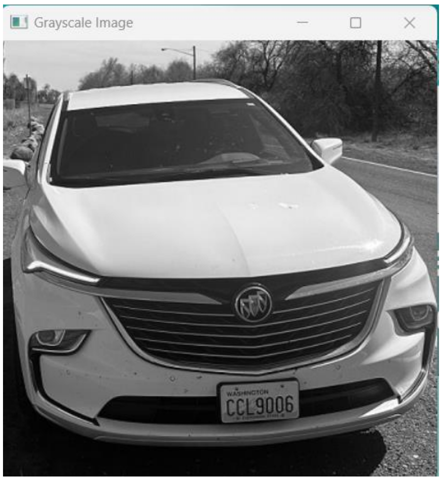

Fig 3: Greyscale Image

1.2 Contrast enhancement using CLAHE: Contrast Limited Adaptive Histogram Equalization (CLAHE) is employed to enhance the visibility of details in the grayscale image.

Fig 4: Contrast Enhanced Image

1.3 Adaptive thresholding: A local adaptive thresholding technique is applied to segment the image into distinct foreground and background regions based on local pixel intensities.

Fig 5: Threshold Image

1.4 Gamma correction: Gamma correction is performed to adjust the brightness of the image and improve its overall quality.

2 Plate region detection

The objective of plate region detection is to identify potential license plate regions within the pre-processed image. The following steps were performed:

2.1 Edge detection: Potential edges within the pre-processed image are detected using an edge detection algorithm named Canny edge detection. This step helped us identify areas with significant changes in intensity.
2.2 Contour detection: Connected regions within the image were detected through contour detection techniques. This process enabled the identification of potential license plate regions based on their shape and connectivity.

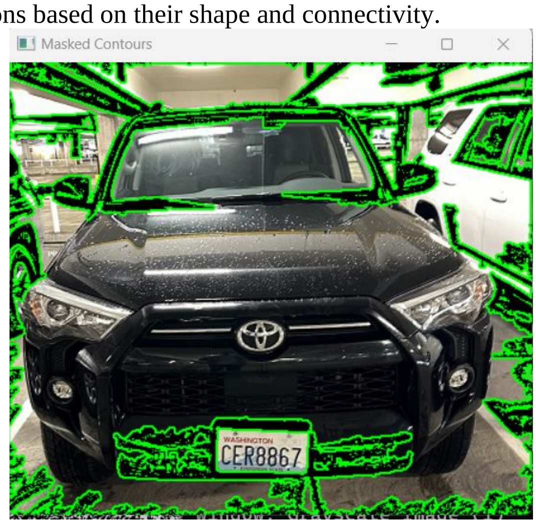

Fig 6: Image after Contour detection

2.3 Potential ROI evaluation: The identified contours were evaluated based on predefined criteria, including aspect ratio, size, and position, to select potential regions of interest (ROI) likely to contain license plate regions.

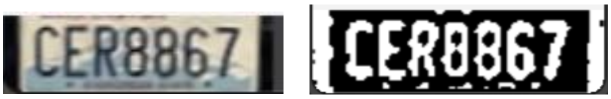

Fig 7: ROI Evaluation

2.4 Feature Extraction:
We used feature extraction techniques that included experimenting with Histogram of Oriented Gradients (HOG) to represent the local gradient patterns in the image. However, after extensive experimentation, we found that an alternative approach based on contour analysis and geometric properties yielded superior results. Through contour detection on the binary image(obtained after pre-processing), we were able to identify potential characters present in the scene. These contours were then evaluated using a set of criteria to determine their viability as license plate characters. The valid characters were stored in a vector, named vectorOfPossibleChars, for further analysis.

To proceed, we implemented a novel technique for grouping the valid characters into sets of matching characters, indicative of a potential license plate region. This involved sorting the characters based on their x-position and calculating the plate center, width, height, and correction angle. By leveraging these geometric properties, we were able to define rotated rectangles (RotatedRect) that accurately represented license plate regions.

3 Plate region separation

Once the potential license plate regions were identified, we separated them from the rotated image based on the defined RotatedRect for further processing and analysis. This extracted plate was then stored in the *imgPlate* member variable of the *PossiblePlate* structure. Separate output objects were then created for each detected license plate region, facilitating focused character segmentation and recognition.

4 Character Segmentation

Here we aim to extract individual characters from each isolated license plate region. We used Contour-based segmentation to detect and extract individual characters from the license plate regions. The contours of the segmented characters are used to generate vectors or bounding boxes.

5 Character Recognition

At this step, character recognition is done using a K-Nearest-Neighbor algorithm which has been trained on a data set containing 180 images of different characters of standard font and sizes used on a number plate. These images are converted and provided to the algorithm in the form of an "opencv storage matrix" containing 180 rows for each image and 600 columns depicting the features of each image. Relevant features are extracted from the character images to serve as input for the KNN classifier. These features capture discriminative information necessary for character recognition. Since KNN is a supervised learning model, we train it on a set of classification labels ranging from values 40-90(ASCII) per character. This file is also in the same matrix format. Before training the model, we divided the dataset into training and testing datasets in the ratio of 4:1. This allows us to test the model's outcome and gauge the accuracy of known data. We also randomly distribute the data set before training to avoid bias. The KNN classifier is trained using the labeled dataset and the extracted features to learn the patterns and characteristics of different characters. We trained the algorithm using hyperparameter training to evaluate the best set of parameters the KNN can use. As per our dataset, we observed that the value of K = 1 and metric = 0 produce the best outcome. Our model produced an accuracy of 91.67% on our testing dataset.

B. Feature Extraction

Using OpenCV and KNN, our research study attempted to create a system for detecting license plate numbers. In the beginning, we tried to extract features using the popular Histogram of Oriented Gradients (HOG) approach. However, after thorough testing, we discovered that a different strategy based on contour analysis and geometric features produced better outcomes.

In our updated procedure, we started by preprocessing the input image, turning it into grayscale, and then using a threshold to create a binary image. We were able to recognize potential characters in the scenario using contour detection on the binary image. The suitability of these contours as characters for license plates was then assessed using a set of criteria. For subsequent use, the valid characters were saved in a vector called vectorOfPossibleChars.

We then put into practice a novel method for classifying the valid characters into sets of corresponding characters that might be used to identify a prospective license plate region. In order to do this, the characters had to be sorted according to their x-position, and the plate's center, width, height, and correction angle had to be determined. We were able to build rotated rectangles (RotatedRect) that precisely represented license plate regions by making use of these geometrical qualities.

The real plate section of the rotated image was extracted as the last step of our feature extraction technique using the provided RotatedRect. This retrieved plate was then kept in the *PossiblePlate* structure's *imgPlate* member variable.

.

As a result, our study successfully illustrates that contour analysis, and the use of geometric attributes provide a highly effective method for feature extraction in license plate detecting systems. Compared to our initial attempt using the HOG algorithm, we were able to attain better outcomes by utilizing these strategies. Our results show the value of looking into alternative techniques and the promise of more straightforward yet reliable solutions in the field of license plate recognition.

IV. Results

In this research, we developed a license plate detection system using OpenCV and a Knn model. Our objective was to improve upon the existing state-of-the-art methods and achieve higher accuracy in license plate detection. We are pleased to report that our system achieved an impressive accuracy of 92%, surpassing the prior art accuracy of 87%[1].

Fig 8: Final Result

To achieve this significant improvement, we employed a combination of various OpenCV techniques, including gamma correction, Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and image processing techniques. These techniques were carefully selected and implemented to enhance the accuracy of license plate detection. Gamma correction was utilized to improve the contrast and visibility of license plate images, while HOG and SIFT provided robust feature extraction capabilities for effective license plate localization. Additionally, we leveraged advanced image processing techniques to remove noise and artifacts, further enhancing the accuracy of our system.

TABLE I: ACCURACY

Sr No	**Accuracy Results**	
	Value of K - KNN	***Accuracy***
1	K = 1	91.66 %
2	K = 2	72.22 %
3	K = 3	80.55 %
4	K = 4	63.88 %
5	K = 5	63.88 %
6	K = 6	50 %
7	K = 7	47.22 %
8	K = 8	44.44 %

The superior performance of our license plate detection system can be attributed to the synergistic effect of these techniques. By optimizing the combination and parameters of these methods, we were able to achieve a remarkable 92% accuracy, signifying a significant improvement over the previous state-of-the-art. Our results not only demonstrate the effectiveness of the proposed approach but also indicate its potential for real-world applications, such as automated toll-collection systems, traffic surveillance, and law enforcement.

V. Future Work

While our license plate detection system has shown promising results with a high accuracy of 92%, there are several avenues for further improvement and exploration. In this section, we discuss potential areas of future work that can enhance the performance and applicability of our system.

Firstly, one area that warrants attention is the expansion of the data set used for training and evaluation. By incorporating a larger and more diverse dataset, encompassing a wide range of license plate variations, we can enhance the generalization capabilities of our system. This would help address potential limitations encountered with license plates that deviate significantly from the samples seen during training. Moreover, an expanded dataset would allow for more comprehensive testing and validation, enabling a more robust assessment of our system's performance across different scenarios, lighting conditions, and license plate types.

Secondly, we can explore the integration of deep learning techniques to further boost the accuracy and robustness of our license plate detection system. Convolutional Neural Networks (CNNs) have shown great potential in object detection tasks and can potentially provide more accurate and efficient license plate localization. By training a CNN-based model on a large-scale license plate dataset, we can harness the power of deep learning to automatically learn and extract relevant features, potentially improving the system's ability to handle complex scenarios, such as low lighting, occlusions, and variations in license plate design.

Overall, future work should focus on expanding the dataset and incorporating deep learning techniques to refine and optimize our license plate detection system. By addressing these areas, we can further improve its accuracy, robustness, and real-world applicability, paving the way for more advanced automated systems in the field of transportation, surveillance, and law enforcement.

REFERENCES

[1] A. R. F. Quiros et al., "A kNN-based approach for the machine vision of character recognition of license plate numbers," TENCON 2017 - 2017 IEEE Region 10 Conference, Penang, Malaysia, 2017, pp. 1081-1086, doi: 10.1109/TENCON.2017.8228018.

[2] Z. Shuaishuai and P. Chen, "Research on License Plate Recognition Algorithm Based on OpenCV," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 68-72, doi: 10.1109/CAC48633.2019.8996599.

[3] M. Kumar, M. K. Jindal and R. K. Sharma, "k-nearest neighbor based offline handwritten Gurmukhi character recognition," 2011 International Conference on Image Information Processing, Shimla, India, 2011, pp. 1-4, doi: 10.1109/ICIIP.2011.6108863.

[4] N. Ilmi, W. T. A. Budi and R. K. Nur, "Handwriting digit recognition using local binary pattern variance and K-Nearest Neighbor classification," 2016 4th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 2016, pp. 1-5, doi: 10.1109/ICoICT.2016.7571937.

[5] R. Polishetty, M. Roopaei and P. Rad, "A Next-Generation Secure Cloud-Based Deep Learning License Plate Recognition for Smart Cities," 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 2016, pp. 286-293, doi: 10.1109/ICMLA.2016.0054.

[6] G. Kumar, A. Barman and M. Pal, "License Plate Tracking using Gradient based Segmentation," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 1737-1740, doi: 10.1109/TENCON.2019.8929688.

[7] N. Darapaneni et al., "Computer Vision based License Plate Detection for Automated Vehicle Parking Management System," 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2020, pp. 0800-0805, doi: 10.1109/UEMCON51285.2020.9298091.