<ICMSS_Parvathi>

[Materials Today: Proceedings]

Automated Vehicle Number Plate Detection

Savio S. Moloparambil[a]1, Aswathi M. Kumar[a]2, Jeyahari R. [a]3, Parvathi R. [a]4

[a] *School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India*

Abstract

Automatic Number Plate Recognition (ANPR) is a specialized image processing method that identifies the text on a given vehicle's number plate. The goal is to create a successful automatic approved vehicle identification system that makes use of the license plate. The system may be placed in many scenarios and locations, some of which may include security in prohibited areas like military and testing zones, or the vicinity of important government buildings like the Supreme Court, Parliament, etc. Using image segmentation in an image, the region containing the vehicle number plate from the image of a vehicle is extracted. Character recognition is achieved using an optical character recognition (OCR) approach. In order to determine miscellaneous details like the owner of any detected vehicle, the location of registration, the address and whereabouts, etc.

Keywords: Automatic license/number plate recognition (ANPR/ALPR); Optical Character recognition (OCR); Python; OpenCV; PaddleOCR; license plate;

1. Introduction

In a world where vehicles only become faster over time, accidents are more frequent and law enforcement on motorways become more difficult to handle. There is the need to find the number plate of a vehicle that may be breaking road laws such as parking in the wrong area, over speeding, skipping toll gates, fender benders or even harsh accidents. There also come certain scenarios where we need to locate vehicles of interest when the only info in hand is the number plate of the missing car. In the aforementioned cases, the functionality of detecting the text of number plates on cars via images or video becomes vital to such cases.

In order to detect number plates of vehicles on roads, highways, streets, public places, toll gates, etc. we look towards machine learning algorithms that are able to know where to look in each image or frame of a video and find ways to extract text from the images. This is where the utilization of OCR comes into play, as it is the method used specifically for character extraction from said images. The aim of this paper is to thus be able to apply such algorithms into code using Python, OpenCV and PaddleOCR in such a way that it can take images as input and provide the text of solely the number plates of vehicles that are in the images. We use Python as the main programming language here as it is more flexible and easier to use for this application, and it comes with the benefit of supporting well-performing packages like OpenCV, and PaddleOCR.

Number plate extraction, character segmentation, and character identification are the general procedures used in ANPR operations. In the subsequent character segmentation stage, just the number plate is identified from the complete input picture and is processed further. Each and every character is separated and segmented during the character segmentation process. In the character identification step, each character is identified based on a selection of distinguishing properties. Number plate extraction is a challenging process, primarily because: Number plates typically comprise a tiny percentage of the entire image; different number plate layouts; and effect of ambient conditions. The

*Corresponding author. Tel.: +91-90616-12080.
E-mail address: saviosajan.m2020@vitstudent.ac.in

*Corresponding author. Tel.: +91-91672-21031.
E-mail address: aswathimanoj.kumar2020@vitstudent.ac.in

*Corresponding author. Tel.: +91-63826-59146.
E-mail address: jeyahari.rajasekar2020@vitstudent.ac.in

*Corresponding author. Tel.: +91-94430-59207.
E-mail address: parvathi.r@vit.ac.in

accuracy of character segmentation and recognition operations is impacted by this phase. While the primary objective of this paper is to be able to detect characters and numbers from a license plate, there however arises a dilemma in choosing between the two most prominent OCR libraries for such a task, namely Tesseract and PaddleOCR. In this paper we shall be using PaddleOCR however as well as discuss on how its results are comparative to that of Tesseract.

2. Literature Review

Reference [1] shows the taking of image size, success rate, and processing time into consideration as criteria, and various ANPR techniques are described. ANPR algorithms are generally divided into Vehicle image capture, Number plate detection, Character segmentation and Character recognition.

Reference [2] shows the performance contrast of various simulated algorithms, some of which in assimilate computer vision, as well as an in-depth assessment of current approaches and developments in automatic number-plate recognition (ANPR) systems (CV). Utilizing recognition algorithms, ANPR equipment can detect and identify vehicles by their number plates.

Reference [3] shows that the an ANPR system can be developed using a deep learning library in python called IMAGEAI as well as NVIDIA Jetson Nano kit to help in the training phase. The system performed badly in brightly lit surroundings when morphological processing was initially used for license plate localization, according to an analysis of the algorithm's effectiveness. The deployment of edge detection methods comes next, increasing the localization's effectiveness. Finally, the considerably more effective IMAGEAI object detection framework is employed.

Reference [4] set out to increase the recognition rate of license plate characters using a variety of techniques, including segmenting characters and extracting the license plate. The obtained data demonstrates that the recommended process combination does indeed produce a very high recognition rate. The technique has been tested using static images of automobiles that have been sorted into several sets based on their difficulty.

Reference [5] shows a complete, effective, and layout-independent system for reading license plates that investigates real-time based models at every level. With the use of post-processing rules, the suggested method unifies the detection of license plates and the classification of layouts.

Reference [6] shows a technique for reading and identifying Bengali-written license plates on vehicles in Bangladesh where the number plate areas are retrieved from the vehicle photos in this system using the template matching technique. Each character is segmented and to recognize the segmented characters on the number plate, a convolutional neural network is utilized to extract characteristics from each character and classify the vehicle city, kind, and number.

Reference [7] shows the need to look into different character extraction and identification methods in order to increase the design efficiency of a number plate recognition system. Automatic number plate recognition (ANPR) is a form of widespread surveillance that reads vehicle license plates using optical character recognition on photographs.

Reference [8] shows a technique that works well for low-resolution photos in real-time applications. It is capable of achieving speedier recognition in challenging circumstances, particularly in low light and bad weather situations as well as for damaged or soiled photographs of license plates. There won't be a need to do image processing such as image denoising, picture enhancement, image segmentation, and so on under these unique situations.

Reference [9] shows a convolutional neural network with three phases, the first of which performs license plate detection using models from the Haar cascade and canonical correlation analysis. After utilizing the bounding rectangle approach to segment, the image, the model is finally used to identify the characters in the image.

Reference [10] shows a license plate recognition system that was created and intended to run on cutting-edge smartphones and is capable of working in the dark without any extra lighting. This research also demonstrates that no prior attempts for license plate identification only on edge devices have been made with an accuracy comparable to the solutions created to run on server-grade hardware.

Reference [11] shows evidence of challenges faced by automatic license plate identification on Egyptian highways, where issues exist that go beyond the capabilities of photography, picture quality, illumination levels, and alignment. This article shows that the situation's scope is more than the degree of competing jurisdictions without any coordination or planning for automated control.

Reference [12] has examined into some principles for ALPR systems and uses a high-performing ALPR network in a cascaded, resample-based method to increase the speed of inference. This technique gets the greatest performance on the CCPD and AOLP datasets because to a new architectural design that includes an integration block, vertex estimation, horizontal encoding, and weight-sharing classifier. This approach also works well with PKUData and CLPD's unseen photos.

Reference [13] shows that the three processes of character recognition—license plate detection, character segmentation, and character recognition—are often separated into the process of reading license plates. Different processing methods are offered in the license plate recognition system to cope with complicated scenarios such uneven lighting, an unfixed shooting angle, various weather conditions, and motion blur in actual settings. These factors make it difficult for the system to identify and recognize the picture.

Reference [14] shows how certain requirements and design strategies must be carefully chosen in order to accommodate various operational and hardware limitations while designing and developing an automated license plate recognition system. In this review work, the methods and strategies currently employed in ALPR solutions have been examined and analyzed. Solutions based on single-stage deep learning have demonstrated excellent performance on a variety of datasets.

Reference [15] shows the taken steps of picture capture, license plate extraction, segmentation, and recognition that are all part of the license plate recognition process. In addition to the usage of Arabic, Libyan license plates contain a few distinctive characteristics that have to be taken into account throughout the identification stages. Pictures of license plates were taken, and the data was then divided into smaller images, each of which included a letter, number, or word from the WHT spectrum. The coefficients and the stored database coefficients were compared. The system saved the character if the chosen coefficients matched those of the reference character. This was performed for each of the subsequent characters in the picture until the vehicle number plate was finished.

3. Proposed Methodology

Pre-processing, detection, identification, and searching are the four main stages of the suggested technique, as indicated in Fig. 1.

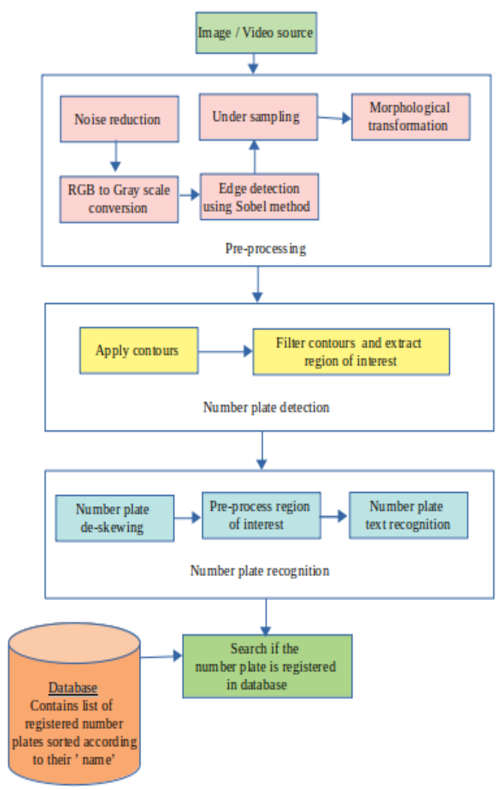

Fig. 1. Methodology of the recognition system (referenced from [1])

4. Brief Description Of Steps

The following steps are included in the automatic license/number plate recognition (ANPR/ALPR) procedure:
Step 1: In a given image, locate the vehicle license plate
Step 2: Once the license plate is found and selected, extract the characters that may be in the plate
Step 3: Use optical character recognition (OCR) software to identify the characters that were extracted.

4.1. Implementing OpenCV and Python

We use a Python class namely PyImageSearchANPR to offer a reusable method for character OCR and license plate localization processes. A license plate typically has rectangular proportions, with the aspect ratio range being (minAR to maxAR).

4.2. Debugging computer vision pipeline

When in debug mode, we provide a helper method to display findings at various points in the imaging pipeline.

4.3. Locating potential license plate candidates

We can locate the license plate candidate contours in an image using our first ANPR approach. We need a grayscale version of the original image (Fig. 2) that has a potential license plate. To assist us streamline our ANPR pipeline, we generalize. From this point forward, let's suppose that the majority of license plates have a white background color (mostly reflective and blank) and a black foreground (letters). We carry out a Blackhat morphological operation to make light backgrounds and dark characters (letters, numbers, and symbols) visible.

Fig. 2. Image before performing image processing

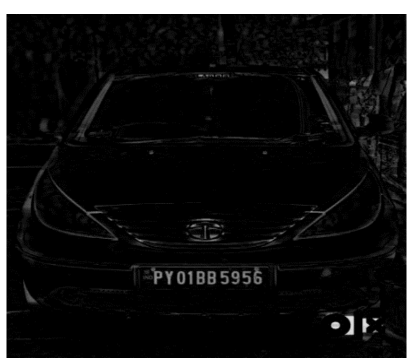

Fig. 3. After performing Blackhat morphological operation

The Blackhat morphological operator of OpenCV contrasts the license plate numerals with the rest of the image of the car's back end. As white letters on a black background, the license plate numbers stand out, and the majority of the background noise has been washed out.

The characters on the license plate are mostly clear, as we can see from Fig. 3. We identify areas in the image that are light and might have characters from a license plate. We use a closing operation to fill in minor gaps and aid in the identification of larger structures in the image using a small square kernel. Then, using Otsu's technique, we conduct a binary black-white threshold on our image to show the light areas in the image that might contain license plate letters and numbers. The area where the license plate is located appears to be practically one single huge white surface as a result of the closure operation and Otsu's inverse binary thresholding.

Fig. 4. Closure and thresholding

Fig. 4 demonstrates how the area around the license plate stands out. The edges of the image will be picked up by the Scharr gradient, which will then draw attention to the license plate's character limits. We calculate the Blackhat image's x-directional Scharr gradient magnitude representation. The intensities that are produced are then scaled down to the range [0, 255]. Fig. 5 shows how the margins of the license plate characters are highlighted:

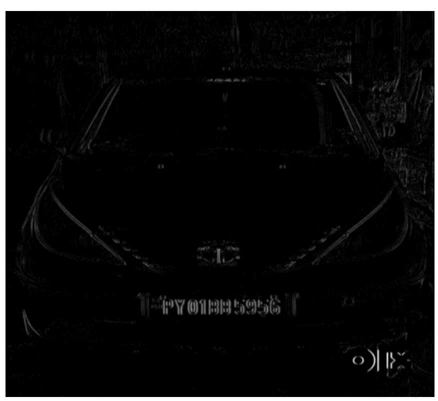

Fig. 5. Accentuating edges using Scharr's algorithm

The characters on the license plate, as shown above, stand out significantly against the background. Now, we can group the areas that may have borders for the characters on license plates smoothly. For noise reduction, we apply a Gaussian blur to the gradient magnitude image. Using Otsu's technique, we again perform a closure operation and a binary threshold. Fig. 6 depicts the characters from a license plate in a continuous white area:

Fig. 6. White area on number plate

These outcomes appear cluttered at first glance. There are numerous more substantial white patches in addition to the somewhat defined license plate zone. To denoise the post-threshold image, we apply a series of erosions and dilations:

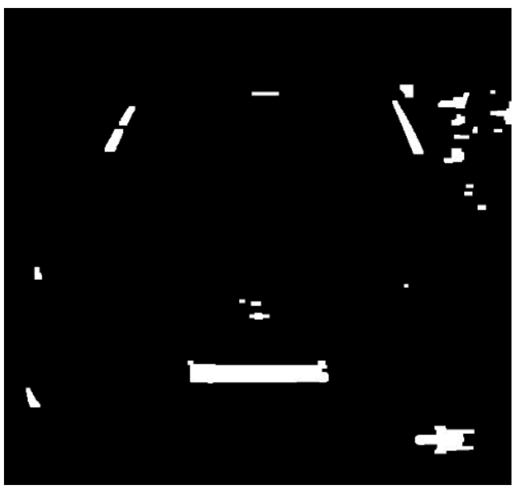

Fig. 7. Cleaned up image after dilation and erosion.

Fig. 7 shows how the erosion and dilation processes significantly reduced the noise in the earlier result from Fig. 6. We previously developed a technique to draw attention to lighter areas of the image (assuming the background is light/white and the letters are dark/black). To showcase the potential license plate choices, we perform a bitwise-AND on the post-threshold result with the luminous parts of the image. To fill in gaps and clean up the image, we then apply a few dilations and an erosion. Our "Final" debugging image is shown in Fig. 8:

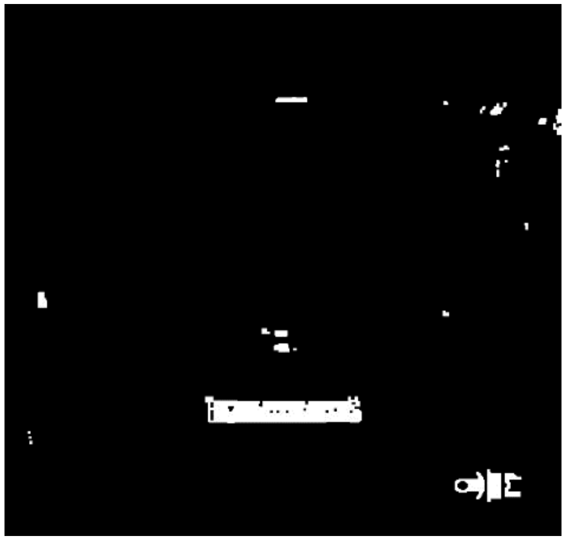

Fig. 8. Visible contour of number plate

The contours are found next. Accepting a grayscale image, we employ conventional image processing methods with a focus on morphological processes to identify a number of potential license plate-containing contours. So overall, we have taken a grayscale image as input and applied conventional image processing methods, focusing on morphological operations to identify a number of potential contours that might house a license plate.

4.4. Pruning license plate candidates

From our list of potential contours, we select the one that contains a license plate most frequently. We identify the license plate's region of interest and extract it from the license plate's contour. By using the aspect ratio of the bounding box of the contour, we can make certain that our contour has the proper rectangular shape for a license plate (Fig. 9). PaddleOCR is used to decode the characters using the output of our ANPR localization process, which is built using Python and OpenCV.

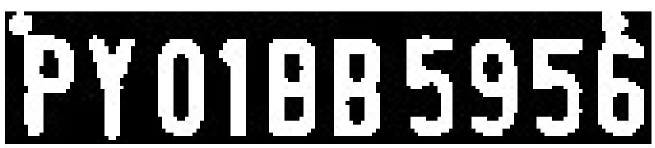

Fig. 9. The outcomes of our ANPR localization workflow, which is based on Python and OpenCV. This sample is excellent for sending to the paddle OCR.

4.5. Optical Character Recognition using PaddleOCR

PaddleOCR has been developed using pre-trained models of many varieties, thus crediting its reliability in text detection. Its main functionality involves text recognition, text detection, and text direction classifier. PaddleOCR is packaged with numerous models, including the flagship PP-OCR as well as being equipped with some highly regarded algorithms some of which include SRN, NRTR. So, we create a class where we can implement PaddleOCR on any desired image.

4.6. Creating a driver script with OpenCV and PaddleOCR

- Load the input image.
- Find and locate the license plate in the image using image functions such as thresholding and contours

- OCR the license plate, specifically at the area where the license plate may be most likely found.
- Display the ANPR result in the output window.

But before that we need to initialize a utility process. String Cleanup-since OpenCV cannot recognize special characters, it registers them as a '?'. So, to prevent special characters/symbols in a number plate from bottlenecking the process, we use a function to parse out all alphanumeric characters as a safety mechanism.

5. Experiment Details

Systems for automatically reading license plates come in a variety of sizes and shapes. Basic image processing methods are utilized in ANPR under favorable conditions including lighting and easily readable license plates. To locate license plates in photos, more sophisticated ANPR systems use specialized object detectors like HOG + Linear SVM, Faster R-CNN, SSDs, and YOLO. Modern ANPR software utilizes Long Short-Term Memory networks (LSTMs) and Recurrent Neural Networks (RNNs) to improve the OCR of the text from the license plates themselves.

5.1. Dataset Preparation

Finding a dataset to train a unique ANPR model is just one of many interrelated issues that make ANPR very difficult. Large, reliable ANPR datasets that are employed to train cutting-edge models are highly guarded and infrequently (if ever) made available to the general public. These databases include private identifiable data on the location, driver, and vehicle. Curating ANPR datasets is laborious and necessitates a significant time commitment from staff members. Contracts for ANPR with state and local governments are frequently very competitive. Because of this, the dataset that a particular organization has selected is frequently more useful than the trained model. Hence, our dataset consists of images of selected formats collected from over various websites. The images are vehicles containing a license plate and taken from different angles of view and under various conditions. For our experiment, we have procured a set of images from Kaggle as well as the very images used by the authors in [1]. In specification, we have selected 2 sets of images for our testing. The first set comprises of 10 images and has been sourced from the GitHub repository in [1]. The second set is an arbitrary set of 100 images that have been collected and compiled from Kaggle, most of which are images of cars from different angles and perspectives to pose a challenging experiment for the OCR algorithms we shall be employing.

5.2. Configuring OCR development environment

Installing OpenCV and virtual Python environments is advised before continuing. Pip, virtualenv, and virtualenvwrapper are all used together. Imutils, the scikit-image library, and the OpenCV binaries must all be installed in a Python virtual environment. Imutils, scikit-image, and opencv-contrib-python libraries can all be installed via pip. PaddleOCR is also required for text and character recognition. The deployment of various machine learning and computer vision algorithms will require all of these setups.

6. Results and Discussion

We apply the Automatic License/Number Plate Recognition using OpenCV and Python. We execute the command for our test images.

Fig. 10. Our PaddleOCR, Python, OpenCV, and Automatic License/Number Plate Recognition algorithm works well.

All of these pictures, including the license/number plate examples on the front or back of the car, demonstrate that ANPR has been successfully applied to them (Fig. 10). The detected license plate is enclosed in a green box and the predicted text of the license plate is also displayed above it. There are a certain set of images where our model fails as it predicts inaccurately. The limitations of our model will be discussed later. We can also observe that PaddleOCR works better than Tesseract.

6.1. Accuracy

Our PaddleOCR model accuracy is low as there are many wrong predictions made by our model. Although the accuracy of our model is not high, it can still predict a large class of data where it works well slightly better when compared to Tesseract. Simple visualizations of this are demonstrated in Table 1 and in both Fig. 11 (Bar plot comparison) and Fig. 12 (Confusion matrix).

Table 1. Accuracy of Tesseract and PaddleOCR

Images Scanned	Tesseract (language = "eng")	PaddleOCR
	Accuracy	
10 (Provided in [1])	10%	30%
100 (Provided in Kaggle)	2%	8%

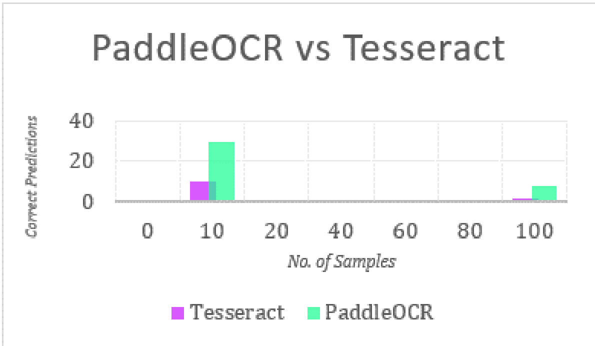

Fig. 11. Bar Plot of the performance comparison between PaddleOCR and Tesseract

Tesseract	***Correct/Present*** *(Actual)*	***Wrong/Missing*** *(Actual)*
Correct/Present *(Predicted)*	1	0
Wrong/Missing *(Predicted)*	9	0

(a)

PaddleOCR	***Correct/Present*** *(Actual)*	***Wrong/Missing*** *(Actual)*
Correct/Present *(Predicted)*	3	0
Wrong/Missing *(Predicted)*	7	0

(b)

Fig. 12. Confusion matrices for the performances of (a) Tesseract; (b) PaddleOCR on the first dataset (given in [1]) of 10 images.

7. Implementation

With the program code now able to extract the text of vehicle license plate numbers from images of said vehicles, the final step of our work involves the implementation of this program in a form that takes images in as input and gives the text of license plates as output. The form factor that suits this sort of implementation would be a web application with said functionalities.

The web application constitutes 2 pages, one for taking the image as input, and the other to return the text extracted from the image. Since the base code of the program is in python, the framework used in this implementation is Flask. Utilizing this framework as well as templates made in HTML, CSS and Bootstrap, the web pages are built and functioning properly.

The home page utilizes HTML form with file input. The file, once uploaded into the home page, is sent to the flask application via a POST request sent by the form that calls on the second page of the web application. Flask takes the file and saves it temporarily on the machine. It first converts the file from .jpeg/.jpg formats to .png or leaves it as is if it is already in the .png format. Once the conversion is done, it gets processed by the class imported from the file where said class implements the text extraction program discussed earlier. To avoid errors involving files not being uploaded when the form is submitted, the file input field is tagged as “required” in HTML. This is so that if the flask application runs without any file input, an error occurs due to missing files that are to be processed for the next page.

Once the text is extracted, the text is sent back into the flask app and then the same text is forwarded to the 2nd page as an argument along with the page template itself. The output page then showcases the image and the extracted number plate text using similar styles used in the home page itself. Once the purpose of the user is met, the home button placed in the output page can be used by the user to input another image as the button redirects to the home page where it awaits the input of an image once more (Fig. 13). During the process of redirecting to the home page, the flask app checks if the temporary image file is still in the system, and when it finds the image, it deletes it as it is no longer useful to the program itself.

Fig. 13. Web Page implementation of the OCR algorithm for number plate detection

8. Applications

Automatic Number Plate Recognition has an array of applications that prove to be very useful:

8.1. Law Enforcement

ANPR is used for law enforcement in many situations which is seen often. Such as the system that the Police use for checking if a vehicle is registered or to identify the vehicles that commit traffic violations using cameras near, which snap a picture of the vehicle that is either speeding or has crossed the red light. From the picture the number plate is extracted from using ANPR to issue the fine. Identifying vehicles also allows the authorities to track them real time.

8.2. Smart parking management system

This is a system we have actively experienced in malls, where each vehicle requires a paid ticket to park. Car parking management for a large number of vehicles requires a solution to detect individual vehicles and keep track of them, this would be a very difficult task to achieve manually, But, ANPR allows these parking garages to have automated parking management since every car can be accounted for using their license plate. It lessens the burden on the system while also making it easier for the garage users to avoid having to manage their tickets and the time spent, preventing them from acquiring penalties for losing tickets or for inaccurate ticket payments.

8.3. Toll Booths

Toll Booths is something that we see commonly across India, especially in Tamil Nadu. These toll booths are manually operated, from fee collection to the opening of the boom barriers. In some countries, this whole process is automated using various tools, including ANPR. For example, on highways which have a constant high amount of traffic, automation can help speed up the tollbooth procedure to help mitigate traffic. ANPR can identify the number plates and the user can pay via online methods or it could be an instant auto payment.

8.4. Traffic Management

In cities with high population densities, traffic can be a perpetual issue. Especially in Indian cities, which have very high population densities. ANPR allows us to measure and analyse area-related traffic data of an entire city. Analysing this data helps us to understand the traffic congestion aid in better traffic planning.

For example, the UK have collaborated with Siemens Traffic for regulation in public and control centres using certain monitoring systems. Projects such as Hampshire's ROMANSE use an interactive website with real time updates on traffic in the city.

9. Limitations

9.1. Recognition rate

The quality of videos and images taken by surveillance cameras has considerable increased over the past decade. But this alone does not guarantee perfect results in the case of Automatic Number Plate Recognition. It may be an issue with the used algorithm which may cause failed recognition. In our case, different OCR algorithms affected the accuracy of the results. In the case of Tesseract OCR, small changes such as clearing the foreground pixels produced better results due to the sensitivity of the algorithm.

9.2. External factors

External factors, of natural causes in specific, are hard to account for in case of automated systems. Capturing a clear image could be difficult in the case of heavy rain which may affect the camera lens itself; or in the case of heavy winds which may cause object to blow onto the camera, obstructing the view. There could also be other issues such as angle of sunlight, smoke, fog, snow etc. which cannot be fixed with a small change in the algorithm. Lack of a clear image can disrupt the process and damage to the hardware can disrupt the whole system.

9.3. Privacy

As discussed earlier, ANPR systems allow the police authorities to track vehicles for the purpose of enforcing law. But it was also discussed that ANPR systems are used in malls and parking garage systems. The issue of privacy has always been principal especially with Meta's privacy security coming into question. These systems store your license plate information and many other information is accessible just with access to your license plate number. It can also become a victim to data thefts.

References

[1] Hirethanad, J. R., Mahindrakar, N., Jayakumar, P., Shariff, S. M., & Premkumar, S. (2020). Automatic Number Plate Recognition for Indian vehicles. [Unpublished manuscript] Github. https://github.com/parvatijay2901/Automatic-Number-plate-detection-for-Indian-vehicles/blob/main/Project_Report.pdf

[2] Mufti, N., & Shah, S. A. A. (2021). Automatic number plate Recognition: A detailed survey of relevant algorithms. Sensors, 21(9), 3028.

[3] Gnanaprakash, V., Kanthimathi, N., & Saranya, N. (2021, March). Automatic number plate recognition using deep learning. In IOP Conference Series: Materials Science and Engineering (Vol. 1084, No. 1, p. 012027). IOP Publishing.

[4] REDDY, G. D., DIVYA, M., MOUNIKA, K., & MOUNIKA, K. LICENSE NUMBER PLATE RECOGNITION SYSTEM.

[5] Laroca, R., Zanlorensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R., & Menotti, D. (2021). An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. IET Intelligent Transport Systems, 15(4), 483-503.

[6] Alam, N. A., Ahsan, M., Based, M. A., & Haider, J. (2021). Intelligent system for vehicles number plate detection and recognition using convolutional neural networks. Technologies, 9(1), 9.
[7] Tejaswini, C. L., & Rangaswamy, C. AUTOMATIC NUMBER PLATE RECOGNITION. Journal homepage: www. ijrpr. com ISSN, 2582, 7421.
[8] Habeeb, D., & Talib, D. A. License plate Recognition using SVM.
[9] Pattanaik, A., & Balabantaray, R. C. (2022). Licence Plate Recognition System for Intelligence Transportation Using BR-CNN. In Advances in Data Computing, Communication and Security (pp. 659-668). Springer, Singapore.
[10] Shashirangana, J., Padmasiri, H., Meedeniya, D., Perera, C., Nayak, S. R., Nayak, J., ... & Kadry, S. (2022). License plate recognition using neural architecture search for edge devices. International Journal of Intelligent Systems, 37(12), 10211-10248.
[11] Tarek, R., & Yousri, A. (2022). Automated License Plate Recognition in Egyptian Environment: Lessons Learned.
[12] Wang, Y., Bian, Z. P., Zhou, Y., & Chau, L. P. (2021). Rethinking and designing a high-performing automatic license plate recognition approach. IEEE Transactions on Intelligent Transportation Systems.
[13] Weihong, W., & Jiaoyang, T. (2020). Research on license plate recognition algorithms based on deep learning in complex environment. IEEE Access, 8, 91661-91675.
[14] Shashirangana, J., Padmasiri, H., Meedeniya, D., & Perera, C. (2020). Automated license plate recognition: a survey on methods and techniques. IEEE Access, 9, 11203-11225.
[15] Elbokhare, A. F., Gamati, E., & Soliman, A. M. (2020). Libyan Vehicle Number Plate Recognition System using (WHT).