IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CLOUD-BASED ANTI-THEFT VEHICLE NUMBER PLATE DETECTION

SYEDA SHAHANA AMREEN, NANDINI V PATIL

PG STUDENT DEPARTMENT OF CSE, ASSISTANT PROFESSOR CSE DEPARTMENT

SHARANBASVA UNIVERSITY KALABURGI, INDIA

ABSTRACT: Investigation is being carried out by experts on intelligent transportation systems, which will significantly ameliorate human lives. A computer vision technique called Automatic License Plate Recognition (ALPR) is used to evoke automobile license plates from photographs. It is an embedded system with many uses and complications. Standard ALPR systems are prohibitive because they use proprietary technologies. Although, this confined methodology restricts the system's continued research and development. The expansion of open-source technology has hoisted the field of computing to new heights. In a multi-cultural setting, people from other communities engage to create solutions to man's perpetual obstacles. The method for detecting and identifying vehicle number plates that are proposed in this project will aid in the identification of legal and unauthorized vehicle number plates.

Keywords: License plate, character recognition, python, computer vision.

1. INTRODUCTION

In recent years, license plate recognition (LPR), also known as ANPR, has been shown to be one of the effective methods for vehicle monitoring. It can be used in many public locations to accomplish a variety of goals, including traffic safety enforcement, automatic toll text collecting, parking system, and automatic vehicle parking system. In general, ANPR algorithms consist of four steps:

(1) Capturing a vehicle's image

(2) Finding license plates

(3) Segmenting characters and

(4) Recognizing characters.

The first stage, which is to take a picture of the car, appears to be relatively simple, but it is very demanding work since it is quite challenging to take a picture of a moving vehicle in real-time such that no part, notably the vehicle number plate, is missing. In many systems nowadays, the processing time for number plate detection and recognition is less than 50 Ms. The effectiveness of the second and third steps in locating the car number plate and separating each character determines the outcome of the fourth step. These systems employ several strategies to identify the car's license plate and then extract the vehicle number from the resulting image.

1.1 Problem Statement: Several image processing and algorithms must be used within a single application to automatically recognize license plates. To locate the license plate number in each image or video frame, text localization, extraction, enhancement, character segmentation, and recognition processes are used. The entire process of a typical LPR system, from picture acquisition through verification, was only partially covered by the prior studies. This study created a full license plate identification system that functions in real-time and is based on restrictions. The phase of a typical system that takes the longest is license plate localization and extraction. LPR systems need to make assumptions and perform optimizations to locate license plates in real-time. developed and put into action. The phase of a typical system that takes the longest is license plate localization and extraction. For LPR systems to be able to locate license plates in real-time, assumptions and optimizations are needed. However, concurrent with this increase in computational needs. Constraints and prior information are used to reduce this adverse effect. The region is subsequently processed for character segmentation and recognition after the license plate area has been extracted.

1.2 OBJECTIVES: The primary goal of this research project is to thoroughly explore and identify alternate solutions to the character recognition and picture segmentation issues within the scope of license plate recognition. to create a python system capable of both number plate detection and recognition. Find a technique that yields accurate findings for the position of the license plate's area. Create a system that can read the characters from a localized license plate from a video frame. Use the mean squared error approach to identify each character that we have retrieved above.

1.3 EXISTING SYSTEM: There is a "Document Image Recognition" algorithm in the current system. The most efficient method for locating the most similar template for an input image in a database is DIR. The algorithm is created using CBP's global matching data. A method for number plate recognition from a vehicle image was provided by Chunhui C et al. in their article [3]. Python is used to build this technique, and edge detection segmentation and picture pre-processing are used to identify characters.

1.4 PROPOSED SYSTEM: In India, there are essentially two types of license plates: those with black characters on a white plate and those with black characters on a yellow plate. the former for personal automobiles and the latter for working vehicles. These two types of plates are the focus of the system.

2. LITERATURE SURVEY

2.1 PREVIOUS WORK:

With advancements in optical character recognition technology, several improvements in digital image processing have been applied in a variety of industries. Digital image processing has been used in a variety of ways in recent years. OCR was made accessible online as a service (Web OCR) in the 2000s, as well as in cloud computing settings and mobile applications like real-time smartphone translation of foreign-language signs. The creation of a reading machine for the blind, which would enable blind people to have a computer read the text to them aloud, would be the best use of this technology. For most popular writing systems, including Latin, Cyrillic, Arabic, Hebrew, Indic, Bengali (Bangla), and Devanagari, Tamil, Chinese, Japanese, and Korean characters, a variety of commercial and open-source OCR systems are available. Tesseract OCR is the OCR engine employed here. An optical character recognition engine for different operating systems is called Tesseract. It was created by Hewlett-Packard as private software in the 1980s, released as free software under the Apache License in 2005, and has had Google-sponsored development since 2006.

Tesseract was regarded as one of the most precise open-source OCR engines at the time in 2006. Between 1985 and 1994, Hewlett Packard labs in Bristol, England, and Greeley, Colorado, developed the Tesseract engine as a proprietary piece of software. Some more adjustments were made in 1996 to convert it to Windows, and some C++ code was switched over in 1998. Most of the code was written in C, with some additional portions written in C++. Since then, all the code has been modified to at the very least use a C++ compiler for compilation. The following ten years saw very little development completed. In 2005, Hewlett Packard and the University of Nevada, Las Vegas published it as open source (UNLV). Google has supported the creation of the Tesseract since 2006. In 1995, Tesseract was one of the three most accurate OCR engines. It is offered for Windows, Linux, and Mac OS X. However, due to a lack of resources, developers have only thoroughly tested it with Windows and Ubuntu. The car number plate can be effectively found and recognized using the described system on actual photographs. Both traffic management and security may be accomplished with this approach.

2.2 HISTORY

The term "digital image processing" refers to the use of a digital computer to process digital images. To obtain an improved image or to extract some important information, we can also state that it is the usage of computer algorithms. When images were originally delivered by underwater cable between London and New York, one of the early uses for digital images was in the newspaper industry. When the Bart Lane cable picture transmission system was invented in the early 1920s, it took less than three hours to send a picture over the Atlantic, as opposed to more than a week. Pictures were printed using specialized printing equipment that tagged them for cable transmission before reconstructing them at the other end. Some of the original issues with enhancing the visual quality of these early digital images had to do with the distribution of intensity levels and the choice of printing manufacturers. The development of digital computers and related technologies, such as data storage, display, and transmission, has been a prerequisite for advancement in the field of digital image processing since digital images require so much storage and processing power. Some of the original issues with enhancing the visual quality of these early digital images had to do with the distribution of intensity levels and the choice of printing manufacturers. The development of digital computers and related technologies, such as data storage, display, and transmission, has been a prerequisite for advancement in the field of digital image processing since digital images require so much storage and processing power. A finite number of elements, each with a location and a value, make up the digital image. These components are also known as pixels, pels, picture elements, and image elements. A constituent of a digital image is identified by its pixels. Digital image processing is the process of obtaining an image of the text-containing area, pre-processing that image, extracting the individual characters, characterizing the characters in a way that is computer-processable, and identifying those individual characters. In the late 1960s and early 1970s, astronomy, distant Earth resource surveys, and medical imaging all benefited from the development of digital image processing techniques. One of the most significant developments in image processing for medical diagnosis was the development of computerized axial tomography (CAT), sometimes known as computerized tomography (CT), in the early 1970s. A patient is encircled by a ring of detectors during a computerized axial tomography procedure, and an X-Ray source spins around the patient while the detector ring is concentric. The X-ray travels through the object and is picked up by the corresponding detectors in the ring at the other end. This process is repeated as the source rotates. Blurred photos that served as the only documentation of rare artifacts that were lost or destroyed after being captured have been successfully restored using image processing techniques. Blurred photos that served as the only documentation of rare artifacts that were lost or destroyed after being captured have been successfully restored using image processing techniques. Early optical character recognition techniques can be linked to telegraphy and the development of reading aids for the blind. Emanuel Goldberg created a device in 1914 that could read characters and translate them into telegraph code. At the same time, Edmund Fournier dale created the Octophone, a portable scanner that emitted tones that matched letters or characters when it was moved across a printed page. Emanuel Goldberg created what he called a "Statistical Machine" in the late 1920s and early 1930s for employing an optical code recognition method to search microfilm archives. He received US Patent 1,838,389 for the innovation in 1931. Buying the patent was IBM. OCR can be utilized in internet-connected mobile device applications that extract text from images taken with the device's camera with the introduction of smartphones and smart glasses. These devices will normally use an OCR API to extract the text from the picture file taken if their operating system does not provide built-in OCR functionality. At the same time, Edmund Fournier d'Alene developed the Octophone, a portable scanner that, when moved across a printed sheet, released tones that corresponded to specific letters or characters. In the late 1920s and early 1930s, Emanuel Goldberg developed what he called a "Statistical Machine" for using optical code recognition to search microfilm archives. In 1931, he was awarded US Patent 1,838,389 for the invention. IBM purchased the patent. With the advent of smartphones and smart glasses, OCR can be used in internet-connected mobile device applications that extract text from photos acquired with the device's camera. If the device's operating system does not offer built-in OCR functionality, it will often use an OCR API to extract the text from the photo file captured.

3: PROPOSED METHODOLOGY

3.1 Problem Formulation

It is seen that the security forces and authorities face problems whenever security forces chase a vehicle, or they can't catch a vehicle that broke traffic rules. Authorities find it very hectic on a busy day to log the vehicle numbers manually in a parking lot. So, to make the entire process autonomous, we can install this system to automatically detect the vehicle which breaks the traffic rules, take a picture of it, and store the number in the database to find the respective owner afterward. The system can be used in parking to take the picture of the vehicle and log the vehicle number in the database (or the cloud, if connected to the internet). This technology reduces the unnecessary hectic manual work required on any busy day, saves the labour cost, and is far more efficient than humans. The number of any vehicle once obtained as text can be displayed, saved in the database, or can be searched through the entire database for the details. This project is so versatile that it can be used as an entire application once converted to software or can be used as a part of any big project.

3.2 Project Objective

● Image Acquisition using the computer's primary camera.

● Image Enhancement and pre-processing to improve the quality of the image and convert the image to binary scale to use it in contour extraction.

● Extract the number plate region from the binary image and display it separately. In the Output Unit, the input image is taken from the input unit and then processed in the processing unit. In the processing unit, all sorts of enhancement and extraction are done and then the number on the license plate is extracted using Optical Character Recognition and then it can either be stored in a database or be displayed on a display device or both or can be used to excite an actuator.

3.3 Details of Processing

● Basics of Digital Image Processing: The image of a vehicle whose number plate is to be recognized is taken from a digital camera which is then loaded to a local computer for further processing. OpenCV (Open-Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. In simple language, it is a library used for Image Processing. It is mainly used to do all the operations related to Images. Python, being a versatile language, is used here as a programming language. Python and its modules like NumPy, Skippy, Matplotlib, and other special modules provide the optimal functionality to be able to cope with the flood of pictures. To enhance the number plate recognition further, we use a median filter to eliminate noises, but it not only eliminates noise. It concentrates on high frequency also. So, it is more important in edge detection in an image, generally, the number plates are in a rectangular shape, so we need to detect the edges of the rectangular plate.

Image Processing mainly involves the following steps:

1. Image acquisition: This is the first step or process of the fundamental steps of digital image processing. Image Acquisition is the capturing of an image by any physical device (in this case the primary camera of the computer) to take the input as a digital image in the computer.

2. Image Enhancement: Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Such as changing brightness & contrast etc. In this step, the quality or rather the clarity of the input image is enhanced, and the image is made clear enough to be processed.

3. Morphological Processing: Morphological operations apply a structuring element to an input image, creating an output image of the same size. The image is converted to a binary image, making it more to apply structural extraction to the image and extract any structure related to a particular mathematical model from it, in this case, a license plate.

4. Segmentation: Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward a successful solution to imaging problems that require objects to be identified individually.

5. Representation: Representation and description almost always follow the output of a segmentation

stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. Description deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

6. Recognition: Recognition is the process that assigns a label, such as, "Plate" to an object based on its descriptors.

4. PERFORMANCE EVALUATION

4.1 Advantages of Using Python for Computer Vision

Our society is more technologically advanced than ever, and the advantages of using Python for Computer Vision (CV). The human brain interprets and makes sense of what the human eyes can see. As technology advances, CV is one of the frontiers and potentially revolutionary technology in computer science. It is a subset of machine learning wherein the latter is also the subset of Artificial Intelligence (AI). Most start-ups wanted to be part of the AI-fuelled companies to establish business relevance and growth. It needs a competent software development team with a compatible programming language such as Python programming language in using systematic efforts to achieve greater productivity.

Python is a programming language that aims for both new and experienced programmers to be able to convert ideas into code easily. It is currently the most prevalent, mature, and well-supported among programming languages in machine learning, that is why many developers use Python for CV.

Computer Vision, on the other hand, allows computers to identify objects through digital images or videos. Implementing CV through Python allows developers to automate tasks that involve visualization. While other programming languages support Computer Vision, Python dominates the competition.

here is a list of the advantages of using Python for Computer Vision:

1. Ease of coding

"Code as plain English" is Python's primary goal. This allows programmers to focus on the design and not on coding. This is perfect for those who are just getting started with machine learning or basic programming. This advantage is very beneficial, especially when faced with complex scenarios.

2. Fast prototyping

Since you can focus more on the design, you can now experiment with more design ideas. Python is well-suited for implementing new features. Libraries like OpenCV are written in C++ and make Python have slower runtime as it will still call C/C++ libraries. This means you will have the development advantage from Python while you can have performance optimization from C++.

3. Vast libraries for machine learning

Python is commonly used for machine learning. Data scientists invest their time contributing since it's easy to code, and it's free. CV developers don't have to worry much about projects that they're working on since most of their cases are already covered by Python libraries.

4. It is open source

Python is free, unlike MATLAB, which also specializes in data analysis, exploration, visualization, etc. For Python, all you need is a computer, and you are good to go. You can even deploy your work for free on sites like Python anywhere.

5. It can be directly integrated with web frameworks

Python has mature web frameworks like Django. It aims for fast development time and neat and realistic designs. Also, Python has micro frameworks that are just as functional as their larger counterparts.

4.2 Python Applications:

1. Web development – Web frameworks like Django and Flask are based on Python. They help you write server-side code which helps you manage the database, write backend programming logic, map URLs, etc.

2. Machine learning – There are many machine learning applications written in Python. Machine learning is a way to write logic so that a machine can learn and solve a particular problem on its own. For example, products recommendation on websites like Amazon, Flipkart, eBay, etc. is a machine learning algorithm that recognizes users' interests. Face recognition and Voice recognition in your phone is other example of machine learning.

3. Data Analysis – Data analysis and data visualization in form of charts can also be developed using Python.

4. Scripting – Scripting is writing small programs to automate simple tasks such as sending automated response emails etc. Such types of applications can also be written in Python programming language.

5. Game development – You can develop games using Python.

6. You can develop Embedded applications in Python.

7. Desktop applications – You can develop the desktop application in Python using libraries like TKinter or QT.

4.3. Specific Requirements:

4.3.1. Non-Functional Requirements:

a) Usability: The user can give a normal clarity image as input and can interpret output with minimum knowledge of basics.

b) Performance: The proposed system after giving the required input takes less time to process and produces output accurately.

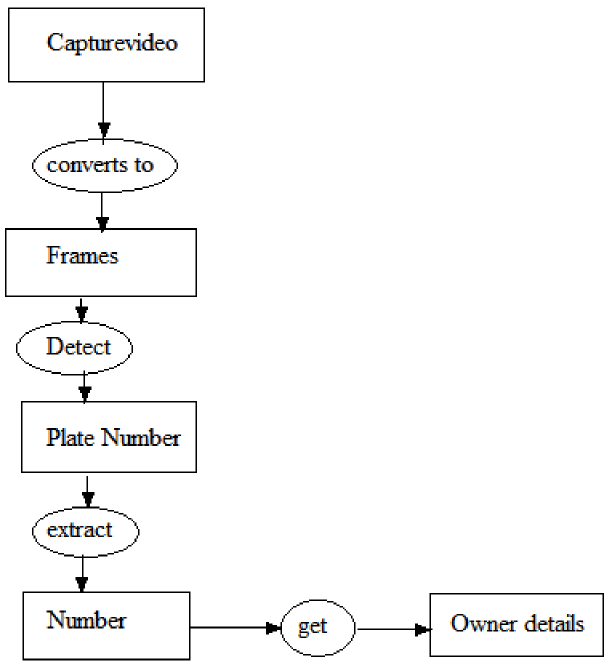
c) Supportability: The system can be expanded further to add new factors that may affect it. The proposed system can work in any environment without any problems.

4.3.2. Design Requirements:

- ❖ Hardware Requirement
- ❖ Processor :Pentium CoreI5 and Higher
- ❖ RAM : 4GB or more.
- ❖ Hard Disk :500GBor more.
- ❖ Monitor : 15-inch Color Monitor
- ❖ Keyboard : 102/104 Keys
- ❖ Mouse : Optical Mouse

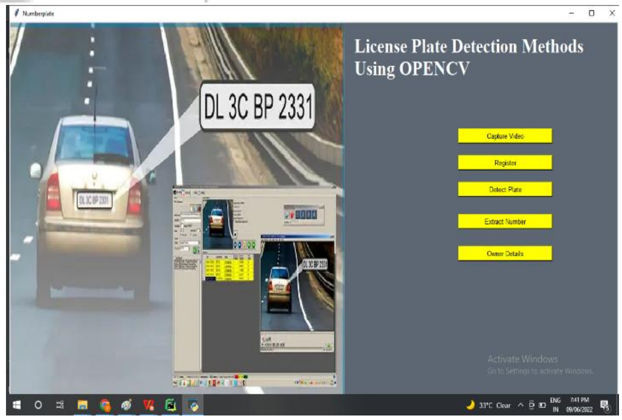
- ❖ Software Requirement
- ❖ Operating System: Windows XP/7/8/10
- ❖ Front End : Python
- ❖ Back End :SQL SERVER 2008

❖ Data Flow Diagram (DFD):



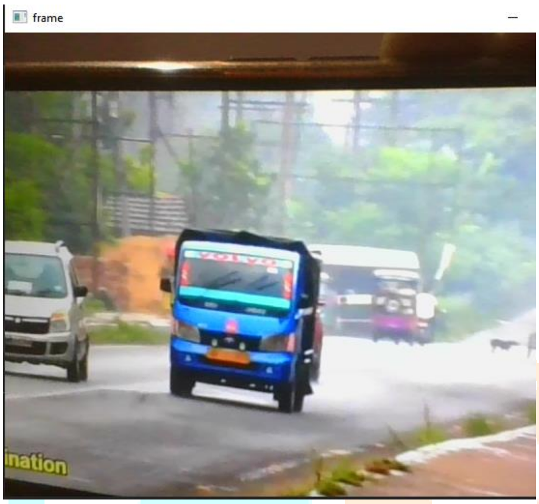
A Data Flow Diagram (DFD) is a diagrammatic representation of the information flows within a system, showing how information enters and leaves the system, where information is stored. Data flow diagrams can be used to provide a clear representation of any business function.

5. RESULTS



Capture: A high-resolution camera is used to take the picture of the vehicle. An IR camera is a preferable option. Regarding the license plates, the camera can be pitched and rolled.

Pre-processing: Pre-processing is the use of predetermined algorithms to improve the quality of an image. It is a crucial and typical stage in every computer vision system. Pre-processing for the current system consists of two steps: Resize - The camera's image size can be huge, which can make the system slow. It has to be scaled down to a workable aspect ratio. Convert Colour Space - Photographic or infrared cameras will either collect images in raw format or encode them according to a variety of multimedia standards. Typically, these pictures will be in three-channel RGB mode (viz. red, green, and blue).



License plate extractor: The most important step in the license plate recognition system is the license plate extractor. Various strategies are used in this procedure to identify and extract license plates from images. There are two stages to this process.

a. detection of license plates using Haar-like characteristics

Haar-like features are employed in image processing methods to identify objects in images. The Haar-like features are used for this purpose and no additional processing is done if our suggested system is chosen to merely detect license plates. This method is outdated, time-consuming, and moreover requires a sizable database to hold the samples that have been collected—roughly 10,000 photos of the plates and characters.

b. Edge Detection for License Plate Recognition On the other hand, if our suggested system needs to identify license plates, the binary image is made from the image.

Following that, the processes below are carried out to extract the license plate from the binary image:

1. A binary image is searched for four connected points.

2. The width/height ratio is compared to those linked locations.

3. The image's license plate region is extracted.

4. The extracted license plate undergoes a transformation. The recovered license plate is then sent to the following component for additional processing. This method is rapid, requires little memory and execution time, and has a high-efficiency level. We have used this strategy in our project because of this.

CHARCTER SEGMENTATION

To remove extraneous data, more image processing is done on the retrieved license plate in this section. Only the characters that pertain to the license number are recovered from the character stream in the extracted license plate. Additionally, this was accomplished by matching the width-to-height ratios with the contours seen on the retrieved license plate.

• **Owner details**: This module displays information on the owner of the vehicle, as well as information about emissions testing and insurance.

6. CONCLUSION

Our research focuses on developing a plate localization and extraction technique from vehicle number plates. First, the location of the plate is extracted. Next, the plate characters are separated into their distinct parts using a variety of pre-image processing algorithms. Lastly, the segmented numbers are correlated with the standard templates in the library. The suggested method is tested with a variety of vehicles, including four-wheelers, and with yellow and white backgrounds. The segmented characters are detected using the Template Matching Method. The segmentation of the license plates with extraneous data is also very accurate.

Each independent function has an error limit of 8% maximum. The error rate for all 200 photos is 17%. A total of 2161.36 seconds passed during recognition. Each image takes an average of 10.80 seconds to be recognized. Deterministic vital criteria for the program's proper operation are the plate status, the surroundings, and the hardware utilized to take images. Successful recognition is almost always ensured by effective image pre-processing**.**

6.1 FUTURE WORKS

To improve the success of the program is needed small improvements at each stage. The image must be centered, fixed, and evenly illuminated during the catch. Differentiate the car colour of the image under study, i.e., to adapt the pre-processing at colour because several problems appear in the plate location when the cars are white and silver. Also is possible to do an adaptive mask Depending on the picture. Improve the choice of level to the threshold and not lose information about the shape of the characters found. Through an adaptive threshold that divides the image into sub-images and chooses the most appropriate level in each case, this solution is associated with a significant increase in execution time. Once characters are segmented the main mistake is that these are distorted or incomplete. Adding a process of reconstruction and the calculation of the Hough transform increases the success rate. There are several solutions that can be applied but keep in mind what you want to sacrifice, the run time, the quality of image objects, the degree of difficulty of implementation, or the hardware and quality cost, among others.

7. REFERENCES

[1] A. Conci, J. E. R. de Carvalho, T. W. Rauber, "A Complete System for Vehicle Plate Localization, Segmentation and Recognition in Real Life Scene", IEEE LATIN AMERICA TRANSACTIONS, VOL. 7, NO. 5, SEPTEMBER 2009.

[2] Ahmed Gull Liaqat, "Real Time Mobile License Plate Recognition System" IEEE White paper California, VOL.2 2011-12-05, Linnaeus University.

[3] Ondrej Martin sky (2007). "Algorithmic and mathematical principles of automatic number plate recognition systems" (PDF). The Brno University of Technology.

http://javaanpr.sourceforge.net/anpr.pdf. [4] P. Kreling, M. Hatsonn "A License Plate Recognition algorithm for Intelligent Transportation System applications". The University of the Aegean and the National Technical University of Athens. 2006. Archived from the original on 2008-04-20.

[5] K.M Sajjad, "ALPR Using Python and Open CV" Dept Of CSE, M.E.S College of Engineering Kuttipuram, kerala.2008-06-21.

[6] Nicole Ketelaars "Final Project: ALPR", 2007-12- 11.