PREPRINT

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Utilising Innovative Image Preprocessing Techniques in Vehicle Plate Detection and Recognition

Soheir Noori [1]*, Asam Almohamed[2], Elham Mohammed Thabit A. Alsaadi[2], and Azhar Abbas[2]

[1]*Department of Computer Science, College of Computer Science and Information Technology, University of Kerbala, 56001 Karbala, Iraq*
[2]*Department of Information Technology, College of Computer Science and Information Technology, University of Kerbala, 56001 Karbala, Iraq*

ABSTRACT

Vehicle number is an essential issue for intelligent traffic monitoring systems. It plays a significant role in the transportation sector. In addition, recent population growth and human needs have increased due to the use of vehicles. Therefore, controlling traffic is becoming a complex problem to solve. An Automated Plate Number Recognition System (ANPR) program is needed for traffic control. This research includes developing and implementing a multi-stage preprocessing methodology with various techniques to identify vehicles' plate numbers. The detection system can achieve successful vehicle management: car parking as a case study. The proposed ANPR system employs a template matching algorithm based on a multi-stage preprocessing technique for character identification. Those techniques include noise filtering, component isolation, and correlation-based similarity measurements. The methodology process steps are grayscale conversion, binarisation, bilateral filtering, edge detection, contour localisation, segmentation using connected components, and character recognition through cross-correlation. The system was evaluated on 20 Iraqi one-line plates and achieved a recognition accuracy of 91%.

Keywords: Accuracy rate, detection and recognition phase, parking management system, plate number, preprocessing techniques

ARTICLE INFO

Article history:
Received: 27 May 2025
Accepted: 05 May 2026
Published: 12 June 2026

DOI: https://doi.org/10.47836/pjst.34.3.05

E-mail addresses:
soheir.noori@uokerbala.edu.iq (Soheir Noori)
asam.h@uokerbala.edu.iq (Asam Almohamed)
elham.thabit@uokerbala.edu.iq (Elham Mohammed Thabit A. Alsaadi)
azhar.ali@uokerbala.edu.iq (Azhar Abbas)
* Corresponding author

INTRODUCTION

Intelligent transportation systems include plate recognition and intelligent infrastructure systems (Anagnostopoulos, 2014; Anagnostopoulos et al., 2006).

e-ISSN: 2231-8526

An essential requirement for observing and analysing transportation activity is the implementation of a monitoring system (Castello et al., 1999; Duan et al., 2004; Sarfraz et al., 2013). ANPR systems have different names due to reflecting different purposes, like mobile license plate reader (MLPR), vehicle recognition identification (VRI), license plate recognition (LPR), car plate recognition (CPR), and license plate reader (LPR), Highway road tolling systems (Al-Shemarry et al., 2022; Panahi & Gholampour, 2017; Song & Sarker, 2014), security systems (Al-Shemarry & Li, 2020) parking management systems, traffic management systems (Ojeniyi et al., 2019).

An Automatic Number Plate Recognition (ANPR) system relies on advanced image processing techniques to identify and extract license plate information from vehicle photographs. Thus, a plate number detection system is required to support the car parking system. It consists of four stages: input image preprocessing, localisation of the license plate, license plate segmentation, and recognition. Each car image carries a unique number. Thus, implementing an ANPR system can provide practical benefits for developing a parking management system. The research aims to develop a detection system to recognise car plate numbers from images. The proposed system was designed and tested on new English letter Iraqi oneline license plates. The Python programming language is used, and the OpenCV library for all image processing tasks. Localising car numbers is essential for intelligent monitoring systems (Angelova et al., 2015; Arafat et al., 2019; Azam & Gavrilova, 2017; Caplan et al., 2011; Wang et al., 2020; Yang et al., 2013; Yousef et al., 2015). The main objectives of this research are as follows:

1. Developing an ANPR software using template matching techniques.
2. Detecting unacceptable activities in transport systems to reduce the required efforts.
3. Increasing the accuracy of the existing ANPR system using Iraqi English car plate numbers.

Moreover, this research is essential to develop and measure the accuracy of the ANPR system. This is the central role of the detection system in controlling traffic flow and detecting vehicle plate numbers quickly, which is used for security purposes. The research outcome could further help detect or verify vehicle plate numbers. The system could also be beneficial for assisting law enforcement bodies, such as the traffic police, to detect and trace vehicle plate numbers easily. The developed ANPR system recognises vehicle plate numbers in good condition using one line or row of vehicle plate numbers.

Despite the wide adoption of ANPR systems, several challenges remain unresolved, particularly in real-world environments. Variations in lighting conditions, image quality, plate orientation, and background noise can significantly affect recognition performance. In addition, many existing approaches rely heavily on machine learning models that require large annotated datasets and high computational resources, which may not always be available. In the context of Iraqi license plates, additional challenges arise due to variations in plate conditions, capture environments, and character similarities. These limitations motivate the need for a more efficient and robust approach that can operate under constrained conditions. Therefore, the main objective of this study is to develop a reliable ANPR system based on multi-stage preprocessing and deterministic template matching techniques. The proposed approach aims to enhance recognition accuracy while maintaining computational simplicity, making it suitable for practical deployment in parking management systems.

The Suggested System Components

The ANPR system software functions on a workstation and can establish connections with other applications that utilise the same databases. By employing multiple image enhancement algorithms, this process improves and standardises the input image containing the license plate number. Subsequently, it employs diverse extraction and classification approaches to extract and identify license plate information.

ANPR systems are often deployed using two main approaches. The first method enables real-time implementation of the whole license plate detection procedure. The second transmits each image the system camera captures across several channels to a remote computer. Next, the detection operation is carried out with a time delay, depending on the rate of the processing procedures utilised. The ANPR system must go through four main steps to recognise a license plate. These steps are stated below as shown in Figure 1.

Capturing Car Images

As a first step, photograph the car using a camera. The accuracy of any detecting system is determined by the quality of the camera, which encompasses its type, resolution, light, shutter speed, and placement technique. The distance between the car and the camera can sometimes cause issues, including low-quality or poor-quality photos.

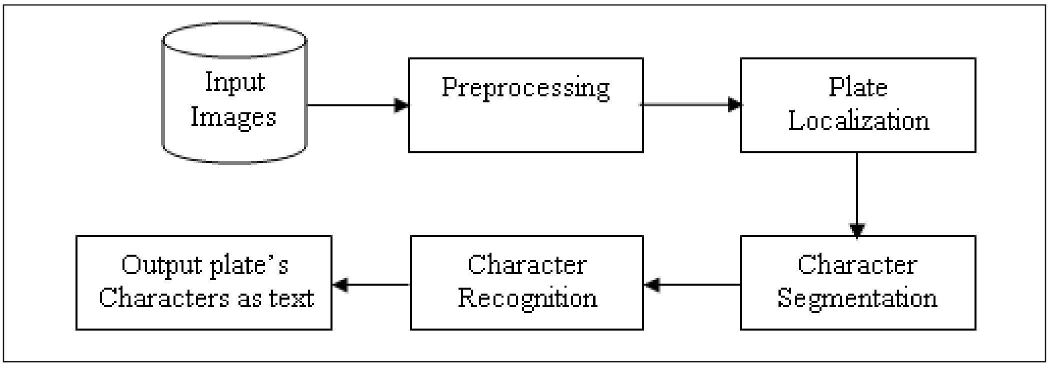

Figure 1. Steps to recognise a license plate in the ANPR system

The Preprocessing Stage

After the picture of the car is taken, more processing is done. Minimising the background noise of license plates improves the processing speed of succeeding stages by optimising the texture pattern. The automotive image undergoes several processing steps, including resolution expansion, elimination of noise, and colour transformation from RGB to monochrome or binary form. Various preprocessing algorithms are used for these tasks.

The Localisation and Extraction Stage

Regions of interest (ROIs) are taken from an automobile picture after preprocessing. This stage also has an impact on the precision of the ANPR system. The vehicle photograph undergoes central cropping and localisation, extracting all remaining pixels using multiple approaches. This will shorten the processing time of the extraction stage, mainly if those techniques use unsupervised learning algorithms (Ojeniyi et al., 2019). However, the license plate may appear anywhere in the picture. Consequently, specific characteristics such as boundaries, edges, colour, background, and textural elements impact the license plate extraction process.

The Classification and Detection Stage

The third step involves applying various supervised learning techniques to determine the number of license plate areas from the extracted data. The quality of the extraction techniques and preprocessing used to acquire the license plate areas will determine this stage (Abdulkadhim et al., 2024; Lee et al., 2013; Littlewort et al., 2004). Classifiers are used to create trained models and then to classify the pertinent extracted features from the license plate. The detectors could pick up several complex license plate details.

Several researchers have employed preprocessing techniques or unsupervised learning algorithms to detect multiple license plate issues. As a result, the detection system's processing time increases (Patel et al., 2013). The output of this stage functions as a decision mechanism to detect regions in an automobile image that correspond to license plates and ascertain their authenticity. The objective of utilising supervised learning algorithms during the testing phase is to achieve optimal system performance with the lowest execution time.

The Segmentation and Recognition Stage

The final stage of the ANPR system includes recognising the symbols retrieved from the observed license plate region. This stage employs various segmentation and recognition techniques, including template matching methods and classifiers such as artificial neural networks and fuzzy classifiers. In this paper, the license plate number is detected from automobile photographs using the template matching method.

During this phase, the license plate number is converted into text encoded by a machine. In this case, the license plate image is used to identify the plate numbers using optical character recognition (OCR). The stage structure of the ANPR system is illustrated in Figure 2.

The detecting system's efficacy is contingent upon the stability and dependability of every single phase. To apply the license plate recognition technology for parking management systems, the primary goal of this research is to create techniques that detect license plates from automobile photos in a more accurate manner, as shown in Figure 3.

Image Manipulation Operations

Image Binarisation

A binary image is a computer image with each pixel having one of two possible values. Any two colours can be used, although black and white are the most common pairing for a binary image. The foreground colour in the picture is used for the objects, whereas the background colour is used for the remainder of the image.

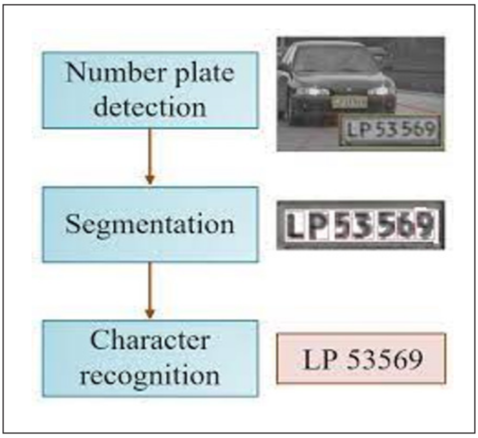

Figure 2. The architecture of an ANPR system

Figure 3. ANPR for parking management systems

Bi-level or two-level images are other terms for binary images. This indicates that a single bit (0 or 1) is recorded for every pixel. For this idea, the terms black-and-white, B&W, monochrome, or monochromatic are frequently used. Yet, they can also refer to any images with a single sample per pixel, such as grayscale images.

In digital image processing, binary pictures frequently appear as masks or as the outcome of specific processes like segmentation, thresholding, or dithering. Only some input/output devices, including fax machines, can handle bi-level images, laser printers, and bi-level computer displays. A purely binary image is defined as any combination of two colours where each pixel is represented by a single bit, either 0 or 1 or 0 and 1.

Image Segmentation

Segmentation is dividing a digital image into many segments in image processing. These segments are always called sets of pixels or super pixels. Segmentation is a technique that makes an image's representation simpler, more understandable, and easier to evaluate. Finding boundaries and objects in images, such as lines, curves, and so on, is usually accomplished using image segmentation. Image segmentation gives each pixel in an image a label, so those pixels share certain visual traits. A collection of contours retrieved from the image or segments that cover the entire image are the outcomes of image segmentation. Each of the pixels in a region is similar concerning some characteristic or computed property, such as colour, intensity, or texture.

Image Filtering

By comparing a pixel with its surrounding pixels in some way, images can be improved (or otherwise distorted, twisted, or mutilated) through image filtering. There are two kinds of picture filtering. First, height-based filtering, which is limited to images with indexed colours and those with 256 colours, is a typical example. Every colour has an index number between 0 and 255, which can be referred to as the "height" of the colour on a map. Second, colour-based filtering employs the actual red, green, and blue components of the colour for calculations, rather than the colour's index number. Both full-colour and 256-colour photos can be used with this.

Image Normalisation

Refining the characters into a block with no extra white space (pixels) on any of the four sides of the character is known as normalisation in image processing. After the image has been processed through the binarisation and segmentation processes, the normalisation procedure is applied if there are any remaining white spaces in the acquired image. Typically, there are more white than black spaces in the box following the binarisation and segmentation of a picture into many character blocks.

In other words, normalisation removes all white space, cuts the character block to fit the character, and then applies this approach to ensure that every character is the same size. For every block, the characters are 36 x 18, much like in this ANPR study. The extracted character is then cut and matched with an equal size, as shown in Figure 4.

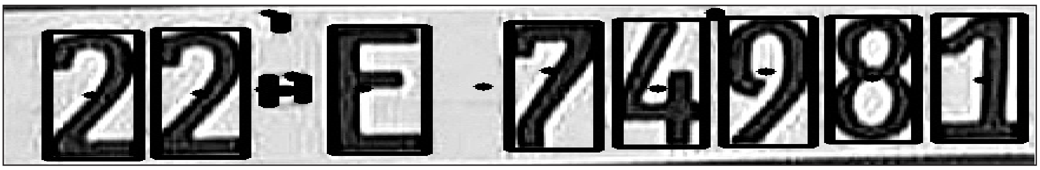

Figure 4. Equal-sized character with blocks after the normalisation process

The Correlation Operation

One helpful method for recognising images is correlation. A correlation is a statistical method for determining the existence and strength of a relationship between two variables. For instance, there is a correlation between height and weight: taller people typically weigh more than shorter ones. The template matching process is the last step in the ANPR. During the character recognition stage, the accuracy of the result with the characters recorded in the database for this ANPR research is measured by comparing the similarity and identifying the best match using the template matching technique.

In image processing, the correlation algorithm is typically the most effective technique to employ at this stage. The cross-correlation and auto-correlation functions are the two correlation functions used in this study. The most effective method to gauge how similar two signals are is to use the cross-correlation function. The characters are then used to carry out the template matching procedure and compared with the characters in the database, as shown in Figure 5. This approach calculates the correlation coefficient between them using a set of known photos of the same size and some unknown photographs of a portion of an image with the highest correlation coefficient. There are two types of correlation:

The Auto-Correlation Function (ACF)

The cross-correlation of a signal with itself is known as autocorrelation. Simply put, it is the degree of resemblance between observations based on their time interval. It is a mathematical tool for locating recurring patterns, including a periodic signal hidden by noise, or identifying a missing fundamental frequency indicated by the signal's harmonic frequencies. In signal processing, it is frequently used to analyse functions or sequences of values, like time domain signals. The single-signal ACF yields data regarding the structure of the signal or its behaviour in the time domain.

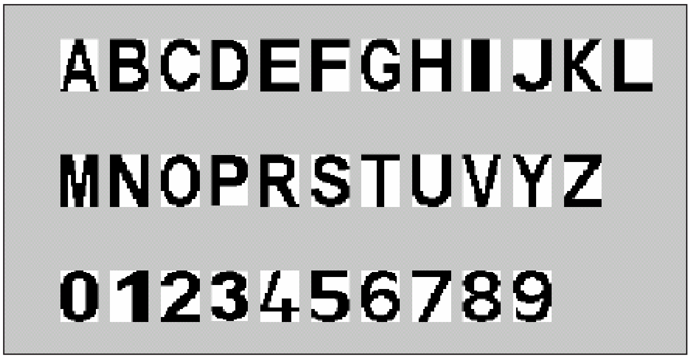

Figure 5. Database characters

The Cross-Correlation Function (CCF)

A measure of the similarities or shared characteristics between two signals is the CCF. Cross-correlation is a signal processing metric that quantifies how similar two waveforms are by applying a time lag to one of them. Another name for this is a sliding inner product or sliding dot product. It is frequently employed to look for a shorter, well-known component in a long-duration signal. It can also be used for electron tomographic averaging, single particle analysis, and pattern identification.

METHODOLOGY

This research aims to use Iraqi car license plates to test the effectiveness of the ANPR technology for parking management. The author covers every process step in this section, including data collection, specifications and special needs for creating ANPR, experimental procedures used, and anticipated results. As shown in Figure 6, the approach diagram for the ANPR system has been created and designed in this stage. To construct and develop the ANPR algorithm, a thorough understanding of the fundamental principles involved in this research is necessary. This section will review several related approaches that can be used to create the ANPR algorithm for this study. Its three main stages are plate region extraction, image segmentation, and character identification.

Although the proposed system does not employ machine-learning classifiers, it incorporates an intelligent deterministic mechanism through template matching and correlation-based decision making. The system automatically filters noise, isolates connected components, and compares each extracted character against reference templates using normalised cross-correlation. This approach provides intelligent classification behaviour without the need for model training.

Capturing Car Images

In order to maintain the accuracy of the threshold values the system employs, a stationary camera placed on the parking station door will be used to record the input image. The distance between the camera and the automobile will be fixed and designated at the front of the station door. Since the suggested technique in this system will produce less accurate results if the input image is oriented, the collected image must be in a static, ideal orientation as shown in Figure 7.

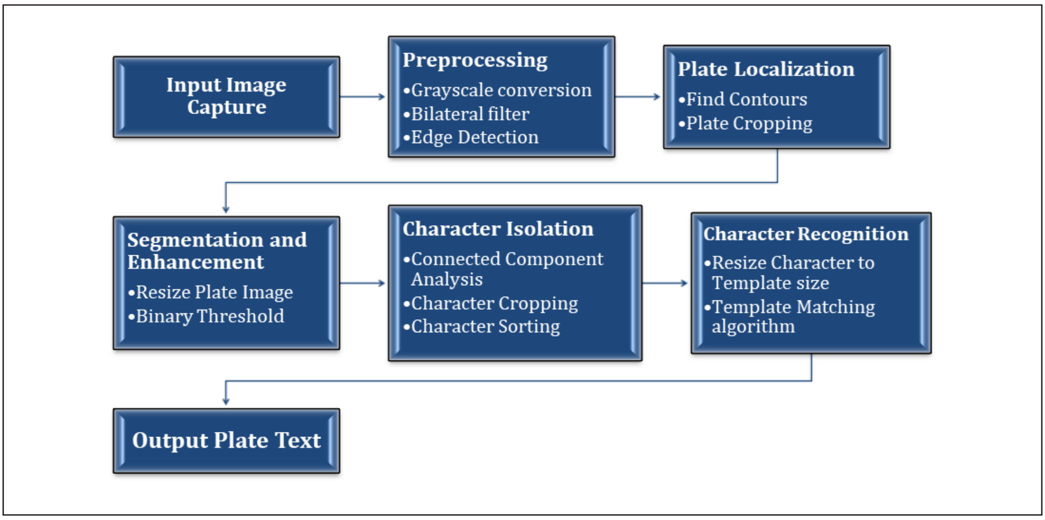

Figure 6. ANPR Methodology

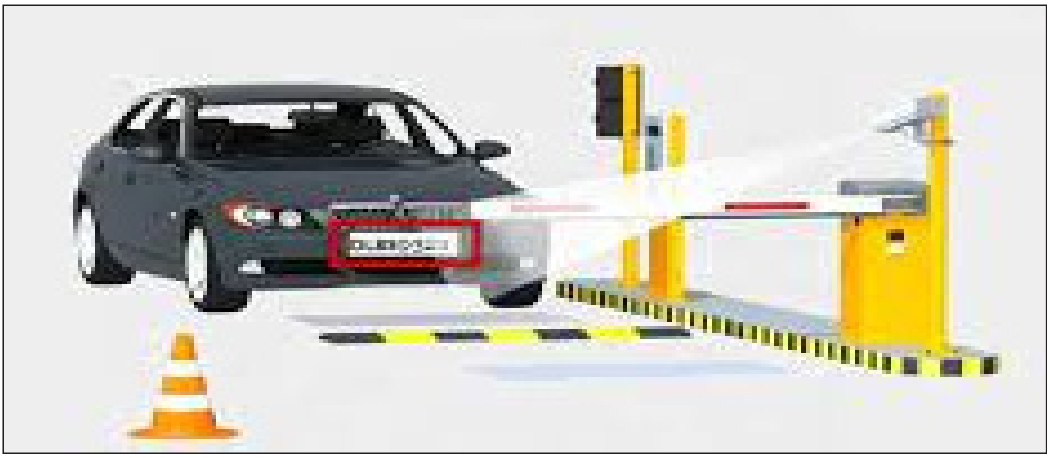

Figure 7. Image capturing stage

Car Image Preprocessing

At this stage, before developing another ANPR stage, a few processes should be implemented in the Iraqi plate region. Thus, several image preprocessing methods should be applied to the car image following the sequence below:

The Grayscale Process

The obtained RGB automobile image needs to go through the grayscale procedure, which turns the three-channel colour image into a grayscale format. This is necessary because, when subsequently converted to a binary scale, the grayscale image might sharpen and intensify every pixel. This grayscale process will use several image processing techniques to create the grayscale image.

The Binarisation Process

After being converted to grayscale, the car image must be converted into a binary form. By using the 0s to represent white pixels and the 1s to represent black pixels, the data from the grayscale format is binarised into a single black-and-white image. This should be created following the grayscale procedure to sharpen the image and improve the binarisation results.

The Car Image Filtering Process

Bilateral filtering is the next step applied to the binarised automobile image. It is used to reduce and eliminate undesirable noise and spots in the image. A specific parameter is used in the execution process to achieve the desired clearing image level. Following this procedure, the image ought to appear sharper and cleaner. Furthermore, it effectively reduces noise while maintaining sharp edges by ensuring that only nearby pixels with intensities comparable to the central pixel are considered for blurring. This is in contrast to the standard Gaussian filter, which ignores pixels that are nearly identical in intensity and thus blurs edges—something that we do not want to happen.

The Car Image Edge Detection Process

The edge detection process will start after the auto-mobile image filtering step. In this experiment, the image processing technique known as "edge detection" is used to determine the start and end locations of each character's white pixel to determine the character's horizontal and vertical bounds. This multi-step edge detection approach uses bilateral filtering and a Gaussian smoothing filter to denoise the image. However, better outcomes can be obtained by applying the Bilateral Filter before the Canny algorithm, particularly when the automobile image has a lot of noise or has poorly defined edges. As a result, everything will appear black except for the margins, which will appear white.

In the upcoming processing stage, it will be helpful to find all closed shapes in the input image using the edges as shown in Figure 8.

Plate Localisation

The plate image cropping procedure, where the phase one output result is displayed, is the most critical step in this initial phase. As previously mentioned, the objective of this procedure is to exhibit solely the plate region area containing the processed characters. Localising the license plate using the contours tool is the first step in cropping the image. With the parameters provided, it locates every closed shape in an input image as shown in Figure 9. Find Contours will use the output image from the clever algorithm as its input.

Furthermore, all of the forms discovered will be ordered from largest to smallest, selecting only the first 32 shapes because license plates are typically found in the range of 0 to 32. Any shape outside this range is too small to be considered a legitimate plate. The width, height, and x, y position data will be taken from each form to separate the actual plate shape from the others, and then filter the plate through a predetermined threshold. First, if the width is less than 420 pixels, and second, if the difference between the height and width/4 is less than 35 pixels, the threshold will be evaluated in both scenarios. The pixel value is then set in the function parameter, and the picture cropping procedure crops the area and shows it in its original colour. At the same time, the remaining portion of the image turns black as soon as the scanner detects the pixel region.

Segmentation and Enhancement

The goal of this phase is to group all chopped characters into blocks of uniform size. The procedure entails multiple critical operations and algorithms, including contour detection, edge detection, and character extraction. The intended outcome is that each character will fit within a specific size border box, as shown in Figure 10.

After resizing the license plate to make it larger, the noise produced by the resize function will be eliminated, and the image will be blurred again using the Gaussian filter. Next, thresholding techniques are used to accomplish plate image segmentation. Otsu's approach has been used to apply a threshold in order to convert the image to binary. The algorithm's output is a single intensity threshold that divides pixels into foreground and background classes. Because the license plate letters will noticeably stand out, higher recognition results will be obtained.

Moreover, the threshold has been reversed, making the text seem white and the background dark because Connected Components can only detect white pixels. The fitting approach technique will be applied to the character resizing in order to standardise the character size with a given parameter. In this experiment, every character must have a height and width of 42 x 24.

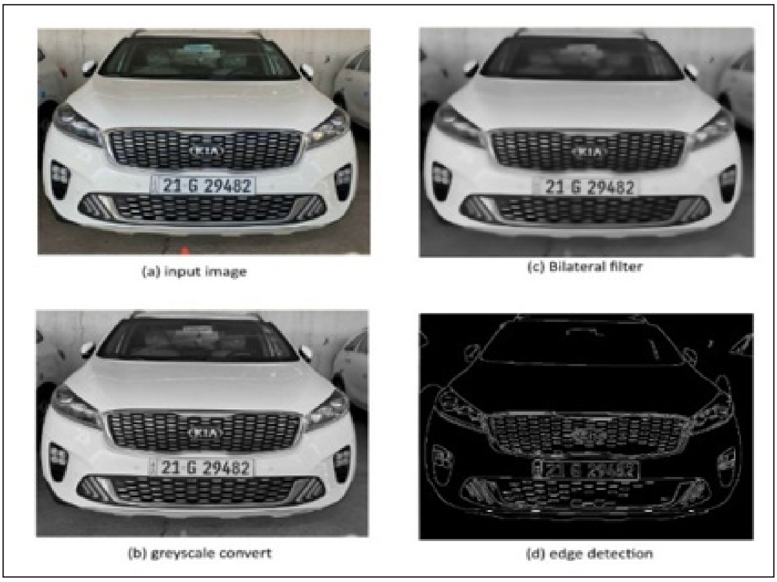

Figure 8. Preprocessing stages

Figure 9. Drawing found closed shapes

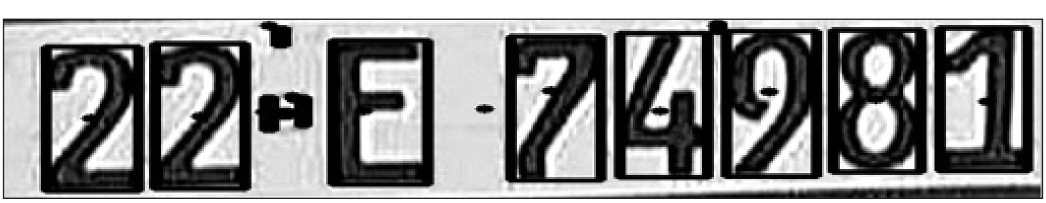

Figure 10. Bounding box for each character on the license plate number

To do this, the process will first resize every character to this size, and then the bounding process will be executed to fit the sized image inside each block container using an algorithm. The final step in this process involves extracting and displaying every character in each boundary box as shown in Figure 11.

Character Isolation

The connected component analysis function has been used to isolate characters in the license plate. It returns four parameters (*numLabels*, *labels*, *stats*, *centroids).*

numLabels	:	The number of the connected components.
labels	:	The actual pixel data of the connected components.
stats	:	Various statistical data about each connected component (location, width, height, pixel area).
centroids	:	The *xy* centroids for each connected component.

All the connected components will first be looped through, excluding the first one, as the first connected component in a given image is the image itself (background), which will not be used here when looping. Threshold conditions will be applied to filter out the connected components that are the actual characters in the license plate. The threshold will be checked against data obtained from the stats variable that the connected component function returns. The exact threshold values are obtained by observing each connected component's data and a bit of trial and error. To ensure the thresholds are caught right, rectangles will also be drawn around each connected component representing a character as shown in Figure 12.

These thresholds are not foolproof; some may pass as valid character shapes. However, this issue will be solved using other thresholds in the character recognition stage. The connected components of the detected character may not be listed in the correct order as they appear (left to right in our case), so a sorting algorithm will be applied to sort the detected components from left to right using the component's X location. A connected component with an X value that goes into 0 and negative values should be sorted to the left of a component with an X value that goes into positive. Now, all characters are segmented and properties sorted as shown in Figure 13.

The numLabels parameter represents the total number of detected connected components in the image, including the background. This value helps determine how many potential objects (characters or noise regions) are present. The *centroids* parameter provides the geometric centre (x, y coordinates) of each detected component. These coordinates are useful for locating each character within the plate region and for maintaining their correct spatial order. By using centroid positions, the system ensures that characters are arranged from left to right before recognition, which is essential for accurate plate interpretation.

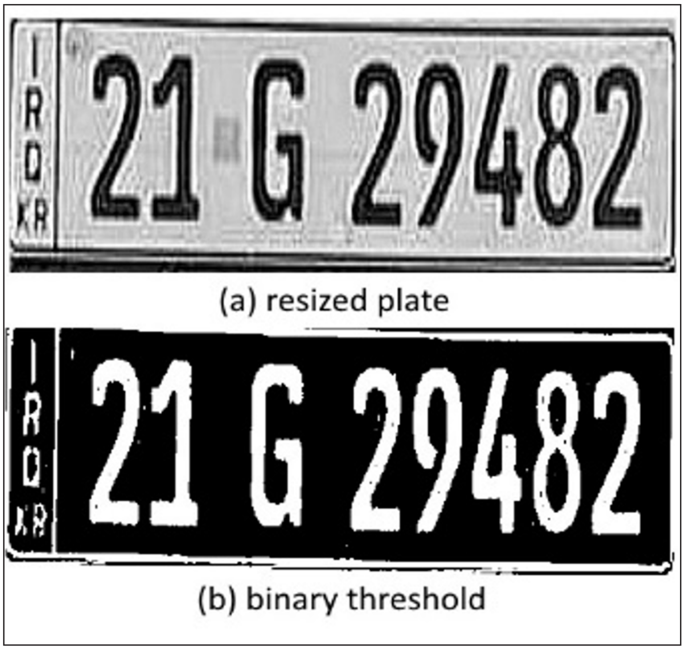

Figure 11. Plate a) resize and b) threshold

Figure 12. Black rectangles around each character

Figure 13. Segmented characters

Character Recognition

The last step of the ANPR algorithm is the character recognition phase, which ought to be the most important since it compares the characters produced throughout the process and the characters saved in the database. This phase uses a template matching approach to compare each segmented character's result with the real character stored in the database. As shown in Figure 14, the templates include letters from A to Z and digits from 0 to 9 for 36 characters.

In the recognition stage, each segmented character is represented as a normalised binary matrix with fixed dimensions. This matrix acts as a digital encoding of the character shape, where pixel values are mapped into binary form (foreground and background). The recognition mechanism relies on comparing this encoded representation with a predefined set of template characters stored in the database. The matching process is performed using normalised cross-correlation, where each input character is evaluated against all templates, and the character with the highest similarity score is selected as the final output. This process can be interpreted as a deterministic encoding–matching mechanism rather than a learning-based approach.

For Arabic characters, encoding is more complex due to the presence of diacritics and connected structures. These features may lead to fragmented representations when using connected component analysis. Therefore, Arabic character encoding was not fully integrated into the current system. This limitation is acknowledged and will be addressed in future work by incorporating structure-aware segmentation and advanced feature representation techniques.

The primary strategy is using correlation and template matching techniques with a particular algorithm study.

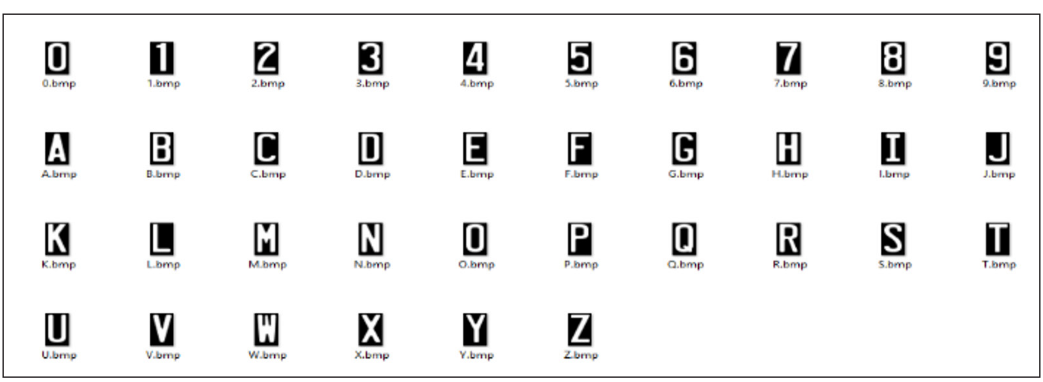

Figure 14. Template characters

The Cross-Correlation Process

In this experiment, the template matching process is the last step. To gauge how accurate the outcome is with the characters saved in the database for this ANPR research, the template matching approach is employed in this character recognition phase to compare the similarities and select the best match. The correlation algorithm is the optimum technique to utilise at this image processing stage. This ANPR research investigation uses two different correlation functions the cross-correlation and auto-correlation functions. The most effective method to gauge how similar two signals are is to use the cross-correlation function. Following segmentation, the characters are passed through a template matching procedure to be compared with the actual characters stored in the database. Subsequently, the result will demonstrate the precision and efficiency of the procedure. The input picture (segmented character) must be scaled to fit the template's dimensions before the matching process begins. The isolated character can then be recognised by comparing it to the characters in the template and determining which one has the most significant similarity. A 2-dimensional correlation approach is utilised to calculate the degree of similarity and identify the best match see Equation 1. The correlation between two characters that match precisely is equal to one.

$$R(X,Y) = \frac{\sum_{X'Y'}(T(X',Y').I(X+X',Y+Y'))}{\sqrt{\sum_{X',Y'}T(X',Y')^2 . \sum_{X',Y'}I(X+X',Y+Y')^2}} \qquad [1]$$

R represents the correlation between our dataset's template image and the original car image. *T* is the template character image. *I* represents the image cropping so that focus is only on the middle part of the image. *X* is the template binary-level image. x' is the average binary level in the template image. *Y* is the source image. y' is the average binary level in the source image.

Also, a threshold is employed only to return the best-matched characters that satisfy a correlation value > 0.6. This threshold was chosen from the observation that characters with a correlation value lower than this are not valid to begin with. They only passed the connected component filtering routine because they satisfied the width and height requirements. So even though they are considered characters, an empty character string will be returned when the threshold is not met to avoid outputting wrong results. Finally, the license plate number will be recognised and output on the screen shown in Figure 15.

RESULTS AND DISCUSSIONS

This section explains the hardware and software requirements, the ANPR system database, the application interface, detection results with limitations for Arabic license plates, the accuracy of the ANPR system, and recognition results.

Figure 15. Final recognition result for ANPR system

Hardware and Software Requirements

The hardware utilised for the implementation of this study consisted of a laptop:

RAM: 8GB Processor: Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz   2.90 GHz
HDD: 256 and above

The development process of the Automatic Plate Number Recognition (APNR) system will incorporate the specified software standards, along with the following essential prerequisites:

- Python 3.10 and Open-CV 4.6.0.66 as the programming languages;
- Operating System (OS): Windows 10 Pro 64-bit.

The ANPR System Database

The essential data were obtained through collaboration with the Karbala Traffic Directorate to acquire photographs of the registered vehicles, while additional images of Iraqi cars were taken from the Internet. The total number of automobile photos is 20. Various criteria, such as dimensions and configuration, hue, layout, and additional mechanisms utilising alternative methods, are also included in this data collection and will be discussed as shown in Figure 16.

During the data-collecting phase, a multitude of approaches are employed, incorporating the following:

- The study explicitly includes 20 Iraqi license plate photographs or images, as determined by specific criteria.
- The file format in which the image was saved or exported, either as a JPG or JPG file. JPG format photographs have superior format compatibility with future computer languages and systems, making them the optimal, widely adopted, and most easily disposable option.

The dataset was collected under partially controlled conditions to ensure a reasonable level of consistency. Most images were captured using a fixed camera position and approximately the same distance from the vehicles. However, complete uniformity was not strictly enforced, and variations in lighting conditions, plate scale, and slight orientation differences are present. To address these variations, preprocessing steps such as normalisation, resizing, and binarisation were applied to standardise the input before segmentation and recognition. In particular, all extracted characters were resized into fixed dimensions, ensuring consistency during template matching regardless of the original plate size or font scale.

It is important to note that although Iraqi license plates generally follow a standard font and layout, minor differences may still exist due to real-world capture conditions. These variations may affect recognition performance and are considered a limitation of the current dataset. Future work will focus on constructing a larger and more diverse dataset with stricter acquisition conditions.

ANPR System Interface

This section provides a quick explanation of the application's interface. This interface was primarily created using a Python open-source GUI program. Instead of experimenting by executing each algorithm separately, the interface is designed to make running through the GUI easier and more intuitive. This system's graphical user interface (GUI) is displayed in Figure 17.

Perfectly Centred Image

As stated during the preprocessing stage, if the input image taken is not exactly centred, the template matching algorithm will yield less accurate results or fail altogether as shown in Figure 18.

Figure 16. Examples of some Iraqi car images in the database

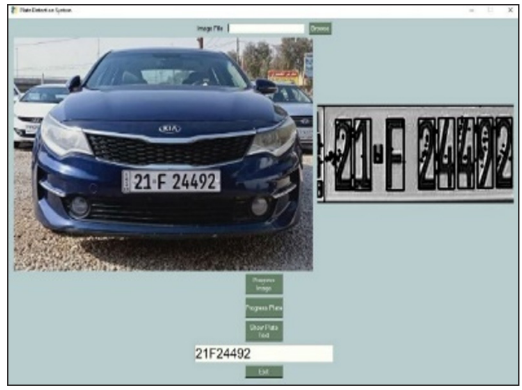

Figure 17. The graphical user interface for the ANPR system

Figure 18. Perfectly centred for Iraqi car images

Arabic Letter Detection

We experimented with recognising Arabic-Iraqi plates before designing the system around new English Iraqi plates. However, a few issues prevented us from adding Arabic letter detection support.

Detecting Letter Elements Containing Specific Symbols

The Connected Component Analysis (CCA) procedure will detect the letter and the dot under it as separate entities, such as detecting the dot (نقطة) under the letter (ب). The authors experimented with solving this issue by detecting the letter and then setting an arbitrary Height and Width bounding box around it instead of setting the Height and Width for the connected component pixel bounds themselves, thus detecting both the letter and its component as a single entity.

A challenge arises when diacritical elements, such as dots, are detected independently from their corresponding characters. For example, the dot associated with the letter (ب) may be identified as a separate connected component rather than part of the main character. In such cases, the system may incorrectly interpret the dot as an independent symbol, leading to additional false detections. One possible approach is to assume that a detected dot belongs to a nearby character. However, this assumption is not always reliable, especially if the primary character is not detected correctly or appears fragmented. Additionally, using fixed bounding regions to group components may unintentionally include parts of neighbouring characters, which reduces recognition accuracy. These challenges highlight the limitations of using basic connected component analysis for Arabic script, where characters often consist of multiple interconnected elements, as shown in Figure 19.

Figure 19. Arabic license plate connected component analysis result, showing the dot component of the (ب) letter being detected as a separate entity

Detecting the zero(صفر)

The zero character (number) in Arabic is smaller than the other characters, so much so that it doesn't pass our Width and Height thresholds for recognising a Connected Component as a valid character. The only way for it to pass the check is if the minimum Width and Height thresholds are set so low, but that also means some noise elements might pass as a valid zero (صفر) character. It has been tried to apply aggressive filters to remove even more noise elements, but that resulted in chopping off pixels from valid characters in the license plate, which caused the Template Matching algorithm to have less accurate results.

Arabic letters present additional complexity due to connected shapes and the presence of diacritics. CCA often detects dots as separate entities and fails to group them with their parent letters. Additionally, characters such as the Arabic zero (٠) frequently fall below the minimum size thresholds. These issues limit the system's ability to support Arabic recognition. Future implementations should adopt CNN-based segmentation, dot-association heuristics, and morphological grouping methods to reliably process Arabic script.

Significance of the Accuracy Results

The achieved accuracy of 91% is considered significant for a template-matching-based ANPR system operating on real Iraqi plates. In addition to the reported accuracy, the system produced limited false positives (FP) and false negatives (FN)cases during character recognition. The FPs mainly occurred when visually similar characters such as (0/O/D) and (B/8) were incorrectly matched. While the FNs occurred when segmented characters failed to satisfy the predefined correlation threshold and were therefore rejected by the recognition stage. These cases were mainly caused by character similarity, image noise, and illumination variations.

As shown in the example below, the correct output data must have been divided by 100 to obtain the precise percentage for each process step.

$$The\ Extraction\ Phase'\ s\ Correct\ Percentage\ (\%) = \frac{Number\ of\ Correct\ Extraction\ Output}{100\ (Total\ Input\ Data)}$$

$$The\ Extraction\ Phase'\ s\ Correct\ Percentage\ (\%) = \frac{Number\ of\ Correct\ Extraction\ Output}{100\ (Total\ Input\ Data)}$$

In this study, 20 Iraqi car images were employed to extract license plate text, compare it against actual plate text, and then calculate the average result refers to Table 1. We can notice characters' similarity in English; some characters could look like others, which would produce less accurate results from the Template Matching algorithm. Examples are the character "0" being detected as "D", the character "B" being detected as "8", and vice versa.

ANPR Recognition Results

Figure 20 displays recognition results using the suggested ANPR method based on photos of Iraqi cars in the database. Even if some license plates had negative characteristics, it is evident that all license plates were identified. As discussed in the next section, certain characters on license plates are challenging to detect and identify due to their similarities.

The proposed ANPR system operates automatically during all detection and recognition stages without requiring human intervention in the final output. Human involvement was limited to recording and validating the actual license plate numbers during dataset preparation to establish the reference values used for performance evaluation.

As mentioned before, the first method in the preprocessing stage is to save and record the actual or original captured plate number from the car image. The captured image is saved in the computer database, and every plate number of the car image will be recorded and saved in our database for future use.

Table 1
Recognition results

Plate Num.	Detected Text	Actual Text	Detection %
1.	21F24492	21F24492	100%
2	21G36271	21G36271	100%
3	22A85281	22A85281	100%
4	22*U*77*O*13	22D77013	75%
5	22K6291*2*	22K62917	87%
6	22G21234	22G21234	100%
7	22J58324	22J58324	100%
8	22G7*O*473	22G70473	87%
9	22J49*O*2*B*	22J49028	75%
10	22G21234	22G21234	100%
11	22G77442	22G77442	100%
12	22J79246	22J79246	100%
13	22G77442	22G77442	100%
14	22G7*O*473	22G70473	87%
15	22B*2228J*	22B77283	62%
16	22_427*3*2	22H42712	75%
17	21G29482	21G29482	100%
18	21J15443	21J15443	100%
19	21H12396	21H12396	100%
20	22B757*D*2	22B75702	87%
		Average: 91%	

Figure 20. ANPR system recognition results

Let's say the first captured plate number image 22H40937; the file for 22H40937.jpeg will be saved, and the data of the image, which is the plate number (22H40937), will be recorded and saved by using the manual method, either handwriting or typing, to be kept in a database that will be designed later in ".txt" format as shown in Figure 21.

The ANPR system does not adjust for the different levels attained. The plate region image itself is the root cause of it. To achieve ideal outcomes, all aspects and characteristics of the captured photos in the dataset must be appropriately included in a solid and perfect criterion. Different levels of accuracy will come from variations in the brightness, contrast, distance, and pixel value of the image. Since each image is taken by hand, it may appear challenging to standardise its properties. However, the varying degrees of outcomes will guide us to the subsequent section on suggestions and future endeavours.

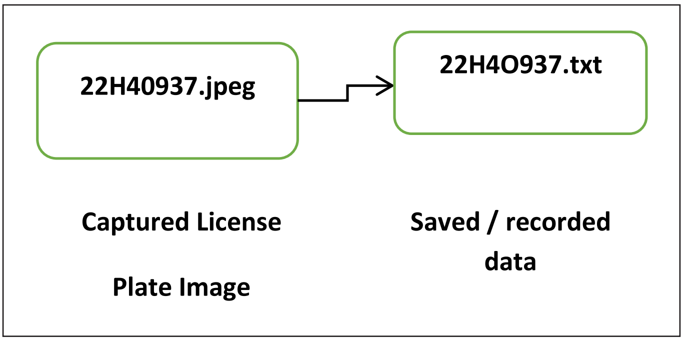

Figure 21. Plat umber data

CONCLUSION

The first goal of the study is to develop the Automatic Plate Number Recognition (APNR) system software. This goal was successfully met and applied. It involved developing the software to encompass all major and minor operations or processes, starting from the binarisation stage and ending with the character recognition stage. The second goal is to gather images of Iraqi license plates and use those numbers to gauge how accurate the APNR system is.

The accomplishment of these goals can be summed up as follows: 20 Iraqi vehicle images were captured using mobile cameras. The data used in this study were collected during every stage of the testing process in order to evaluate the precision with which the APNR program was implemented, utilising the template matching method. Consequently, all approaches for system stages are improved to obtain maximum accuracy when evaluated using the Iraqi car plate region, as the methods from earlier research and those used in system stages did not yield any favourable results using Iraqi vehicle plates.

The dataset used in this experiment was gathered by hand, and its correctness and efficacy were assessed using the most recent algorithm in the market. The ANPR system is an essential tool for improving the efficiency and safety of transportation networks in our day-to-day lives. It shows high potential to lessen the effort required to manage parking lots and monitor traffic. It may be inferred from the trials carried out during the last system phase that the use of the template matching algorithm in license plate recognition is not entirely accurate, since only 91% of the outcomes are correct. More research is required to enhance the current system, maximise outcomes and improve research across the board in the future. As a result, a few key recommendations can be put into practice that should thoroughly cover the algorithm at every stage of the study.

Hence, further investigation into improving the accuracy of character recognition results should be conducted. Some specific English characters, including 0, O, D, B, 8, 2, and 7, may sometimes lead to inaccurate detections. Errors occur throughout the recognition process due to the presence of similar characters. This could arise because of various factors affecting the image-capturing process, such as dirt, blurriness, capture angle, lighting conditions, and other similar concerns. Possible avenues for further research may involve enhancing the similarity between segmented characters and database ones.

Moreover, future work should focus on expanding the dataset and integrating deep-learning-based OCR models such as CNNs, CRNNs, and Transformer-based architectures. These models are expected to improve the recognition of visually similar characters and reduce segmentation errors. Furthermore, future developments should include dedicated Arabic character processing modules that address dot-association, adaptive thresholding, and morphological reconstruction to support both Arabic and English Iraqi plates with higher robustness.

ACKNOWLEDGEMENT

The authors thank the Karbala Traffic Directorate and Dr Husam Hasan Kadhim at the University of Kerbala for helping with proofreading.

Data Availability Statement: The written work involves data.

Conflicts of Interest: The authors assert that they have no conflicts of interest.

Funding: This research received no external funding.

REFERENCES

Abdulkadhim, E. G., Al-Shemarry, M. S., & Alsaadi, E. M. T. A. (2024). An efficient algorithm for covert contacting in IoT. *AIP Conference Proceedings, 3094*(1), Article 040008. https://doi.org/10.1063/5.0209934

Al-Shemarry, M. S., & Li, Y. (2020). Developing learning-based preprocessing methods for detecting complicated vehicle licence plates. *IEEE Access, 8*, 170951-170966. https://doi.org/10.1109/ACCESS.2020.3024625

Al-Shemarry, M. S., Li, Y., & Abdulla, S. (2023). Identifying license plates in distorted vehicle images: Detecting distorted vehicle licence plates using novel preprocessing methods with hybrid feature descriptors. *IEEE Intelligent Transportation Systems Magazine, 15*(2), 6-25. https://doi.org/10.1109/MITS.2022.3210226

Anagnostopoulos, C.-N. E. (2014). License plate recognition: A brief tutorial. *IEEE Intelligent Transportation Systems Magazine, 6*(1), 59-67. https://doi.org/10.1109/MITS.2013.2292652

Anagnostopoulos, C. N. E., Anagnostopoulos, I. E., Loumos, V., & Kayafas, E. (2006). A license plate-recognition algorithm for intelligent transportation system applications. *IEEE Transactions on Intelligent Transportation Systems, 7*(3), 377-392. https://doi.org/10.1109/TITS.2006.880641

Angelova, A., Krizhevsky, A., Vanhoucke, V., Ogale, A. S., & Ferguson, D. (2015). Real-time pedestrian detection with deep network cascades. In *Proceedings of the British Machine Vision Conference 2015* (pp. 32.1-32.12). BMVA Press. https://doi.org/10.5244/C.29.32

Arafat, M. Y., Khairuddin, A. S. M., Khairuddin, U., & Paramesran, R. (2019). Systematic review on vehicular licence plate recognition framework in intelligent transport systems. *IET Intelligent Transport Systems, 13*(5), 745-755. https://doi.org/10.1049/iet-its.2018.5151

Azam, S., & Gavrilova, M. (2017). License plate image patch filtering using HOG descriptor and bio-inspired optimisation. In *Proceedings of the Computer Graphics International Conference 2017* (Article 18). Association for Computing Machinery. https://doi.org/10.1145/3095140.3095141

Caplan, J. M., Kennedy, L. W., & Petrossian, G. (2011). Police-monitored CCTV cameras in Newark, NJ: A quasi-experimental test of crime deterrence. *Journal of Experimental Criminology, 7*(3), 255-274. https://doi.org/10.1007/s11292-011-9125-9

Castello, P., Coelho, C., Del Ninno, E., Ottaviani, E., & Zanini, M. (1999). Traffic monitoring in motorways by real-time number plate recognition. In *Proceedings of the 10th International Conference on Image Analysis and Processing* (pp. 1199-1203). IEEE. https://doi.org/10.1109/ICIAP.1999.797765

Duan, T. D., Duc, D. A., & Du, T. L. H. (2004). Combining Hough transform and contour algorithm for detecting vehicles' license plates. In *Proceedings of the 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing* (pp. 747-750). IEEE. https://doi.org/10.1109/ISIMP.2004.1434172

Lee, Y., Han, D. K., & Ko, H. (2013). Reinforced AdaBoost learning for object detection with local pattern representations. *The Scientific World Journal, 2013*, Article 153465. https://doi.org/10.1155/2013/153465

Littlewort, G., Bartlett, M. S., Fasel, I., Susskind, J., & Movellan, J. (2004). Dynamics of facial expression extracted automatically from video. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop* (pp. 80-80). IEEE. https://doi.org/10.1109/CVPR.2004.327

Ojeniyi, A., Aro, T., & Thiak, A. M. (2019). Design of an agent-based traffic control system. *International Journal of Engineering and Advanced Technology, 8*(6S3), 71-76. https://doi.org/10.35940/ijeat.F1012.0986S319

Panahi, R., & Gholampour, I. (2017). Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications. *IEEE Transactions on Intelligent Transportation Systems, 18*(4), 767-779. https://doi.org/10.1109/TITS.2016.2586520

Patel, C., Shah, D., & Patel, A. (2013). Automatic number plate recognition system (ANPR): A survey. *International Journal of Computer Applications, 69*(9), 38-42. https://doi.org/10.5120/11871-7665

Sarfraz, M. S., Shahzad, A., Elahi, M. A., Fraz, M., Zafar, I., & Edirisinghe, E. A. (2013). Real-time automatic license plate recognition for CCTV forensic applications. *Journal of Real-Time Image Processing, 8*(3), 285-295.https://doi.org/10.1007/s11554-011-0232-7

Song, M. K., & Sarker, M. M. K. (2014). Modelling and implementing two-stage AdaBoost for real-time vehicle license plate detection. *Journal of Applied Mathematics, 2014*, Article 697658.https://doi.org/10.1155/2014/697658

Wang, D., Tian, Y., Geng, W., Zhao, L., & Gong, C. (2020). LPR-Net: Recognising Chinese license plates in complex environments. *Pattern Recognition Letters, 130*, 148-156. https://doi.org/10.1016/j.patrec.2018.09.026

Yang, Y., Wang, Z. F., Yu, B. C., & Yin, Z. H. (2013). Vehicle license plate location based on high-frequency vertical edge and mathematical morphology. *Advanced Materials Research, 734-737*, 3252-3255. https://doi.org/10.4028/www.scientific.net/AMR.734-737.3252

Yousef, K. M. A., Al-Tabanjah, M., Hudaib, E., & Ikrai, M. (2015). SIFT based automatic number plate recognition. In *2015 6th International Conference on Information and Communication Systems (ICICS)* (pp. 124-129). IEEE. https://doi.org/10.1109/IACS.2015.7103214