

Research Article

UAV with Vision to Recognise Vehicle Number Plates

Naveed Ali Khan Kaimkhani ¹, Muhammad Noman,² Sabit Rahim ³,
and Hannan Bin Liaqat ⁴

¹INET Technical University Berlin, Berlin, Germany

²Electrical Engineering Federal Urdu University of Arts Science & Technology, Islamabad, Pakistan

³Computer Science Karakoram International University Gilgit, Gilgit, Pakistan

⁴University of Gujrat, Hafiz Hayat Campus, Gujrat, Pakistan

Correspondence should be addressed to Naveed Ali Khan Kaimkhani; naveedali64@yahoo.fr

Received 9 January 2022; Revised 5 August 2022; Accepted 3 September 2022; Published 30 October 2022

Academic Editor: Adarsh Kumar

Copyright © 2022 Naveed Ali Khan Kaimkhani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The trend of using Unmanned Aerial Vehicles (UAVs) in industries is rapidly growing. Nowadays, they are used in many applications, from surveillance to disaster assistance. Almost all applications of UAVs require cameras either to perform specific vision tasks such as facial recognition and vehicle number plate identification or to avoid obstacles in the flight path of the UAV. The most emerging application of UAVs today is to provide security and surveillance, and they are mostly used for vehicle number identification. Typical Automatic Number Plate Recognition (ANPR) uses static high-resolution cameras mounted in specific places to identify the vehicle's number plates. Identifying the characters on the number plate becomes a very crucial task when the number plate is at an arbitrary angle to the drone camera. The camera gimble angle, the height of the UAV from the vehicle, and the relative speed of the UAV w.r.t vehicle play a very key role in identifying the vehicle license plate correctly. This study explains how Automatic Number Plate Recognition (ANPR) is performed on real-time images using MATLAB with the help of a UAV to analyze the effect of the above-mentioned key factors. The process is completed in three steps: collecting visual data from the drone, processing that data, and obtaining the recognized number plate.

1. Introduction

Unmanned aerial vehicle (UAV) surveillance has been a hot research topic for the last four decades. Research involving UAVs has drastically increased in industries and academics over the years. Due to the wide range of applications [1], UAVs are being implemented in both military and commercial projects. UAVs are unmanned vehicles that have an onboard electronic flight controller that controls them using the received signal from the base station. A base station is a point from where the UAV is being controlled or where it is responding to data. From using the UAVs as firefighters [2], agricultural pesticide sprayers [3], and photographers [4], the usage expands to "UAV as a courier" as Amazon has announced to use them for parcel delivery to their customers.

Security is the main issue when it comes to closed private organizations, and unregistered vehicles are the main threat

to the organizations. So, the vehicles need to be verified automatically as some intruders may falsely declare themselves as members of that particular organization and can enter inside.

With extraordinary research work in number plate recognition algorithms in the last two decades, many projects were implemented by using static cameras for number plate identification, which has the limitations of mobility. Those types of implementations become very inefficient when it comes to the congested parking lots, as the camera cannot focus on the vehicle number plates at the defined angle. There are systems developed that have up to 98% accuracy; however, none of these examples used drones for identifying the number plates. The objective of this research paper is to introduce a UAV-based vehicle number plate verification system which removes the limitations of mobility and static camera surveillance. Moreover, an intelligent algorithm for image enhancement and

processing has been introduced. Sections 3 and 4 explain the computation cost and efficiency of the introduced algorithm.

The main motivation for this research paper comes from the project that was implemented at Barry University [5] for the parking lot management system. They used a DJI Phantom 3 Professional drone to take pictures of the number plates of the vehicles parked in the university parking lot. Whereas, the drone was given a specific path for flight to take pictures, and then, those pictures were processed further to identify number plates. However, they used a fixed camera at a specific angle, and the required angle was manually calculated for each parking lot, which was one of the limitations of the project. The second limitation of the project was that the process was not real time. The drone first completed its flight, and then, the visual data were copied from the drone to the PC for processing. Also, the project was implemented considering that the vehicles were parked in a specific manner and the number plates were facing the camera.

In [6], a multitarget tracking system was introduced using UAVs, which considers that multiple UAVs are flying in a specified area and communicate intelligently to offload the data. This proposed work is, however, implemented in a closed parking lot and works based on tracking license plates as opposed to tracking the vehicles, which would be very inefficient in closed parking lots.

The UAV used in the project is self-designed as it provides ease in determining the angle and manual control. Although there are commercially available drones that are widely used in surveillance applications such as DJI drones, which also have Android/iOS applications to control them [5], however, we preferred using the self-designed UAV with a very high-resolution camera and digital video transceivers as it will affect the overall system efficiency by reducing the noise to a large extent. This study will discuss how the angle of the camera, the altitude, and the speed of the drone affects the efficiency of the system when identifying the number plates of the vehicles.

The study is organized as follows. Section 2 discusses the available solutions and their limitations. Section 3 gives an overview of the research work and explains the technology and methods used to identify the number plate from the images obtained from the drone. Section 4 provides insight into the results obtained and system efficiency dependency on various factors, and finally, in Section 5, we discussed the difficulties encountered in our project and the future work that can be performed based on the results that we obtained.

2. Related Techniques

The efficiency obtained from static cameras used for vehicle number plate identification is much higher. Authors of [7] used OPENCV to identify the number plates. The algorithm based on C++ language was applied to the static images that were first captured with the camera. The number plate was separated from the background using the Sobel filters and then by the threshold of the image to remove the noise. Finally, morphological processing was performed to isolate the characters from the number plate.

Mathematical morphology based on nonlinear neighbourhood operations was implemented in [8]. The license

plate is located using the morphological operations, and the characters were recognized using the Hausdorff distance. This method is also explained in [9–11] along with other commonly used methods.

In [12], authors applied number plate recognition algorithms to the data set of dirty Persian number plates. The method used was not real time and was applied to pre-captured images. The system was designed for the recognition of plates under harsh weather with varying sizes of number plates, and it ran with 97% accuracy on the English number plates. A similar approach was discussed in [13] for identifying the number plates with higher accuracy.

Several algorithms for number plate detection were proposed in [14]. Canny-edge detection gave the most efficient results of the other proposed methods. It consists of six steps and is very simple, clear, and accurate. First, the image is loaded and resized for further processing. Then, the image is converted into grayscale and then complemented to find the edges. Then, the filters were applied to separate the plate from other objects, and finally, the numbers were recognized.

A method for high accuracy for number plate was introduced in [15]; it uses a combination of Hough transform and contour algorithms. Close boundary algorithms were used to detect objects moving at faster speeds. The images were taken at different angles from the static camera. This technique performed well, and the authors were able to achieve the recognition efficiency of 99%.

Automatic locating of the number plates of vehicles by Principal Visual Word (PVW) is discussed in [16]. They used the model of bag-of-words (BoW), which is widely used in partial-duplicate image searches. Three main stages were used in this approach, which include PVW generation, visual word matching, and number plate locating. The method becomes unsuccessful when the resolution of the number plate is very low. This problem was addressed in [17], where the authors used low-quality videos to identify the number plates.

In [18], the authors used the gray level morphology approach, and in [19], they identified the number plates from the images that were taken from the distant camera. The algorithm they used included four stages starting, with the edge-detection picking the characters. Then, the characters were isolated from the image using the character extraction, which had an efficiency rate of 95%, and the last step was to localize the license plate, which was running at an efficiency rate of 81%. The efficiency varied under different weather conditions.

A real-time vehicle management system was introduced for verifying the vehicle number using the ground patrolling vehicle having a security camera on the roof [20]. It uses two cameras, one for vehicle tracking and the other for capturing the image of the license plate of the vehicle that was being tracked. The vehicle was tracked using the condensation algorithm, and to make this algorithm more effective, they used the “Self-Organizing Map” (SOM) to build a discrete vehicle shape model. When the first camera detects that the vehicle has reached the designated target line, it sends a signal to the second

camera to capture the image of the license plate of the vehicle. That image was then fed to the license plate identification algorithm to recognize the characters of the number plate by effectively accomplishing the character segmentation. The segmented characters were then identified by using the SVM (space vector machine) with 99% accuracy. Furthermore, the authors also got high-efficiency rates using SVM in their proposed models [21, 22]. The only limitation arises when the vehicle to be tracked was in heavy traffic. A similar approach was used in [23, 24] with slightly tough weather conditions and a complex data set, but the efficiency was slightly low, around 90%.

In [25], license plate identification was discussed mainly based on morphological operations. After converting the RGB image to a gray-scale image, opening (erosion followed by dilation) and closing (dilation followed by erosion) were performed to detect the edges and characters in the image. The data set consists of 120 images, which were recognized at an efficiency rate of 90.8%.

In [26], they used drones for traffic surveillance, and a method to automate the flying path of UAV was proposed to overcome the problem of UAV maximum flight distance using an algorithm based on the Pareto optimality technique. In [27], drones were used in the smart city to provide civil security by continuously monitoring the area. In [28], they performed an ANPR on license plates with variable sizes and having Latin scripts. They used a dataset of 50 different images captured from the fixed angle and attained 90% efficiency.

3. System Mathematical Modelling

The image $I(x, y)$ is the input image to the system, where “ x ” and “ y ” are horizontal and vertical co-ordinates of each pixel.

The pixel intensity of the given image is $f(p)$, given as

$$f(p) = I(x, y), \quad (1)$$

where $p = [x, y]$.

The kernel used for performing median filtering and for convolution for edge detection is a square matrix of 3×3 given as

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

3.1. Thresholding. The gray-scale image is converted into the monochrome black and white (binary image) by comparing it with the experimentally calculated global thresholding value.

Suppose “ T ” is the threshold value of the image; then, considering “ z ” as the original pixel and $z(T)$ as the intensity of pixel “ z ”, the binary value of pixel “ z ” is calculated as

$$z' = \begin{cases} 0, & \text{if } z(T) < T, \\ 1, & \text{if } z(T) > T. \end{cases} \quad (3)$$

Global thresholding value is calculated by running the following algorithm until desirable results are achieved:

- (1) Select initial estimate of threshold value “ T ”.
- (2) The image is now divided into two sets:

$$\begin{aligned} A &= \{[x, y] | I(x, y) < t\}, \\ B &= \{[x, y] | I(x, y) \geq t\}. \end{aligned} \quad (4)$$

- (3) Average gray level calculation Φ_A and Φ_B :

$$\begin{aligned} \Phi_A &= \frac{1}{|A|} \sum_{[x,y] \in A} I(x, y), \\ \Phi_B &= \frac{1}{|B|} \sum_{[x,y] \in B} I(x, y). \end{aligned} \quad (5)$$

- (4) Determine a new threshold value:

$$T = \frac{1}{2} (\Phi_A + \Phi_B). \quad (6)$$

- (5) These steps are repeated until “ ΔT ” is smaller than the initial guess T .

3.2. Edge Detection

$$\begin{aligned} I'(x, y) &= I(x, y) * M(x, y) \\ &= \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} I(x, y) \cdot M[\text{mod}_w(x-i), \text{mod}_h(y-j)], \end{aligned} \quad (7)$$

where “ W ” and “ h ” are the dimensions of the image I .

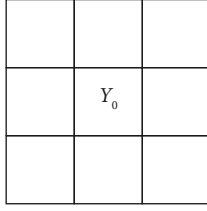
Convolution defines how a pixel is affected by its neighbour pixels. A single pixel is surrounded by 8-neighbour pixels.

x_1	x_2	x_3
x_4	x_0	x_5
x_6	x_7	x_8

When convoluted with a kernel of 3×3 of the following arrangement:

m_1	m_2	m_3
m_4	m_0	m_5
m_6	m_7	m_8

A destination image has a single effected pixel at location y_0 .



The pixel represented by y_0 in the destination image is affected by the pixels from $x_0 \dots x_8$. According to the above formula,

$$y_0 = x_0 * m_0 + x_1 * m_1 + x_2 * m_2 + x_3 * m_3 + x_4 * m_4 + x_5 * m_5 + x_6 * m_6 + x_7 * m_7 + x_8 * m_8. \quad (8)$$

3.3. Histogram Normalization. A gray-scale image undergoes histogram normalization to distribute its intensities without affecting the original image characteristics:

$$H(i) = \left| \left\{ [x, y] 0 \leq x < w \hat{0} \leq y < h \hat{I}(x, y) = i \right\} \right|. \quad (9)$$

From the above equation, we determined each gray level “ i ” number of occurrences, where $i \in 0 \dots G-1$. G is the total number of gray levels (0–255).

From the above equation, we can determine the minimum. Maximum and the average values are as

$$\begin{aligned} H_{\min} &= \min \{I(x, y)\}, \\ H_{\max} &= \max \{I(x, y)\}, \end{aligned} \quad (10)$$

where $0 \leq x < w$, $0 \leq y < h$,

$$H_{\text{avg}} = \frac{1}{w * h} \sum_{x=0}^{w-1} \cdot \sum_{y=0}^{h-1} .I(x, y). \quad (11)$$

The histogram distributes the intestines and provides the number of bins. The bin with the most weight is chosen and is passed back to the ANPR algorithm for preparing the results for template matching.

4. Methodology

The self-designed UAV with a high-resolution SONY RX1R-II camera was used to collect the dataset of 300 different images of vehicles having standard and stylish license plates on them at different angles and altitudes. The license plates were captured at arbitrary angles from different altitudes and at different speeds using UAV to analyze the effect of these variables. UAV is continuously transmitting video back to the base station using video transceivers, through which we can avoid any obstacles in its path of flight. To capture the images at specific angles, heights, and speeds, we sent commands wirelessly from the base station to the onboard microcontroller wired to the camera. The camera, after receiving the commands, starts capturing the images at a

predefined speed. The incoming visual data are then stored on the PC to perform the license plate recognition algorithm on it. The following image provides an overview of the whole process. The UAV was controlled manually from the base station using the RC transmitter. The speed and height of the UAV are thus adjusted manually to collect the desired data. Figure 1 shows the steps followed from obtaining the raw data to extracting the targeted information from it.

The steps in Figure 1 are repeated for every number plate to be identified. After collecting a sufficient dataset of license plates, ANPR was applied to it. This algorithm is divided into two main steps: License Plate detection and License Plate recognition. The approaches and techniques are differed according to each country’s policy on number plate styles. The implementation of the algorithm is varied with the different situations. Each country around the globe has various license plate sizes and measurements. The dataset that we collected is particularly based on Pakistani license plates. However, when the same algorithm was applied to the online-available dataset of standard European and American number plates, the results had almost the same accuracy as the local number plates. The results of European and American standard plates are also discussed later in the study.

Pakistani license plates have different font styles on license plates, and it becomes very hard to identify the original characters. Furthermore, for capturing the images from the camera at specific angles, serial commands were sent through the Laird AC4790 wireless serial transmitter to the ATMEL microcontroller that was wired serially to the camera for zooming and capturing images. By sending these commands to the camera, several parameters can be controlled, like the camera shutter speed at which it captures the pictures and the encoding type of captured images. Those captured images were transmitted and were further processed on the base station. Each of these steps is explained below.

4.1. Collecting Visual Data from UAV. The UAV was controlled from the base station using an RC transmitter. It has an onboard flight controller as well as a camera that transmits video in real time to the base station. It has an ATMEL microcontroller on it that receives serial commands from the base station wirelessly using the Laird AC4790 module and then controls the camera zoom and also takes pictures at specific angles and transmits them to the base station. We prefer transmitting only specific video frames as images are less likely to be affected by noise as compared to video and occupy less space on the host computer image processing.

4.2. Image Processing. These HD images, captured from a drone camera (Figure 2), were sent back to the base station where a PC is used to process the image using MATLAB in real time. These images were fed to the MATLAB algorithm to locate the number plate and perform Optical Character Recognition (OCR) to identify the number plate of the

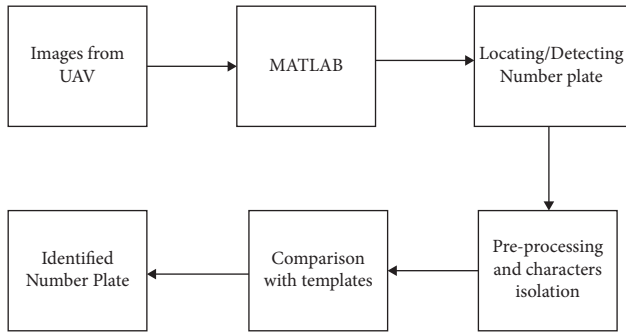


FIGURE 1: System overview.



FIGURE 2: Image captured using UAV.

vehicle. The algorithm to identify the number plate is divided into 3 steps. Each of them is explained below.

4.2.1. Removing Noise and Preprocessing. The image obtained from the camera is in RGB format. Processing the images in RGB requires a much more powerful CPU and GPU, which is not necessarily required. So, we need to convert that image into a grayscale image to perform digital filters and thresholds. The threshold is the process of converting the grayscale images into binary images. Figure 3 describes the steps involved in processing the image digitally. After converting the image to grayscale, we apply filters to remove noises. The most common type of noise found in the captured images is salt and pepper noise.

After removing the noise, the image is then converted to a binary image by thresholding it to perform shape-base processing. Thresholding provides a more reliable and efficient approach to converting a grayscale image into a binary image. Pixel intensity is distributed between 0 and 255, and a black or dark pixel has a value of 0 in MATLAB. For converting an image into a binary image, we divide the pixels into two categories with the help of a specific threshold value. The pixels with intensity levels greater than 127 are white and are marked as “1” and those with intensity

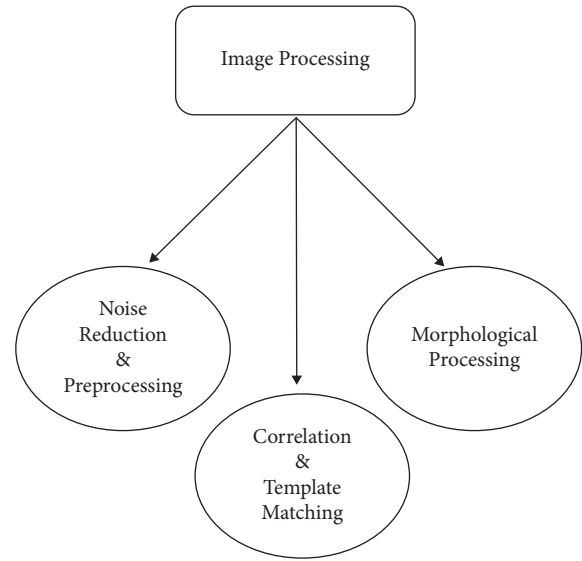


FIGURE 3: Image processing steps overview.

levels less than 127 are categorized into black pixels “0.” Now, as we have a binary image, we can perform binary operations such as morphological processing.

Before applying the morphological operations, we need to isolate the characters from the rest of the plate. Two of the most common, yet efficient, methods are used to do this job, i.e., convolution or median filtering. Figure 4 shows the results obtained when making the crucial choice between the convolution and median filters to isolate the number plate characters from the background. We would have used convolution to remove this noise, but it does not reserve the edges of characters on the number plate, leading to lower accuracy of the algorithm. This problem occurs not only on Pakistani number plates but also on American and European number plates, as it is clear from Figure 4. It can be noticed that when the edges of characters are away from adjacent characters or the edges of number plates, convolution and median filters provide almost the same results.

When the image is copied into the MATLAB workplace, it starts character extraction by using the ANPR algorithm. The same image is processed 5 times by an algorithm to calculate the confidence level. Every time “1” is added to the confidence level variable if the number matches the already detected number. This technique, however, increases the time to process each image but provides an accurate estimation of the efficiency of the algorithm in real time. After the image is processed, the algorithm provides 5 possible results for each license plate with different confidence levels, from 80% to 95%. The number with the maximum confidence level is picked up and stored.

The results of the above experiments (Figure 4) show extremely interesting results. Consider the first two images from the figure, where the characters are clear and there is an adequate space among them. The result of the convolution and median filters is almost identical. However, when considering the license plate UP14BN4001, the convolution kernel destroyed the information regarding the



FIGURE 4: Convolution and median filter results for western number plates.

TABLE 1: Results for UP14BN4001 using a median filter.

Plate	Confidence in percentage
UP14BN4001	93.019
UP14BN4001	90.490
UPI4DN400I	88.501
UPI4BN400I	88.004
UP14DN400I	86.207

alphanumeric numbers and cannot be read anymore by the template matching algorithm. Table 1 shows the results of the abovementioned number plate by using a median filter.

However, the results which are produced after the application of convolution to the above number plate number are useless and cannot even make a guess close to it, as it is obvious from Figure 4. The median filter provides excellent results in all of the three experiments above. The same conclusion was made when the former approach was used for the Pakistani number plates (Table 2) captured using the drone. The results for four different number plates using both techniques are discussed in detail below.

Whereas, the results of Table 2 based on the convolution method applied to the vehicle shown in Figure 2 captured by the UAV are useless, as the convolution brightened up the edges of each character, which were already close to each other. However, the use of the median filter, as clear from Figure 5 provided excellent results (shown in Table 3) and correctly guessed the characters with a confidence level of 96.3%.

Figure 5 shows a detailed comparison between convolution and median filters. It is clearly depicted that the median filter is the best fit in our scenario.

TABLE 2: Results for AHZ315 license plate using convolution.

Plate	Confidence in percentage
AHZ3K	95.512
A11236	91.388
AH236	89.154
AA23K	88.734
AH23S	85.945

TABLE 3: Results for AHZ315 license plate using a median filter.

Plate	Confidence in percentage
AHZ315	96.307
AH2315	93.110
AH231S	87.368
AHZ315	83.109
AHZ31S	81.057



FIGURE 5: Processed binary images.

Again, when the characters on number plates are close to each other in Figure 5, the edges of the characters start touching each other and the accuracy of the algorithm decreases. So, instead, a median filter is used to remove this

noise as it calculates the median of the intensity of the neighboring pixels, thus reserving the edges. Furthermore, the Kernel plays an important role while performing the filtering as it selects the neighbor pixels. The comparison between different kernel sizes is shown below. A kernel size of 3×3 is applied on the same plate and is compared with kernels of larger sizes. Figure 6 clearly shows the impact of kernel sizes on different plates.

During the above experiments (results shown in Figure 6), it was realized that a larger kernel size reduces the details of characters, and the algorithm performs very poorly on numbers that are identical to each other, such as “0” and “0” and “1” and “1”. Kernel size should neither be too big nor too small as it affects the edges of the characters. After detailed testing, a kernel size of 3×3 was chosen to perform median filtering, which produced quite good results.

4.2.2. Morphological Processing. To isolate the characters from the background, we need to perform dilation and erosion separately on the same image. Dilation increases the white pixels in the image while erosion increases the dark pixels depending upon the kernel used for morphological processing. Subtracting that eroded image from the dilated image, we get the edges of the characters present on the number plate. These are not only numbers but also other edges obtained that are not numbers, such as the edges of the number plates. Now, these garbage pixels should be removed from the image somehow. The best practice that is often used is to calculate the height of the bounding boxes and then find a histogram of those. Usually, the standard number plates have characters of the same height, so we separate those boxes by using their “x’s” and “y” coordinates in the image. The bin which has the most bounding boxes of the same height is chosen for template matching. Hence, the boxes that are not matched with other boxes of the same height are ignored. Figure 7 shows the results after applying the morphological processing to the image shown in Figure 2.

4.2.3. Template Matching. The characters obtained after the morphological operation are compared with the stored templates of the characters using correlation. The correlation factor will be high (close to 1) if a character is closely matching the stored template (some examples shown in Figure 8). We pick the character from memory that has a greater correlation factor with the character on the number plate. We repeat this process for every character on the number plate and get a final array of the alphanumeric characters.

5. Result Analysis and Discussion

Now, as we have characters from number plates instead of images that were taken from the camera at different angles, we can perform data analysis. Each image took around 350 ms to process and to fetch the data from it. The system was implemented on both standard and stylish number plates. The experiment was performed on 300 vehicles at different angles by varying the height of the drone between

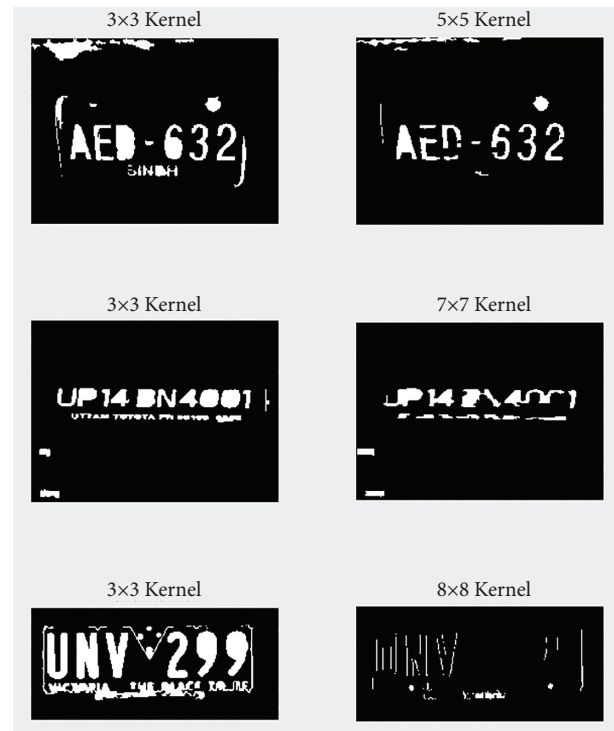


FIGURE 6: Comparison between different kernel sizes.



FIGURE 7: Morphological processing.

“0” and “8” meters. When the UAV is parallel to the horizon of the vehicle number plate, the efficiency is maximum, of 95.5%. When the height of the UAV increases, the efficiency starts to decrease (refer to Figure9, for obtained results). We increased the height by keeping the number plate in the center of the video frame and then setting the camera angle by using serial commands; the following graph was plotted based on the data collected.

With the increase in height of the drone, the camera angle increases, and this increase in angle is not linear though but is directly proportional to the height of the drone, as shown in Figure 10. However, the height of the drone is not the only factor that determines the angle; other factors such as the elevation of the road also affect the setting of the camera gimble angle. As the drone is controlled

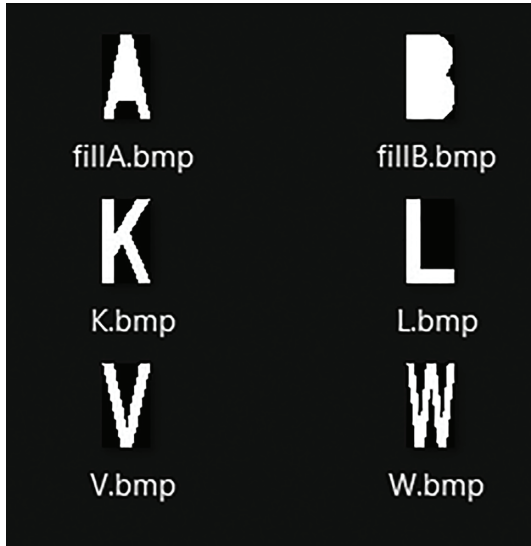


FIGURE 8: Stored templates of few characters.

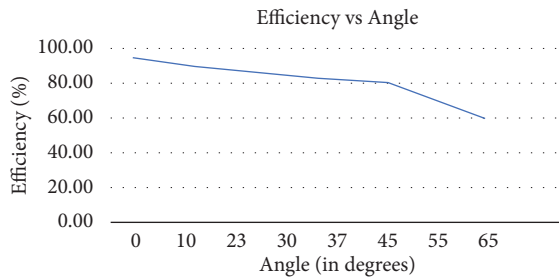


FIGURE 9: Efficiency vs. angle of drone w.r.t number plate.

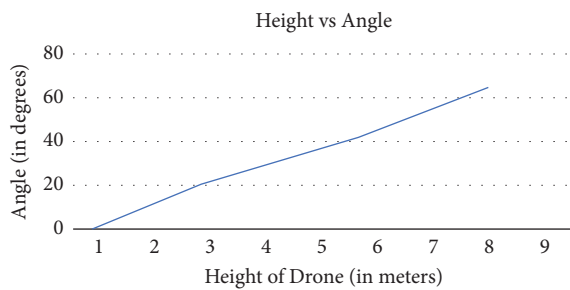


FIGURE 10: Height vs. angle of drone w.r.t number plate.

manually, the angle must be adjusted so that the number plate is in the middle of the frame. Figure 9 demonstrates the impact of the increase in angle on the overall efficiency of the system.

The third most important factor is the relative speed of the drone w.r.t vehicle. When the relative speed between the drone and the vehicle is nearly zero or very low, the efficiency of the algorithm is comparatively higher as the drone can capture static and clear pictures of the plate.

In Figure 11, the drone was moving at 20 km/h and the vehicle was standing stationary on the roadside. Even after removing the noise, the characters on plate “ABT290” cannot be identified by the algorithm as shown in Figure 11.



FIGURE 11: Processed image at a higher speed w.r.t vehicle.

So, by concluding the results obtained at different speeds, the correct recognition rate is highest when the relative speed between vehicle and drone is near “0” and is lower in other cases.

A similar problem arises with the nonstandard license plates and the plates’ having stylish font style characters on them. In most of the cases, they almost gave meaningless data. The application of the used algorithm on nonstandard number plates where the number plate characters were made from shiny metals/materials like aluminium where a few characters were brighter than others due to better reflection of sunlight and thus the accuracy of the results is very poor.

The shadow on some characters caused the numbers to fade away, and the algorithm could not identify those numbers. However, some digits were identified by the algorithm correctly.

Although the system performed pretty well on the standard number plates, still the efficiency went low due to the confusion between the similar templates such as “8” and “B,” “0” and “O,” “1” and “I,” and “5” and “S.”

6. Future Work and Challenges

The images were taken at the rate of two images per second, but this is not enough when we go to higher speeds, either from the drone or if the vehicle velocity is very high. One image was processed in around 350 ms by the MATLAB algorithm, which is quite slow when the obtained images are more than two frames per second. To increase the processing speed, a more powerful CPU is required or a GPU will be the best choice for this task. An onboard GPU can increase the processing speed and can increase efficiency. Furthermore, we can process the images on the drone and can send the characters directly to an android application directly where we can take the required decisions based on the collected data.

Moreover, an Android/IOS application such as Litchi can be used to automate the whole process and plan the missions of the drone. It will provide more control as compared to manual control because the flight mission can be programmed from the computer as well. The main task in performing the image processing and vehicle number identification is the detection of the characters on the number plate. Around 85% of the time is consumed in just detecting the characters, and the rest of the tasks, including template matching and correlation, are performed in just 15% of the total time that the algorithm took to process the

images. So, by using more efficient algorithms and using more powerful libraries such as OpenCV, we can dramatically reduce this time and can increase the efficiency reasonably.

7. Conclusion

The real-time implementation of Automatic Number Plate Recognition (ANPR) using the UAV is much faster and more effective as compared to verifying the vehicles manually. The proposed study removes the limitations from the initial research work, conducted at Barry University, where the drone was given a specific path of flight with a fixed camera whose angle of depression was calculated experimentally to take pictures of number plates. This implies that the required angle was manually calculated for each parking lot and that the flight path needed to be designed separately, which were the limitations of the research work. Furthermore, the process was not real time as the drone first had to complete the dedicated path and then visual data was copied from the onboard memory to process.

As discussed, the efficiency of the proposed system depends upon many factors, including the relative speed of the drone and the relative angle between the drone and the vehicle. All angles considered in the study are horizontal w.r.t the ground. The efficiency is higher when the vehicle is parked under better light conditions and when the relative speed of the drone concerning the vehicle is zero. This efficiency decreases drastically when the relative speed of the drone is high as compared to the drone, as the camera cannot focus on the characters and the algorithm can no longer differentiate the identical characters efficiently. The efficiency also greatly depends upon the brightness factor. While analysing the obtained data, those number plates that were directly facing the sunlight had very low efficiency as most of the characters had shadows on the other characters and had uneven light intensity.

Data Availability

The data used to support the findings of the study are included within the article in the Supplementary Information files.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] A. Hashmi, "A novel drone-based search and rescue system using bluetooth low energy technology," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7018–7022, 2021.
- [2] V. Spurny, V. Pritzl, V. Walter et al., "Autonomous fire-fighting inside buildings by an unmanned aerial vehicle," *IEEE Access*, vol. 9, Article ID 15872, 2021.
- [3] S. Khan, M. Tufail, M. T. Khan, Z. A. Khan, J. Iqbal, and A. Wasim, "Real-time recognition of spraying area for UAV sprayers using a deep learning approach," *PLoS One*, vol. 16, no. 4, Article ID e0249436, 2021.
- [4] D. Popescu, F. Stoican, G. Stamatescu, L. Ichim, and C. Dragana, "Advanced UAV-WSN system for intelligent monitoring in precision agriculture," *Sensors*, vol. 20, no. 3, p. 817, 2020.
- [5] J. Dasilva, R. Jimenez, R. Schiller, and S. Z. Gonzalez, "Unmanned aerial vehicle-based automobile license plate recognition system for institutional parking lots," *Systemics, Cybernetics and Informatics*, vol. 15, 2017.
- [6] P. Nathan, G. Frakgos, K. Ortiz, and M. Oishi, "A UAV-enabled dynamic multi-target tracking and sensing framework," in *Proceedings of the Global Communications Conference*, Taipei, Taiwan, December 2020.
- [7] B. Omman and A. Menon, "Detection and recognition of multiple license plate from still images," in *Proceedings of the 2018 International Conference on Circuits and Systems in Digital Enterprise Technology*, IEEE, Kottayam, India, December 2018.
- [8] I. Kamal, K. Housni, and Y. Hadi, "Online dictionary learning for car recognition using sparse coding and LARS," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 1, pp. 164–174, Yogyakarta, 2020.
- [9] W. Rahmaniar and A. E. Rakhmania, "Online digital image stabilization for an unmanned aerial vehicle (UAV)," *Journal of Robotics and Control (JRC)*, vol. 2, no. 4, pp. 234–239, 2021.
- [10] F. Faradji, A. H. Rezaie, and M. Ziaratban, "A morphological-based license plate location," in *Proceedings of the 2007 IEEE International Conference on Image Processing*, pp. I-57–I-60, San Antonio, TX, USA, October 2007.
- [11] W. Zhai, T. Gao, and J. Feng, "Research on pre-processing methods for license plate recognition," *International Journal of Computer Vision and Image Processing*, vol. 11, no. 1, pp. 47–79, 2021.
- [12] R. Panahi and I. Gholampour, "Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767–779, April 2017.
- [13] S. Hadavi, H. B. Rai, S. Verlinde, H. Huang, C. Macharis, and T. Guns, "Analyzing passenger and freight vehicle movements from automatic-Number plate recognition camera data," *Eur. Transp. Res. Rev.* vol. 12, no. 1, p. 37, 2020.
- [14] "Allam Mousa "Canny edge detection based vehicle plate recognition" International," *J. Signal Process, Image Process, Pattern Recognit.*, vol. 5, no. 3, 2012.
- [15] M. S. Al-Shemarry, Y. Li, and S. Abdulla, "An efficient texture descriptor for the detection of license plates from vehicle images in difficult conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 553–564, 2020.
- [16] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4269–4279, 2012.
- [17] T. Ajanthan, P. Kamalaruban, and R. Rodrigo, "Automatic number plate recognition in low quality videos," in *Proceedings of the IEEE 8th International Conference on Industrial and Information Systems*, ICIIS, Peradeniya, Sri Lanka, December 2013.
- [18] N. Al Said, Y. Gorbachev, and A. Avdeenko, "An unmanned aerial vehicles navigation system on the basis of pattern recognition applications—review of implementation options and prospects for development," *Software: Practice and Experience*, vol. 51, no. 7, pp. 1509–1517, 2021.
- [19] H. Anoual, S. Elfkhi, J. Abdellillah, and D. Aboutajdine, "Vehicel license plate detection in images," in *Proceedings of*

- the International Conference on Multimedia Computing and Systems*, Ouarzazate, Morocco, April 2011.
- [20] H. Lee, D. Kim, D. Kim, and S. Y. Bang, "Real-time automatic vehicle management system using vehicle tracking and car plate number identification," in *Proceedings of the ICME*, IEEE, Baltimore, MD, USA, July 2003.
 - [21] Z. Dong and X. Feng, "Research on license plate recognition algorithm based on support vector machine," *Journal of Multimedia*, vol. 9, no. 2, 2014.
 - [22] Y. Yang and Q. Zeng, "Multipoint Acceleration information acquisition of the impact experiments between coal gangue and the metal plate and coal gangue recognition based on SVM and serial splicing data," *Arabian Journal for Science and Engineering*, vol. 46, no. 3, pp. 2749–2768, 2021.
 - [23] J. Chang, S. Ryoo, and H. Lim, "Real-time vehicle tracking mechanism with license plate recognition from road images," *The Journal of Supercomputing*, vol. 65, 2011.
 - [24] M. Almutiry, "UAV tomographic synthetic aperture radar for landmine detection," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 5933–5939, 2020.
 - [25] A. Wang, X. Liu, Y. Han, and C. Qi, "License plate location algorithm based on edge detection and morphology," in *Proceedings of the 7th IFOST*, September 2017.
 - [26] X. Liu, Z. Guan, Y. Song, and nd Chen, "An optimization model of UAV route planning for road segment surveillance," *Journal of Central South University*, vol. 21, 2014.
 - [27] A. Giyenko and Y. I. Cho, "Intelligent unmanned aerial vehicle platform for smart cities," in *Proceedings of the 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS)*, pp. 729–733, Sapporo, Japan, August 2016.
 - [28] S. A. Haider and K. Khurshid, "An implementable system for detection and recognition of license plates in Pakistan," in *Proceedings of the 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies*, pp. 1–5, ICIEECT, Karachi, Pakistan, April 2017.