www.dergipark.gov.tr
ISSN:2148-3736

El-Cezerî Journal of Science and Engineering
Vol: 10, No: 1, 2023 (121-136)
DOI : 10.31202/ecjse.1172426

ECJSE

Research Paper

License Plate Recognition System Based on Artificial Intelligence with Different Approach

Aslı GÖDE[a], Ahmet DOĞAN[b]

[a]Management Information Systems Department, Burdur Mehmet Akif Ersoy University, Burdur, Turkiye
agode@mehmetakif.edu.tr
[b]Management Information Systems Department, Osmaniye Korkut Ata University, Osmaniye, Turkiye
ahmetdogan@osmaniye.edu.tr

Received: 08.09.2022 **Accepted:** 24.01.2023

Abstract: A license plate recognition system (LPRS) generally provides control and security. These systems are created using methods such as artificial intelligence, machine learning, artificial neural networks (ANN), deep learning, fuzzy logic, expert systems, and image processing. This study aims to create an LPRS using artificial intelligence and image processing techniques. The prepared system is for rectangular-sized plates. An LPRS consists of 3 main stages. The first stage is to detect the plate region. At this stage, converting to grayscale, bilateral filtering, canny filtering, and contour were applied to vehicle images. The second stage is to crop the plate region. In the second stage, the masking method was employed. Finally, the pytesseract algorithm was used to recognize license plate characters in the last stage. To create the system, Raspberry Pi 4 Single-Board Computer (SBC) was used for hardware; python programming language was utilized for software. The results showed that the system worked successfully at 100% in the first two stages and 91.82% in the last stage. The results suggest that the system works successfully.

Keywords: License Plate Recognition, Raspberry Pi, Filtering, Contours, Optical Character Recognition.

1. Introduction

Industry 4.0 is defined as the integration of technologies (i.e., Artificial Intelligence, Cyber Security, Big Data, Autonomous Robots, Augmented Reality, Internet of Things, Cloud Services, Augmented Reality, and 3D-Printing), allowing the conversion of how organizations operate along with significant changes in business models and manufacturing processes [1]. Artificial intelligence is considered one of the three greatest events in history [2]. Artificial intelligence is what human intelligence can do: It comes to the forefront compared to other technological developments in terms of bringing the skills of reasoning, meaning-making, decision-making, learning, and producing solutions to a computer or machine [3]. The usage areas of artificial intelligence are expanding, and research in this area is increasing. Artificial intelligence is used in medical [4], security [5], entertainment [6], e-commerce [7], tourism [8], education [9], banking [10], and many other areas. In addition, artificial intelligence is used in traffic control [11]. Artificial intelligence applications in this field have also become widespread due to the increase in vehicles and the need for advanced traffic control systems. Some of these applications used in traffic control are fuzzy logic or expert systems (used for controlling traffic flows), genetic algorithms (used for traffic optimization), and ANN (used for traffic flow estimation) [12]. In addition to these applications, LPRS using artificial intelligence technologies is very important for traffic control. LPRSs' usage in traffic control includes [13], but is not limited to, sites [14], parking lots [15], highways [16], vehicle inspection stations [17], hospitals [18], public buildings [19], border gates [20], military zones [21], and universities [22]. The main purposes of the use of LPRS are to determine the position of vehicles passing a certain point, to identify the vehicle, to examine the vehicle's

How to cite this article
Göde, A,, Doğan, A., "*License Plate Recognition System Based on Artificial Intelligence with Different Approach*" El-Cezerî Journal of Science and Engineering, 2023, 10(1); 121-136.
ORCID ID: [a]0000-0001-7785-6200; [b]0000-0002-7116-3558.

behavior, and to provide traffic control [23]. An LPRS consists of three critical stages. The first is the process of detecting the location of the license plate area, the second is the process of character separating from the plate region, and the third is the process of recognizing license plate characters. An LPRS has benefits such as ensuring security at the borders of countries and cities, calculating vehicles' length in the parking lot and determining the fare, calculating the density of parking lots, identifying traffic violations and penalties, detecting lost or stolen vehicles, saving time and eliminating long lines of cars, reducing costs, increasing security, calculating the transportation time of vehicles between two points on highways, no need for additional personnel at the entrances and no need for remote control, input card, tag, etc. [24].

The previous studies' findings indicate that most studies used LPRS consisting of three stages: detecting the plate region, cropping it, and recognizing license plate characters. Our study consists of three stages, like previous studies. It is important to accentuate that there is not much difference between those studies' first two stages and ours in terms of the methods used. However, the method used in the third study, the current study, and others differ. More specifically, as other researchers did, we detected the plate region in the first stage and cropped the plate region stage in the second stage. However, as other researchers used artificial neural networks in the third stage, we used artificial intelligence in this stage. As a result, we differentiated from the existing literature with the artificial intelligence we applied at this stage, which is our study's most critical motivation component. For example, we made it possible to recognize all license plates in different languages worldwide.

1.1. Literature Review

The reviewed articles suggest that most LPRSs are systems consisting of 3 stages. Only two studies were found consisting of a single stage. While one of them was successful, the other was unsuccessful. In most of the studies in the literature, image processing techniques (filtering, morphology, gray level, etc.) were used in the first two stages (finding plate regions and separating plate characters). Studies using only artificial neural networks used image processing techniques in the first stage and artificial neural networks in the second and third stages to train the system. In the majority of the studies in the literature, the canny filter is used in the image processing parts. Besides the canny filter, a bilateral filter is used in this article. The most crucial difference between the study and existing literature is that artificial intelligence was used in the third stage (recognizing license plate characters). Thanks to the pytesseract algorithm containing both language and character parameters, time is saved without the need to train letters and numbers. At the same time, it has become possible to read license plates belonging to all countries, not a single country. In the existing literature, studies take place recognizing license plates of a single country. In the article, we improved this handicap by using the artificial intelligence algorithm. Some studies in the related literature are as follows.

Singh and Singh [25] developed LPRS using machine learning algorithms and Python programming language in India, and their system worked at 96% accuracy. Guan et al. [26] developed an LPRS based on capsule networks in their study. This study showed that LPRS has high accuracy. Rashtehroudi et al. [27] developed a system to recognize Iranian license plates using deep learning techniques and Python programming. They used 1000 images taken from a security camera for a dataset. The accuracy rates were 90.2% for neural networks and 87% for support vector machines. These results indicated that the pytesseract algorithm increased LPRS performance by 99.2%. Li et al. [28] designed a system that recognizes Chinese license plates. According to this study's results, LPRS using median filters provided both more savings and more accurate results. Ohzeki et al. [29] developed an LPRS with machine learning without using number recognition in their study. They used three different methods in their systems. These are the CT5L method, machine learning, and the Otsu algorithm. Zhang et al. [30] developed an LPRS that recognizes Chinese license plates using machine learning. 172 plate images were tested;100% accuracy was achieved. Cheng et al. [31] developed a system that recognizes Chinese license plates using the Python programming. The accuracy rates were 99% detection of the license plate region stage and 97.1% recognition of the license plate character stage in that study. In general, the success rate of the system was found to be 95%. Lee et al. [32] designed an LPRS based on a convolutional neural network (CNN) from deep learning algorithms. Five hundred vehicle images were used for training neural networks. In LPRS, 100 vehicle images were tested, and the system was 97% successful. Selmi et al. [33] aimed to use the CNN deep learning method to detect the plate region. This study showed that the system accuracy rates were higher when compared with other studies. Dalida et al. [34] developed a system that recognizes Filipino vehicle and motorcycle license plates in their study. This study revealed that the system achieved 94.01% accuracy in the character recognition stage. Odone [35] created an LPRS using a support vector machine (SVM). The LPRS only worked on Italian car plates. The system found the operating accuracy to be 96%. Nagare [36] developed an LPRS using ANN in her study and found that the learning vector quantization neural network technique gave more accurate results than the back-propagation neural network technique. Oz et al. [37] used ANN in their study. Multilayer network was used as network model, and extended delta rule was used as learning rule. The network model they prepared consisted of 209 inputs, 36 outputs, and 120 hidden neurons. The results indicated that LPRS achieved results up to 100%.

2. Background

An image can be defined as a 2D function f(x,y). Here x and y are the plane coordinates. The amplitude of the f function in any (x,y) coordinate is called the gray level or intensity of the image at that point. If the intensity values of x, y, and f are all finite and discrete in size, that image is called a digital image. The digital intensity processing area is defined as processing digital images through a digital computer [38]. A digital image consists of a finite number of components; each component has a value. These components are called pels, pixels, picture elements, or image elements. It is the most used pixel expression [39]. There are three types of computer image processing processes. The first low-level processes include simple operations such as image input and output, image settings (such as brightness, contrast, and color), image filtering, and image dilation. Middle-level processes include segmentation that divides an image into areas or objects, image statistics, and binary operations. High-level processes include image classification, image clustering, and image making [40]. Thus, image processing is defined as extracting the features that enable the objects in these images, whose input and output are images, to be recognized [38]. Digital images are collected in 4 basic groups: binary, gray-level, color, and special. Binary images consist of only black and white colors. Pixels in the image take only two values, 0 and 1. A grayscale image includes gray shades in addition to black and white. In a gray-level image, the pixels take a value between 0 and 255. 0 represents the lightest white color and 255 is the darkest black color. The color image is displayed as 24-bit data on computer screens. It occurs when three gray-level images of the same object coded as R (Red), G (Green), and B (Blue) are displayed on top of each other. If the images cannot be represented by one of the first three standard image groups, it is a striking image [41]. The image processing technology is primarily used in fields such as artistic effects, biomedical, industrial inspection, geographic information systems, law enforcement, human-computer interface, face recognition, and movement recognition [42]. Figure 1 shows the basic architecture of digital image processing.

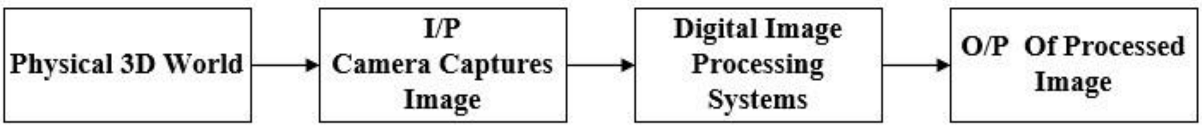


Figure 1. The basic architecture of digital image processing [43]

2.1. Image Preprocessing

The first stage of image processing is image preprocessing. Images obtained through cameras and different devices cannot be analyzed directly because they contain adverse effects such as noise and light deficiency. Therefore, filters are applied to improve the source image and eliminate these drawbacks. The preprocessing stage includes techniques such as color space transitions, scaling, histogram operations, thresholding, filtering, and morphological transformation [44]. Figure 2 shows the basic steps applied in image preprocessing.

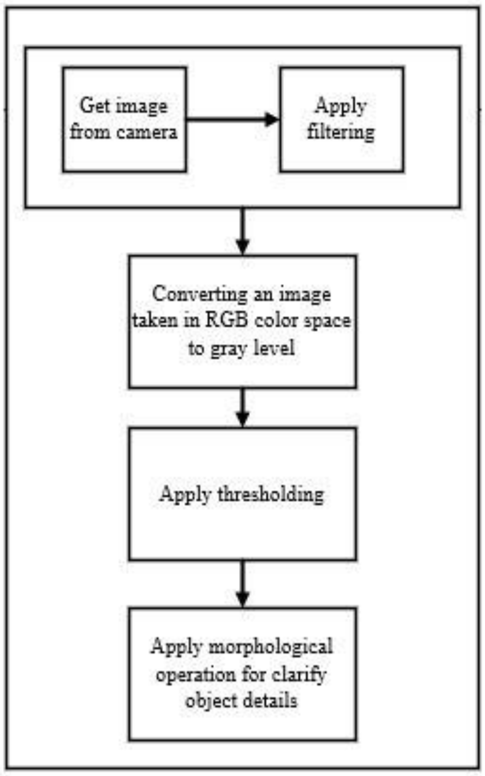


Figure 2. The basic steps are applied in image preprocessing [44]

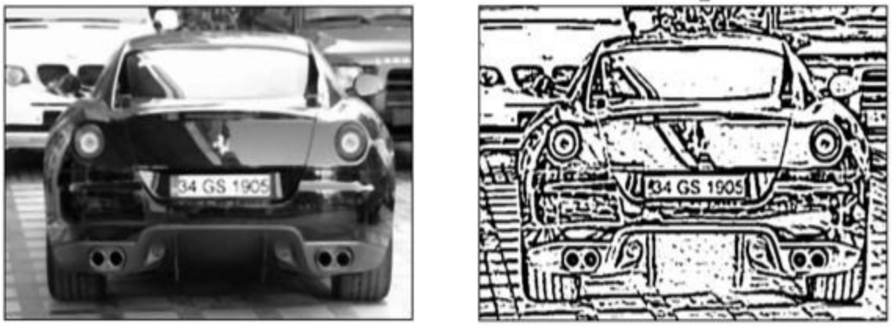
2.1.1. Basic filters used in image preprocessing

Filtering operations are defined as changing or enhancing an image to remove certain features or highlight interesting features in an image [43]. The primary purpose of filtering is to clarify the distinction between the physical features in the image and make it easier to comment on the image [44]. Mean, gaussian, sobel, laplacian, canny, and bilateral filters, which are among the basic filtering methods, have been tested in the pycharm program using the python programming language. The outputs obtained are explained with images.

Mean filtering: Mean filter is a type of linear filter and is used to remove noise. First, a kernel (x, y) is created on the image to clear the noise. With the kernel, the arithmetic, harmonic or geometric average of the pixels remaining in the template is taken. Thus, a new pixel value is obtained. Then, the resulting pixel value is replaced with the value in the center of the kernel. In this case, this process is repeated for all pixels in the image, and the image is cleared of noise. The mathematical formula of the arithmetic mean filter is given in Equation 1 [44].

$$f(x, y) = \frac{1}{mn} \sum (s, t) \in S_{xy} \, g\,(s, t) \quad (1)$$

The mean filtered result image is shown in Figure 3.



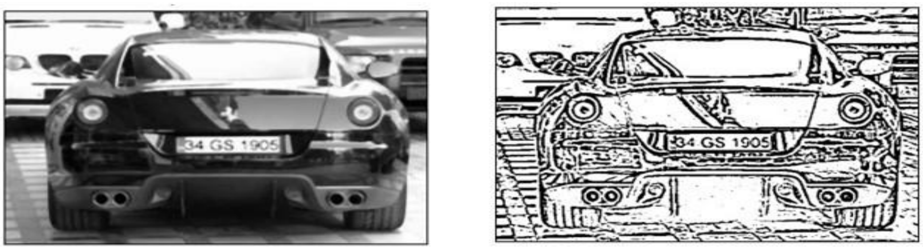
(a) Original image (b) Mean filtered image

Figure 3. (a) Original license plate image with gray level and (b) mean filtered image result

Gaussian filtering: Gaussian filter is a two-dimensional (2D) blur filter applied to smoothen the image and remove noise. Applying a gaussian filter is practical and efficient [45]. The mathematical formula of the two-dimensional gaussian filter is shown in Equation 2 [46].

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2)} \quad (2)$$

In this equation, G (x,y) represents the gaussian filtering value, σ (sigma) is the standard deviation of the gaussian function, and x,y is the cartesian coordinates. The image smoothing effect is higher if sigma has a significant value [47]. Gaussian filtered image is shown in Figure 4.



(a) Original image (b) Gaussian filtered image

Figure 4. (a) Original license plate image with gray level and (b) gaussian filtered image result

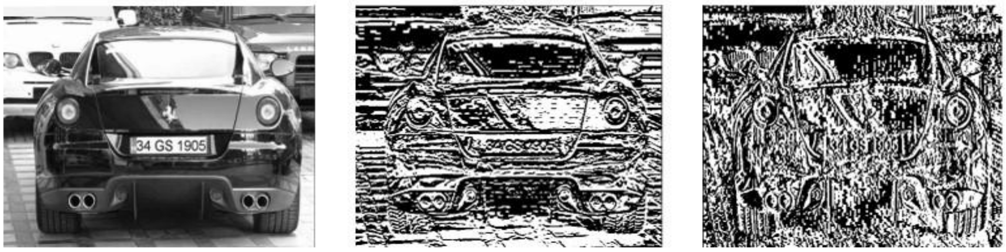
Sobel filtering: The Sobel filtering method determines the borders in the image. It is one of the most common methods used to detect edge regions. Using the sobel filtering method, horizontal and vertical edge detection can be done on the image [44]. While applying the sobel filtering method creates a horizontal and vertical 3x3 matrix of the input image [48]. Sobel kernel matrices with G_x vertical and G_y horizontal are shown in Equation 3.

$$G_x = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \qquad G_y = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{matrix} \quad (3)$$

The gradient size of the sobel filter method can be calculated with the formula shown in Equation 4 [48].

$$G = \sqrt{|Gx|^2 + |Gy|^2} \quad (4)$$

Sobel filtered image is shown in Figure 5.



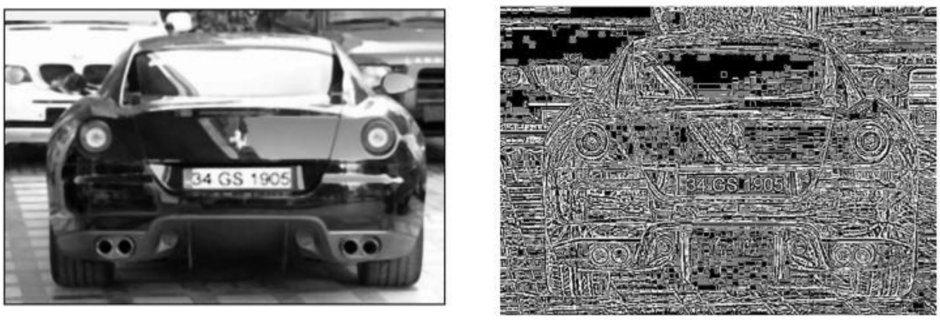
(a) Original image (b) X-axis image with Sobel filter (c) Y-axis image with Sobel filter

Figure 5. (a) Original license plate image with gray level and image result with sobel filter applied to (b) x and (c) y axis

Laplacian filtering: Laplacian filter is used to detect edges, highlight edges, and sharpen edges. It is also a frequently used filter for image enhancement and face recognition. In the laplacian filter, the edge is determined by taking the second-order derivative. In two-dimensional images, derivatives are taken for both the x and y axes [49]. The mathematical formula of the laplacian filter is shown in Equation 5 [50].

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{5}$$

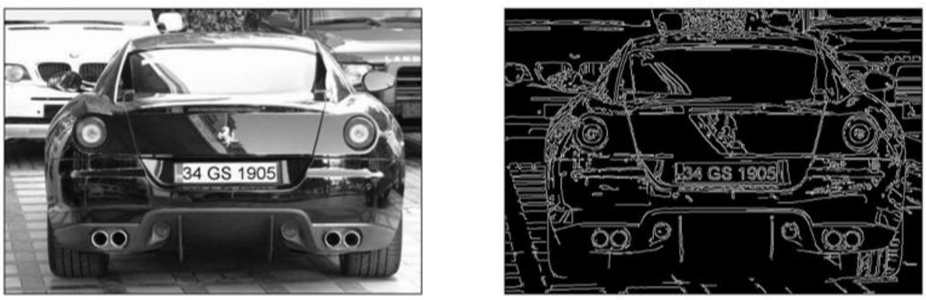
The formula shows that the laplacian filter is obtained by adding the x and y derivatives. The expression ∇^2expresses the value of the laplacian filter [49]. The laplacian filtered image is shown in Figure 6.



(a) Original image (b) Laplacian filtered image

Figure 6. (a) Original license plate image with gray level and (b) laplacian filtered image result.

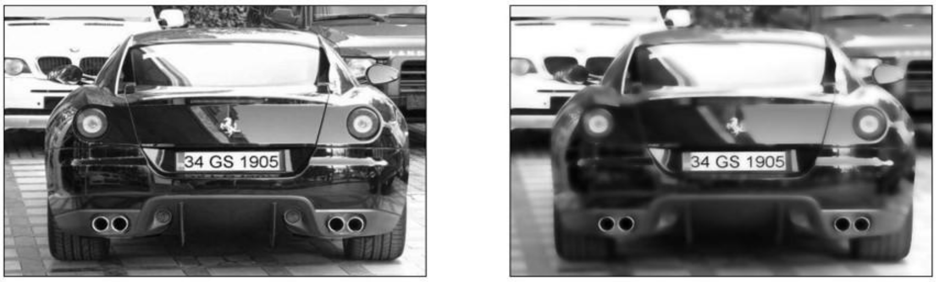
Canny filtering: Canny filtering method finds sharp and clearly defined edges in the image. The canny filtering method takes place in 5 steps. Firstly, gaussian filter is applied to soften the image. Secondly, edge width and direction are obtained with differential filters. Thirdly, non-maximum values are removed to obtain candidate edge points. Fourthly, candidate edge points are checked by double thresholding, and fifthly, the candidate edge points connected to the edge points are combined, and the result is obtained [51]. The canny filtered image is shown in Figure 7.



(a) Original image (b) Canny filtered image

Figure 7. (a) Original license plate image with gray level and (b) canny filtered image result

Bilateral filtering: Bilateral filtering reduces noise, preserves edges, and softens the image. In addition, bilateral filtering avoids mixing dense pixels on the image and protects sharp edges. Therefore, bilateral filtering is a non-linear filtering method [52]. A bilateral filtered image is shown in Figure 8.



(a) Original image (b) Bilateral filtered image

Figure 8. (a) Original license plate image with gray level and (b) bilateral filtered image result

2.1.2. Thresholding operations

Thresholding operations are generally used to convert gray-level images to binary images. Thresholding operations are divided into two groups static and dynamic thresholding. In static thresholding, a threshold value is determined by the user. The algorithm determines the threshold value in dynamic thresholding [44]. Threshold operations were tried one by one using a python programming language in the pycharm program. Finally, the outputs obtained are explained with images.

***Static thresholding*:** Static thresholding is divided into five groups: binary type, binary, inverted type, zero types, zero invented type, and truncate type [44]. Figure 9 shows the results obtained for each type after applying a static thresholding filter to the plate image.

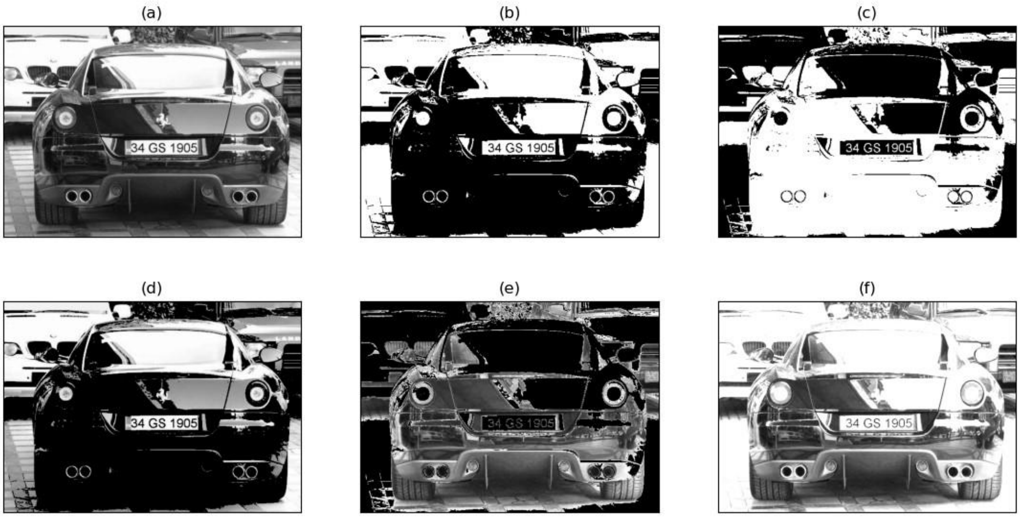


Figure 9. Original license plate image and images results applied static thresholding (a) Original image, (b) Image obtained as a result of binary type thresholding, (c) Image obtained as a result of inverted binary type thresholding, (d) Image obtained as a result of zero type thresholding, (e) Image obtained as a result of zero invented type thresholding image, (f) Image obtained as a result of truncate type thresholding

Otsu algorithm (Dynamic thresholding): The working logic of the grassy thresholding method is to divide the pixels in the image into two. Then, the separation of pixels in the image is provided according to the distribution of pixel values [53].

$$W_1 = \{0, 1, 2, \dots, T\} \quad (6)$$

$$W_2 = \{T+1, T+2, \dots, L-1\} \quad (7)$$

In Equations 6 and 7, W_1 and W_2 represent two separated classes, and L represents the gray-level pixel values. A mathematical formula used to calculate the percentage weight of each pixel is given in Equation 8 [54].

$$P_i = \frac{n_i}{N} \quad (8)$$

In this equation, i is the number of pixels, P_i is the percentage weight of each pixel, n_i is the number of pixels in each gray level and N is the number of pixels in the image.

$$P_{W_1} = \sum_{i=0}^{T} P_i \ , \ P_{W_2} = \sum P_i = 1 - P_{W_1} \quad (9)$$

In this equation, the total weight limits are expressed. The averages of W_1 and W_2 classes are calculated with the expressions given between Equations 10 and 13 [3].

$$\mu_{W_1} = \sum_{i=0}^{T} {i\,P_i}/{P_{W_1}} \quad (10)$$

$$\mu_{W_2} = \sum_{i=T+1}^{L-1} {i\,P_i}/{P_{W_2}} \quad (11)$$

$$\sigma^2(T) = P_{W_1} P_{W_2} \left(\mu_{W_1} - \mu_{W_2}\right)^2 \quad (12)$$

$$T^* = \underset{1 \le t \le L}{ArgMax}\, \sigma^2(T) \quad (13)$$

In this equation, T* represents the calculated otsu threshold. The otsu thresholding method is simple and stable. Therefore, it is often used [3]. Figure 10 shows the result obtained after applying the otsu thresholding technique to the plate image.

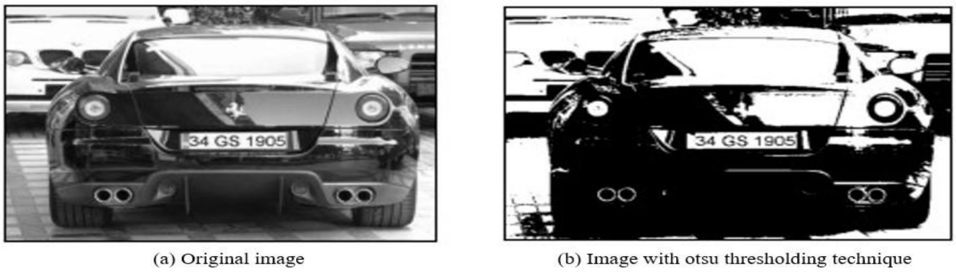


Figure 10. (a) Original license plate image and (b) image result applied the otsu thresholding technique

2.1.3. Morphology operations

Morphology is defined as the science of shape [20]. Morphology operations are one of the image processing methods created with the theory of sets based on the geometric shape of the objects in the image. Morphology operations distinguish objects from the digital image, extract wanted objects, and reduce noise and segment. Morphological filtering is used as a first or last preprocessing in image processing with targets such as refining and erosion. Morphology operations can be used in both binary and grayscale images [27]. Morphology operations were tried using the python programming language in the pycharm program. The outputs obtained are explained with images. Figure 11 shows the result of morphology operations such as erosion, dilation, and opening and closing techniques applied to the plate image.

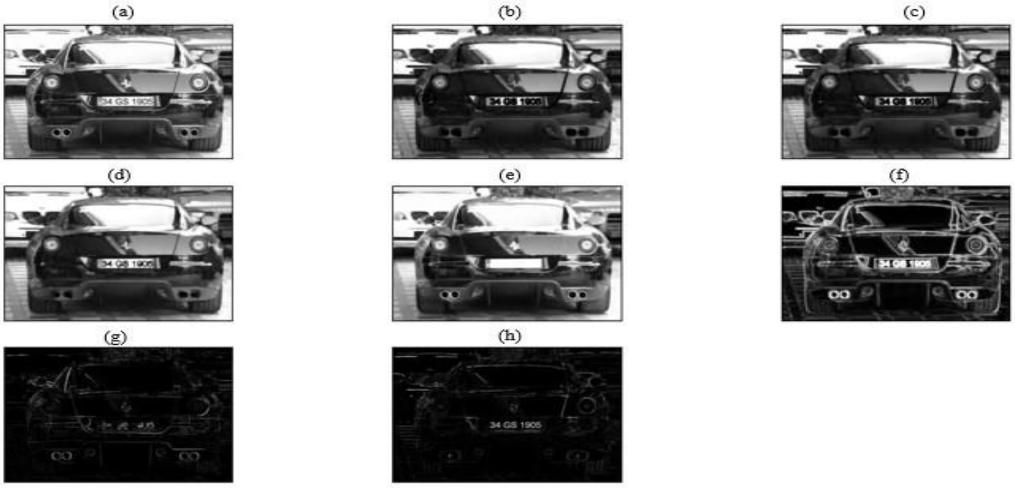


Figure 11. Original license plate image and images results applied morphological processing (a)Original image, (b) Image obtained as a result of erosion process, (c) Image obtained as the result of dilation process, (d) Image obtained after the opening process, (e) Image obtained after the closing process, (f) Image obtained as a result of gradient process, (g) Image obtained as a result of top-hat process, (h) Image obtained as a result of the black-hat process

2.2. Character Recognition

Character recognition is an image processing step that allows extracting text content from different data files. At this stage, Optical Character Recognition (OCR) engines enable converting text content in sources such as PDF files, books, paper documents, and images into computer-readable and editable electronic data [55]. Pytesseract, Microsoft Office Document Imaging, OmniPage, and Fine Reader are the most widely used optical character recognition software [56]. The flowchart of character recognition is shown in Figure 12.

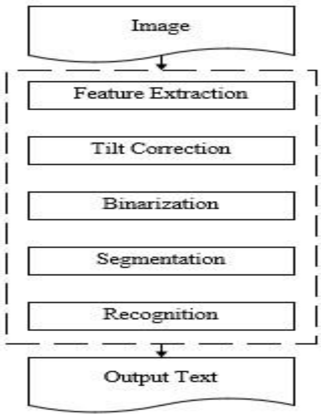


Figure 12. Flowchart of OCR [55]

3. Proposed Methodology

This study detected license plates with artificial intelligence and image processing techniques. In the study, 110 vehicle images obtained from the internet were used. Raspberry Pi 4 was used as a hardware component to create LPRS. Python programming language, OpenCV, NumPy, and imutils libraries were used as software components. The prepared LPRS consists of 3 stages. The first step is to find the plate region. For this, gray leveling, bilateral filtering, canny filtering, and contour extraction processes are applied to the plate image. The second stage is the cropping of the plate region. In the second stage masking method was used. The third stage is to recognize license plate characters. In this stage, the pytesseract algorithm was utilized.

The Raspberry Pi Foundation developed a credit card-sized SBC for artificial intelligence, machine learning, robotic coding, image processing, IoT, and education. It has features such as RAM, microprocessor, and GPIO pins on it. Raspberry Pi 4 has a 1.5GHz quad-core 64-bit Cortex-A72 processor, 2GB RAM, two USB 2.0 and two USB 3.0 ports, two micro HDMI ports, and one MIPI DSI display port, and 1 MIPI CSI. In addition, it has a camera port, audio and composite video output, USB-C type adapter port, Gigabit Ethernet port, and SD card slot. It also supports Bluetooth 5.0 and 5.0 GHz Wireless. Raspberry Pi's operating system is Linux-based Raspberry Pi OS.

Using Raspberry Pi 4 SBC, LED light control with voice command, face detection system, object detection and color recognition system based on OpenCV, smart speaker systems, etc. Many systems can be developed. Figure 13 shows the features of Raspberry Pi 4 SBC.

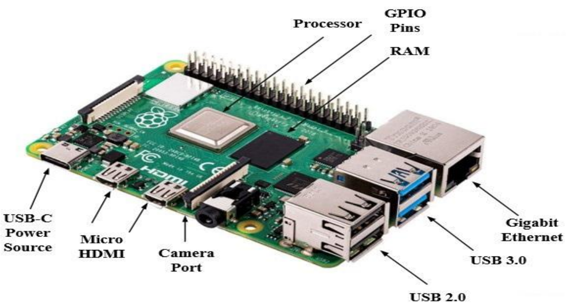


Figure 13. Raspberry Pi 4 single-board computer features

The applied preprocessing techniques used in the proposed methodology are presented as follows:

Step 1: Grayscale conversion

The grayscale image consists of gray in addition to black and white. The original image is converted to a gray level since the lousy quality of conversion in some color spaces and expressing each color space with different geometric shapes [57]. Figure 14 shows an example of the original license plate image converted to a gray level.

Figure 14. (a) Image of the original license plate and (b) converted to gray level

Step 2: Bilateral filtering

The bilateral filtering method was used to reduce noise and protect the plate edges. Figure 15 shows the bilateral filter applied to the gray level image.

(a) Original image (b) Bilateral filtered image

Figure 15. (a) Gray-level plate image and (b) bilateral filtering

Step 3: Canny filtering

The canny filtering method was used to determine the plate edges in the image sharply and clearly. Figure 16 shows the canny filter applied.

Figure 16. Canny filtered image result

Step 4: Contour extraction

Contour operation determines the region's boundaries to be cropped or cut from the image. The image must be converted to a gray level before the contour operation can be performed. Contour operation was used to find the rectangular plate region in the image.

The applied segmentation techniques used in the proposed methodology are presented as follows:

Step 1: Masking

The masking method is used to cut the wanted area from images and to cover unwanted areas.
Figure 17 shows the plate image clipped by a masking method.

Figure 17. Cropped image with masking method

The applied recognition techniques used in the proposed methodology are presented as follows:

Step 1: Tesseract algorithm

Tesseract is a fast and open-source software developed by Hewlett-Packard. It supports many languages and is free to use. The tesseract algorithm has three parameters: Language, Text Layout, and Characters. The default language is English. It also supports the Japanese language. There are three options in Text Layout. “Word” takes characters in an image as a single word and processes them. “Line” takes a single line of text. “Block” takes the text in images as a block and processes it. The characters parameter supports all characters by default. Tesseract processes using artificial intelligence [56].

4. Application

4.1. Overall Framework

First, the previously captured plate image is uploaded to the system for reading. Then, the image is converted to a gray level to process the image more efficiently and obtain accurate results. Next, a bilateral filtering method is applied to remove the noise on the image. Then canny filtering method is applied to make the plate area sharp and clearly. After these stages system searches for the area to be contoured, that is, it searches for the rectangular plate region. After the system finds what might be the best plate region, it detects the edges of the region and connects edges by drawing lines. Then plate region found is cropped by the masking method. Finally, image characters on the plate image are converted into text using the pytesseract algorithm. Outputs of the system are an original vehicle image, cropped plate region, and text. A basic flowchart of how the system works is given in Figure 18.

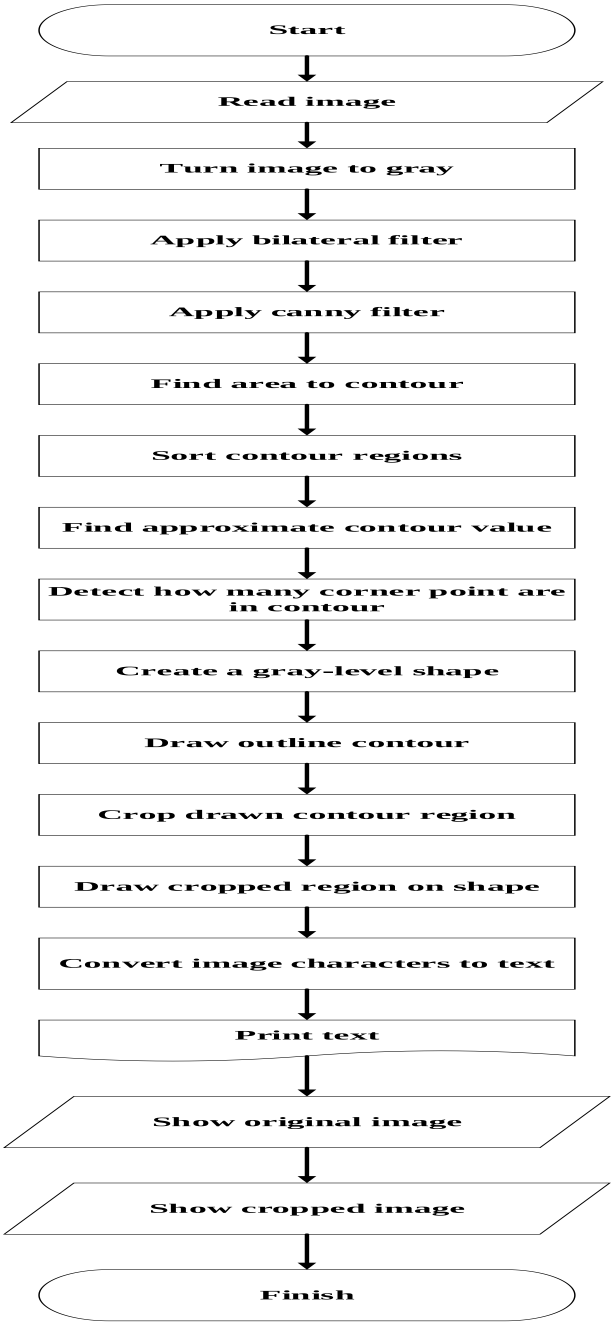


Figure 18. Basic flowchart of LPRS

Figure 19 shows the image given to the system for reading.

Figure 19. Original vehicle image

The plate region is cropped by applying gray leveling, bilateral filtering, canny filtering, contour extraction, and masking to the original vehicle image. Figure 20 shows obtained plate image.

Figure 20. Cropped plate image

Text output is shown in Figure 21.

Figure 21. Plate translated to text

5. Conclusion and Recommendations

In this study, an LPRS was created by using artificial intelligence and image processing techniques. Raspberry Pi 4 SBC was used as hardware to create the system. Python programming language was used in the system. Opencv, NumPy, imutils, and pytesseract libraries were installed on the system. In the study, 110 vehicle images taken from the internet were used. Prepared LPRS; It consisted of three stages: finding license plate region, cropped plate region, and recognizing plate characters.

In the first stage, we use Bilateral and canny filterings for grayscale conversion to clear the image's noise and make it easier to locate the plate region. Also, in this stage, we used to contour to detect the plate region in the image. The results revealed that 110 image locations were adequately detected. Next, we used the second stage masking method to crop license plates. Again, the results indicated that the used images (110 images) were successfully cropped in this state. Finally, we used the pytesseract algorithm in the last state to recognize license plate characters and translate those into text. The majority of the license plate characters were recognized correctly in this stage. More specifically, this algorithm correctly recognized 101 images' characters (out of 110 images' characters), referring to a 91.82% successful rate. As a result, our results suggest that LPRS can be considered a successful system.

Some difficulties were encountered within the scope of the study. One of the significant difficulties was that the symbols and flags on the plate made it difficult to read the plate. Another major challenge was that the license plate was more difficult to read in images taken at night. Comparing, it has been seen that the images taken during the day are easier to read.

Based on the experience we obtained from this study, some suggestions for future studies are presented to the researchers. LPRS can be developed not only in rectangular sizes but also in different sizes. LPRS can be developed to identify images, video, and real-time license plates. ANN technique can be used for LPRS. A comparison of success between the plates in our country and the international ones can be made. In addition, other than LPRS studies, studies can be carried out to determine the model, color, and speed of the vehicle.

Author's Contributions

This article is derived from the Master's Thesis "Vehicle License Plate Recognition System Based On Artificial Intelligence". All authors have read and agreed to the published version of the manuscript.

Competing Interests

The author(s) declare that they have no competing interests.

References

[1]. Trotta, D., Garengo, P., “Industry 4.0 key research topics: a bibliometric review”, 2018 7th International Conference on Industrial Technology and Management (ICITM), 2018, 113-117.
[2]. Pirim, H., “Yapay zekâ”, Journal of Yasar University, 2006, 1(1): 81-93.
[3]. Yilmaz, A., “Yapay zekâ”, 10th, I. Soylu, (Eds.), In KODLAB, Istanbul, (2022).
[4]. Ekrem, O., Salman, O. K. M., Aksoy, B., Inan, S. A., “Yapay zekâ yontemleri kullanilarak kalp hastaliginin tespiti”, Journal of Engineering Sciences and Design, Special Issue: International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2020), 2020, 241-254.
[5]. Jeong, D., “Artificial intelligence security threat, crime, and forensics: taxonomy and open issues”, In IEEE Access, 2020, 8: 184560-184574.
[6]. Yannakakis, G. N., Hallam, J., “Towards optimizing entertainment in computer games”, Applied Artificial Intelligence, 2007, 21(10): 933-971.
[7]. Pallathadka, H., Ramirez-Asis, E. H., Loli-Poma, T. P., Kaliyaperumal, K., Ventayen, R. J. M., Naved, M., “Applications of artificial intelligence in business management, e-commerce and finance”, Materials Today: Proceedings, 2021.
[8]. Zlanatov, S., Popesku, J., “Current applications of artificial intelligence in tourism and hospitality”, In Sinteza 2019-International Scientific Conference on Information Technology and Data Related Research, Singidunum University, Serbia, 2019, 84-90.
[9]. Cope, B., Kalantzis, M., Searsmith, D., “Artificial intelligence for education: knowledge and its assessment in ai-enabled learning ecologies”, Educational Philosophy and Theory, 2021, 53(12): 1229-1245.
[10]. Kaya, O., “Artificial intelligence in banking”, Accessed: May 16, 2022. [Online]. Available: https://www.dbresearch.com/PROD/RPS_ENPROD/PROD0000000000495172/Artificial_intelligence_in_banking%3A_A_lever_for_pr.pdf.
[11]. Olayode, O. I., Tartibu, L. K., Okwu, M. O., “Application of artificial intelligence in traffic control system of non-autonomous vehicles at signalized road intersection”, Procedia CIRP, 2020, 91: 194-200.
[12]. Tektas, M. Akbas, A., Topuz, V., “Yapay zekâ tekniklerinin trafik kontrolunde kullanilmasi uzerine bir inceleme”, International Traffic and Road Safety Congress, Gazi University, Ankara, 2002.
[13]. Kocer, H. E., Cevik, K. K., “Artificial neural networks based vehicle license plate recognition”, Procedia Computer Science, 2011, 3: 1033-1037.

[14]. Chung, Y. N., Lu, T. C., Hsu, C. H., Wu, S. Y., "Applying real time image recognition technology to apartment parking lot security system", ICIC Express Letters, 2016, 10(4): 877-882.
[15]. Rashid, M. M., Musa, A., Rahman, M. A., Farahana N., Farhana, A., "Automatic parking management system and parking fee collection based on number plate recognition", International Journal of Machine Learning and Computing, 2012, 2(2): 94.
[16]. Ozbay, S., Ercelebi, E., "Automatic vehicle identification by plate recognition", International Journal of Computer and Information Engineering, 2007, 1(9): 1418-1421.
[17]. Huang, Y. P., Chen, C. H., Chang, Y. T., Sandnes, F. E., "An intelligent strategy for checking the annual inspection status of motorcycles based on license plate recognition", Expert Systems with Applications, 2009, 36(5): 9260-9267.
[18]. Parvin, S., Rozario, L. J., Islam, M. E., "Vehicle number plate detection and recognition techniques: a review", Advances in Science, Technology and Engineering Systems Journal, 2021, 6(2): 423-438.
[19]. Anagnostopoulos, C. N. E., "License plate recognition: a brief tutorial", IEEE Intelligent Transportation Systems Magazine, 2014, 6(1): 59-67.
[20]. Yousef, K. M. A., Al-Tabanjah, M., Hudaib, E., Ikrai, M., "SIFT based automatic number plate recognition", 2015 6th International Conference on Information and Communication Systems (ICICS), 2015, 124-129.
[21]. Qadri, M. T. and Asif, M., "Automatic number plate recognition system for vehicle identification using optical character recognition", In 2009 International Conference on Education Technology and Computer, IEEE, 2009, 335-338.
[22]. Kaplan, O., Sagiroglu, S., Colakoglu, F. O., "Arac plaka tanima sistemi", Electrical-Electronics and Computer Engineering Symposium (ELECO'2002), Bursa, 2002, 355-359.
[23]. Oral, M., Celik, U., "Motorlu arac plaka goruntulerinden karakter ayristirma ve tanima", 10th National Congress of Electrical-Electronic Computer Engineering, 2003, 1(2): 499-502.
[24]. Bingol, O., Kuscu, O., "Bilgisayar tabanli arac plaka tanima sistemi", Journal of Information Technologies, 2008, 1(3): 1-5.
[25]. Singh, A., Singh, P., "License plate recognition", Journal of Management and Service Science, 2021, 1(2): 1-14.
[26]. Guan, T., Zhen, Y., Song, X., Liu, J., Wang, M., Wang, Z., Zhang, Y., Zhang, D., "Design of license plate recognition system based on capsule network", Conference Series: Materials Science and Engineering (IOP), 2020, 892(1): 012049.
[27]. Rashtehroudi, R. A., Shahbahrami, A., Akoushideh, A., "Iranian license plate recognition using deep learning", International Conference on Machine Vision and Image Processing, IEEE, 2020, 1-5.
[28]. Li, M., Miao, Z., Wang, J., Wang, S., Zhang, Y., "A method of license plate recognition based on bp neural network with median filtering", Journal of Physics: Conference Series, IOP Publishing, 2019, 1229(1): 012033.
[29]. Ohzeki, K., Geigis, M., Schneider, S. A., "License plate detection with machine learning without using number recognition", In 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), IEEE, 2019, 333-340.
[30]. Zhang, M., Yu, W., Su, J., Li, W., "Design of license plate recognition system based on machine learning", In 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), IEEE, 2019, 518-522.
[31]. Cheng, C., Mei, L., Zhang, J., "License plate recognition via deep convolutional neural network", IOP Conference Series: Earth and Environmental Science, IOP Publishing, 2018, 189(6): 062030.
[32]. Lee, S., Son, K. Y., Yoon, B. W., Park, J., "Video based license plate recognition of moving vehicles using convolutional neural network", 18th International Conference on Control, Automation and Systems, 2018, 1634-1636.
[33]. Selmi, Z., Halima, B. M., Alimi, M. A., "Deep learning system for automatic license plate detection and recognition", 14th IAPR International Conference on Document Analysis and Recognition, IEEE, 2017, 1: 1132-1138.
[34]. Dalida, D. P. J., Galiza, N. J. A., Godoy, O. G. A., "Development of intelligent transportation system for philippine license plate recognition", IEEE Region 10 Conference (TENCON), 2016, 8(5): 3762-3766.

[35]. Odone, F., "Experiments on a license plate recognition system", DISI University degli Studi di Genova, Technical Report number: DISI-TR-06-17, 2017.
[36]. Nagare, P. A., "License plate character recognition system using neural network", International Journal of Computer Applications, 2011, 25(10): 36-39.
[37]. Oz, C., Koker, R., Cakar, S., "Yapay sinir aglari ile karakter tabanli plaka tanima", Electrical-Electronics and Computer Engineering Symposium (ELECO'2002), Bursa, 2002, 1-6.
[38]. Gonzales, R. C., Woods, R. E., "Sayisal goruntu isleme", Telatar, Z. (Eds.), Palme Yayinevi, Ankara, (2021).
[39]. Shinde, B. S., Dani, A. R., "The origins of digital image processing & application areas in digital image processing medical images", IOSR Journal of Engineering, 2012, 1(1): 066-071.
[40]. Qidwai, U., Chen, C. H., "Digital image processing: an algorithmic approach with Matlab", 1st ed., Chapman and Hall/CRC, 2009.
[41]. Burger, W., Burge, M. J., "Digital image processing: an algorithmic introduction using java", Springer, 2016.
[42]. Vasudevan, S., Nagarajan, M., "A basic study of image processing and its application areas", International Journal of Engineering Research and Technology, 2017, 6(7): 343-348.
[43]. Coady, J., O'Riordan, A., Dooly, G., Newe, T., Toal, D., "An overview of popular digital image processing filtering operations", 13th International Conference on Sensing Technology (ICST), 2019, 1-5.
[44]. Aksoy, B., "Python ile imgeden veriye goruntu isleme ve uygulamalari", Revised edition, Nobel, Ankara, 2021.
[45]. Cabello, F., León, J., Iano, Y., Arthur, R., "Implementation of a fixed-point 2d gaussian filter for image processing based on FPGA", In 2015 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), IEEE, 2015, 28-33.
[46]. Basu, M., "Gaussian-based edge-detection methods-a survey", In IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2002, 32(3): 252-260.
[47]. Nandan, D., Kanungo, J., Mahajan, A., "An error-efficient gaussian filter for image processing by using the expanded operand decomposition logarithm multiplication", Journal of Ambient Intelligence and Humanized Computing, 2018, 1-8.
[48]. Biswas, S., Ghoshal, D., "Blood cell detection using thresholding estimation based watershed transformation with sobel filter in frequency domain", Procedia Computer Science, 2016, 89: 651-657.
[49]. Yildirim, M., Kacar, F., "Adapting laplacian based filtering in digital image processing to a retina-inspired analog image processing circuit", Analog Integrated Circuits and Signal Processing, 2019, 100(3): 537-545.
[50]. Bhairannawar, S., Patil, A., Janmane, A., Huilgol, M., "Color image enhancement using laplacian filter and contrast limited adaptive histogram equalization", Innovations in Power and Advanced Computing Technologies (i-PACT), 2017, 1-5.
[51]. Xuan, L., Hong, Z., "An improved canny edge detection algorithm", 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), IEEE, 2017, 275-278.
[52]. Gavaskar, R. G., Chaudhury, K. N., "Fast adaptive bilateral filtering", In IEEE Transactions on Image Processing, 2019, 28(2): 779-790.
[53]. Hoang, N. D., "Detection of surface crack in building structures using image processing technique with an improved otsu method for image thresholding", Advances in Civil Engineering, 2018.
[54]. Huang, M., Yu, W., Zhu, D., "An improved image segmentation algorithm based on the otsu method", 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2012, 135-139.
[55]. Kuruvilla, J., Sukumaran, D., Sankar, A., Joy, S. P., "A review on image processing and image segmentation", 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), 2016, 198-203.
[56]. Gocerler, B., "Yapay zekâ tabanli plaka tanima ile bariyer kontrolu", M.Sc. thesis, Trakya University Institute of Science, (2022).
[57]. Piskin, M. Opencv ile Goruntu Isleme. Accessed: May 17, 2022. [Online]. Available: https://mesutpiskin.com/blog/wp-content/uploads/2017/01/OpenCV%20Kitap.pdf.