DOI: 10.24996/ijs.2023.64.6.38

ISSN: 0067-2904

Tackling Skewness, Noise, and Broken Characters in Mathematical Expression Segmentation

Yahya Baher Al-Askary*, Saad Al-Momen

Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq

Received: 28/2/2023 Accepted: 2/4/2023 Published: 30/6/2023

ABSTRACT:

Segmentation is one of the most computer vision processes importance, it aims to understand the image contents by partitioning it into segments that are more meaningful and easier to analyze. However, this process comes with a set of challenges including image skew, noise, and object clipping. In this paper, a solution is proposed to address the challenges encountered when using Optical Character Recognition to recognize mathematical expressions. The proposed method involves three stages: pre-processing, segmentation, and post-processing. During pre-processing, the mathematical expression image is transformed into a binary image, noise reduction techniques are applied, image component discontinuities are resolved, and skew correction is performed. Two skew correction methods are proposed: The first method is the Deskewing using iterative PCA, and the second method is the PCA prediction. The line fitting-correction image deskewing and both gave better results than the well-known Hough transformation method. In the segmentation stage, the vertical and horizontal distances between mathematical expression components are utilized to segment the components. Post-processing is employed to reassemble split symbols into a single entity. The proposed method achieves an average detection rate of 97.32%, demonstrating improved recognition outcomes for mathematical expressions.

Keywords: Segmentation, OCR, Skew correction, Hough transforms, PCA, Line fitting, Linking characters, Horizontal projection, Vertical projection.

معالجة الانحراف والضوضاء والحروف المكسورة في تجزئة صور التعبير الرياضي

يحيى باهر العسكري*، سعد محمد علي المؤمن

قسم الرياضيات، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصه

يعد التقسيم أحد أكثر عمليات رؤية الحاسوب أهمية، فهو يهدف إلى فهم محتويات الصورة عن طريق تقسيمها إلى مقاطع أكثر وضوحاً وأسهل في التحليل ومع ذلك، تأتي هذه العملية مع مجموعة من التحديات، بما في ذلك انحراف الصورة والتشويش وقص الكائن. في هذه البحث، تم اقتراح حل للتحديات التي يتم مواجهتها عند استخدام التعرف الضوئي على الحروف اثناء تمييز التعبيرات الرياضية في الصور . تتضمن طريقة التجزئة المقترحة ثلاث مراحل: المعالجة المسبقة والتجزئة والمعالجة اللاحقة. أثناء المعالجة المسبقة، يتم تحويل صورة التعبير الرياضي إلى صورة ثنائية، ويتم تطبيق تقنيات تقليل الضوضاء، وكذلك حل تقطع الرموز في الصورة،

*Email: yahia.baher2103m@sc.uobaghdad.edu.iq

كما يتم إجراء تصحيح الانحراف. تم اقتراح طريقتين لتصحيح الانحراف: اولاً, تصحيح الانحراف باستخدام تحليل المكونات الرئيسية التكراري و تصحيح الانحراف باستخدام تنبؤ تحليل المكونات الرئيسية مع تصحيح التركيب الخطي، وكلاهما أعطى نتائج أفضل من طريقة تحويل هوف المعروفة. في مرحلة التجزئة، يتم استخدام المسافات الرأسية والأفقية بين مكونات التعبير الرياضي لتقسيم المكونات. اخيرا استخدمت المعالجة اللاحقة لإعادة تجميع الرموز المقسمة ذات الكيانات المتعددة الى كيان واحد. الطريقة المقترحة حققت متوسط معدل اكتشاف 97.32٪، مما يدل على تحسين نتائج التعرف على التعبيرات الرياضية.

1. Introduction

Optical Character Recognition (OCR) is a technology that enables computers to identify and interpret text present in images and convert them into a machine-readable format. The origins of OCR can be traced back to the early 1900s when the technology was first developed to automate the process of reading and transcribing documents [1]. Initially, OCR systems relied on mechanical and electromechanical technology, which resulted in low accuracy. However, with the advent of digital technology in the 1960s and 1970s, the accuracy of OCR systems improved significantly. Further, advancements came in the 1980s and 1990s with the development of machine learning algorithms and the application of neural networks, which led to even greater recognition accuracy. In recent years, OCR technology has undergone rapid development due to the widespread availability of high-performance computing and the emergence of deep learning techniques. As a result, OCR is now widely used in various applications such as document scanning and indexing, digital archiving, and mobile text recognition [2].

Applying OCR to mathematical expressions is challenging due to the complexity of mathematical notation, including symbols that are not found in standard text, variations in font style and size, and different notations. This makes it difficult for OCR to accurately recognize and understand mathematical expressions [3]. The presence of noise in images such as smudges, scratches, or marks, presents another challenge in using OCR to segment and recognize mathematical expressions. This noise can affect the quality of the image and make it harder for OCR to accurately recognize the symbols [4].

Another more difficult challenge is when using OCR to recognize mathematical expressions, the image may be skewed which can make it difficult to properly segment the characters and recognize them. To improve the OCR performance, it is necessary to preprocess the image and correct any skew or other distortions before attempting to recognize the characters. The Deskew techniques can be applied using different algorithms such as Hough transform [5], edge detection [6], and Principal Component Analysis (PCA) [7]. These techniques are used to determine the rotation or skew of the image and then rotate or skew the image back to its original position. Principal Component Analysis (PCA) is a statistical technique that is commonly used in image processing to reduce the dimensionality of an image and to remove noise and other unwanted variations. One specific application of PCA in image processing is deskewed, which is the process of correcting the alignment of an image that has been rotated or skewed [7].

2. Related Works

Within the realm of mathematical expression recognition, numerous researchers have contributed various ideas and methods. One noteworthy contribution was made by a group of authors who proposed a novel OCR method for recognizing printed mathematical expressions. The method leveraged deep learning techniques, specifically Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs), resulting in high accuracy on several benchmark datasets. Furthermore, the authors introduced a new dataset that contains images of printed mathematical expressions, and demonstrated the efficacy of their approach to this data [8]. In another recent work, a different group of researchers presented a novel end-to-end OCR method for recognizing printed mathematical expressions. Their approach utilized Graph Convolutional Networks (GCNs) which yielded state-of-the-art performance on several benchmark datasets. By leveraging GCNs, the proposed method achieved superior results by effectively capturing and incorporating structural information of mathematical expressions [9].

These contributions highlight the ongoing progress and advancements in the field of mathematical expression recognition. Researchers continue to innovate and develop new approaches to tackle the challenges of recognizing and interpreting these complex symbols and structures.

This paper aims to propose a solution for the difficulties faced when using OCR for mathematical expression recognition by achieving precise segmentation of the components of the expressions, thereby enhancing recognition outcomes.

3. Proposed Method

The proposed method consists of three main stages: Preprocessing, segmentation, and post-processing. In the preprocessing stage, the image of the mathematical expression is converted into a binary image and a noise reduction operation is applied to address any noise introduced during the scanning process. Discontinuities in the image components are then addressed, which can result from defects in the scanning process or from the noise removal process. Finally, the skew correction is performed to correct any skewing that may have occurred during scanning. This step is crucial as the suggested segmentation stage heavily relies on the vertical and horizontal spaces separating the components of the mathematical expression. The details of this stage are given in section 3.

The proper alignment of the image achieved through skew correction enables the segmentation of the components of the mathematical expression by utilizing the vertical and horizontal spaces between them. The methodology for this segmentation is discussed in detail in section 4.

The segmentation stage of the image may produce inaccurate results, particularly with symbols that have multiple pieces, for example, symbols like ($=$, $\div$, $\equiv$, $\geq$, $\leq$). To address this issue, there is a need for a post-processing stage that involves rejoining these split symbols back into a single entity, ensuring the proper segmentation of the image. The details of this stage are given in section 5.
The proposed method is depicted in a block diagram format in Figure 1, which shows the three stages of the proposed method and the individual steps involved in each stage.

4. Preprocessing

There are several challenges that one may encounter when trying to segment a scanned image of a mathematical expression. These include noise [10], poor image quality [11], complex layouts, overlapping characters [12], different fonts and writing styles [13], non-uniform backgrounds [11], broken characters [14], and skew [6]. Preprocessing the image can help to address these challenges and improve the accuracy and reliability of the segmentation process. Preprocessing steps may include noise reduction, image enhancement, skew correction, and background removal.

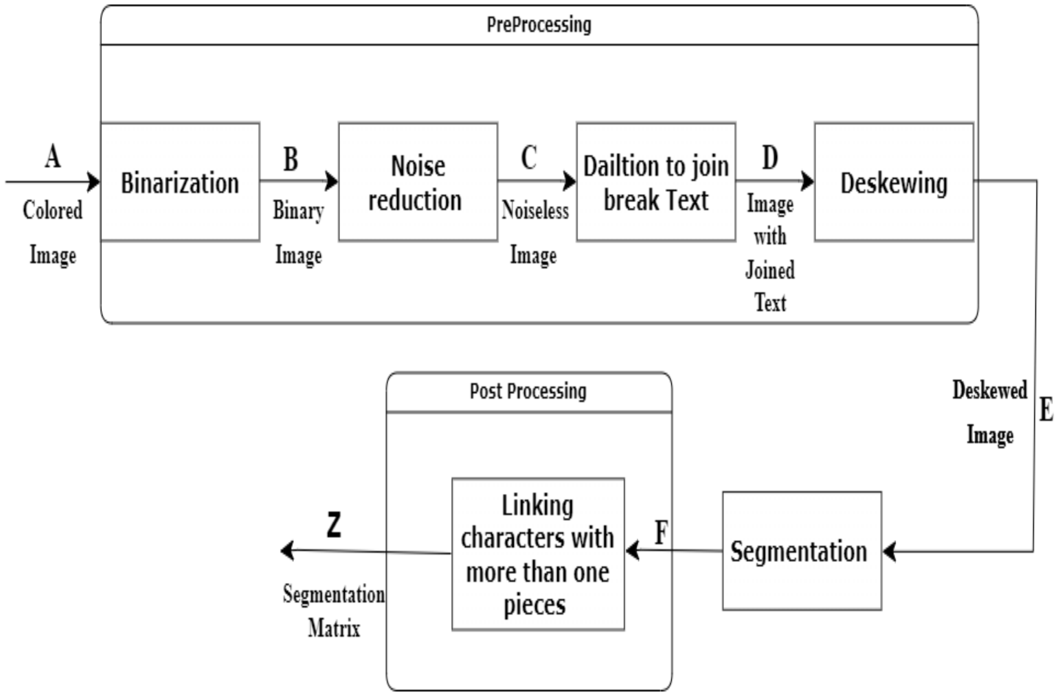


Figure 1: Block diagram of the proposed method.

4.1 Image Binarization

In the process of segmenting a mathematical expression, image binarization plays a crucial role in isolating the characters of the expression from the background and one another. This allows for more accurate segmentation of the image. There are several methods for performing image binarization including global thresholding [15], local thresholding [16], adaptive thresholding [17], and Otsu's method [18].

Otsu's method is a preferred option for image binarization due to its automatic nature and ability to determine the optimal threshold value based on the image histogram. This helps to effectively separate the foreground and background of the image. The effectiveness of Otsu's method has been proven in various applications [19], [20]. In this work, Otsu's method was used for image binarization to achieve accurate results.

4.2 Noise removal

Noise reduction is a crucial preprocessing stage for removing unwanted elements from images, particularly in the context of image segmentation where noise can result in over-segmentation. Median filtering [21], a non-linear approach, replaces a pixel value with the median value of the surrounding pixels, effectively removing noise and preserving image edges make it a valuable tool for image segmentation. The implementation of noise removal enhances the accuracy and reliability of image segmentation by minimizing the impact of noise on the image [22].

4.3 Joining the Broken Characters

Broken characters in mathematical expressions can pose a challenge in proper segmentation. To overcome this issue, the spaces in the broken text can be filled in through the use of dilation [14]. Dilation is a process in mathematics that enlarges an object without altering its shape, which is widely used in image processing to fill in spaces or gaps. By applying dilation to the broken text, the broken characters can be joined which leads to improving accuracy in the segmentation process.

Dilation is represented mathematically as the dilation of an image A using a structuring element$B$, it is written as follows [14]:

$$A \oplus B = \{z | \left(\hat{B}\right)_z \cap A \neq \emptyset\}. \tag{1}$$

This equation is based on obtaining the reflection of B about its origin and translating this reflection by z. A demonstration of joining broken characters can be found in Figure 2.

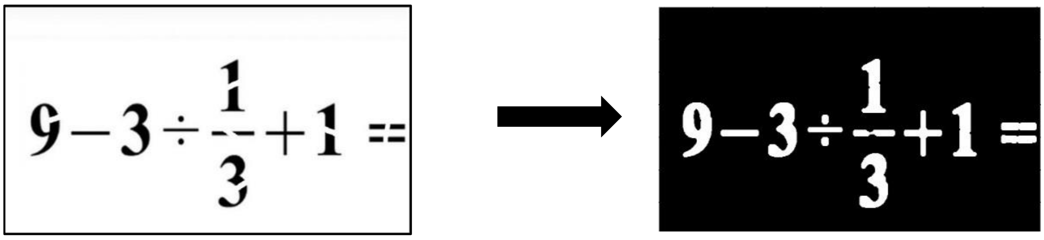


Figure 2: Joining broken characters using Dilation.

4.4 Deskewing

During the scanning of a document, skewness may occur, which leads to a distorted image of mathematical expressions. This can cause issues during segmentation as the proposed method relies on finding the vertical gaps between characters. To address this, the image should be preprocessed to eliminate the skewness and align horizontally with an angle close to zero [7]. Two methods have been proposed here for this, namely using iterative Principal Components Analysis (PCA) and finding a predicted rotation angle and correcting it through line fitting on points near the expression's midline.

The first method, using PCA, involves using the eigenvectors of the covariance matrix of the image to iteratively rotate the image until the angle of rotation is close to zero. The second method, finding a predicted rotation angle and correcting it through line fitting, involves using the horizontal and vertical projections of the image to find the angle of rotation. Both methods can be used to correct the skewness in the image and make it more suitable for segmentation.

4.4.1 Deskweing Using Iterative PCA

The mathematical foundation of PCA is based on linear algebra and eigenvectors. When it comes to image rotation, the PCA can determine the rotation angle of an image by identifying its principal component and using it to align the image. This approach resembles deskewing, but instead of using a fixed angle, the PCA extracts the angle from the data. The PCA can serve as a preliminary step before applying other image processing techniques to enhance their efficacy [23].

The focus is placed on the edges of the symbols and numbers in the equation to improve the calculation and accuracy. This is accomplished by employing the Canny filter, which applies two thresholds to the gradient: a high threshold for low-edge sensitivity and a low threshold for high-edge sensitivity. The edges are initially detected with low sensitivity and then expanded to include connected edge pixels from the high sensitivity result, which assists in filling any gaps in the edge detection [24] [25].

To perform PCA, the coordinates of the outline pixels generated by the canny filter are gathered in matrix A.

$$A = \begin{pmatrix} r_1 & c_1 \\ r_2 & c_2 \\ r_3 & c_3 \\ \vdots & \vdots \\ r_N & c_N \end{pmatrix}, \tag{2}$$

where r_i and c_i represents row and column coordinates, respectively, and N represents the total number of pixels.

The matrix A has a size of $N \times 2$. The mean M_A is a row vector containing the mean of the elements in each column of A is calculated using the following formula:

$$Mean: M_A = \frac{1}{N} \sum_{i=1}^{N} A_i. \tag{3}$$

And the covariance matrix C_A can be computed as:

$$C_A = \frac{1}{N-1} \sum_{i=1}^{N} (A_i - M_A)(A_i - M_A)^T, \tag{4}$$

The matrix C_A is a pair of orthogonal eigenvectors and is 2×2. Assume C_A is general matrix,

$$C_A = \begin{pmatrix} a & b \\ b & d \end{pmatrix}, \tag{5}$$

then $a = variance(a_{ii})$, $d = variance(a_{jj})$, and $b = \text{covariance}(a_{ij})$. The eigenvalues of this matrix represent the magnitude of the variances and eigenvectors that indicates the direction of these variances.
Then the eigenvectors can be calculated as follows:

$$v_k = \frac{1}{\sqrt{b^2 + (\lambda_k - a)^2}} \begin{pmatrix} b \\ \lambda_k - a \end{pmatrix}, \tag{6}$$

where v_k and λ_k $(k = 1,2)$ represent the eigenvectors and eigenvalues of C_A, respectively [26]. These eigenvectors are then reflected in a matrix X of the form:

$$X_{2\times 2} = (v_2 \quad -v_1). \tag{7}$$

The skew angle of the image can be extracted using the following formula:

$$\vartheta = \cos^{-1}(X_{1,2}), \tag{8}$$

where $X_{1,2}$ is the element in the first row and second column of matrix $X_{2\times 2}$.
Determining the direction of the skewed image, whether it is clockwise or counterclockwise, can be resolved using a simple formula:

$$\theta = \begin{cases} \vartheta, & if\ (X_{11} * X_{12}) > 0, \text{skwe angle is } CW \\ -\vartheta, & if\ (X_{11} * X_{12}) < 0, \text{skwe angle is } CCW \end{cases} \tag{9}$$

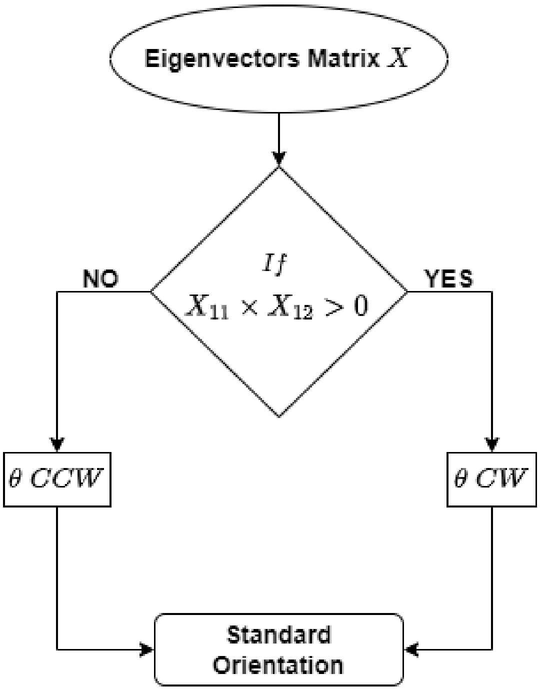


Figure3: Determining the direction of rotation (Clockwise or Counterclockwise).

So, the newly obtained angle θ represents both the skew angle and direction of the image's skew [7].

Since this method provides an approximation of the angle of rotation, it was used iteratively, to converge on an angle close to zero. This iterative process allows for fine-tuning the angle, ensuring that it is as close to zero as possible. Algorithm 1 and Figure 4 describe the steps of this method.

Algorithm 1: Iterative PCA Method for Image Deskewing
Input: Binary Image **Output**: Deskewed Image Step 1: Find the edges of the binary image. Step 2: Calculate the rotation angle using PCA, $\boldsymbol{\theta_i}$. Step 3: Rotate the binary image by $\boldsymbol{\theta_i}$. Step 4: Check if the angle is close enough to zero using a threshold value or a number of iterations. Step 5: If the angle is not close enough to zero and the number of iterations is less than the maximum iteration number, repeat step 2. Step 6: Return the deskewed binary image.

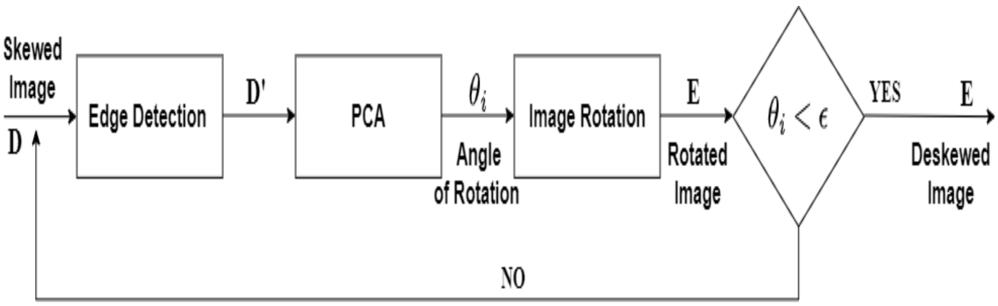


Figure 4: Flowchart of the iterative PCA-based image deskewing method.

4.4.2 PCA-Prediction and Line Fitting-Correction Image Deskewing

Using PCA, the predicted value of the rotation angle is found and used to rotate the image. Points centered around the central line of the equation are identified, and a line fitting is performed on these points. The angle of inclination of this line is calculated and adopted as the final rotation angle for the image. Algorithm 2 and Figure 5 describe the steps of this method.

Algorithm 2: PCA and Line Fitting based Method for Image Deskewing
Input: Binary Image **Output**: Deskewed Image Step 1: Find the edges of the binary image. Step 2: Use PCA to calculate the predicted value of the rotation angle for the binary image $\boldsymbol{\theta_1}$. Step 3: Rotate the binary image by the predicted angle $\boldsymbol{\theta_1}$. Step 4: Find points around the central line of the equation. Step 5: Perform line fitting on the found points. Step 6: Calculate the angle of inclination of the line. Step 7: Adopt this angle as the final rotation angle for the binary image $\boldsymbol{\theta_2}$. Step 8: Rotate the binary image by the final rotation angle $\boldsymbol{\theta_2}$. Step 9: Return the deskewed binary image.

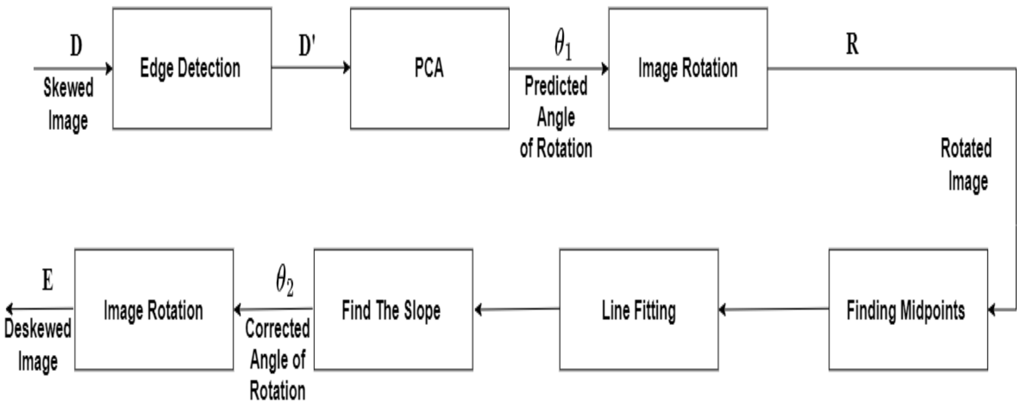


Figure 5: Flowchart of PCA and line fitting-based image deskewing method.

5. Segmentation

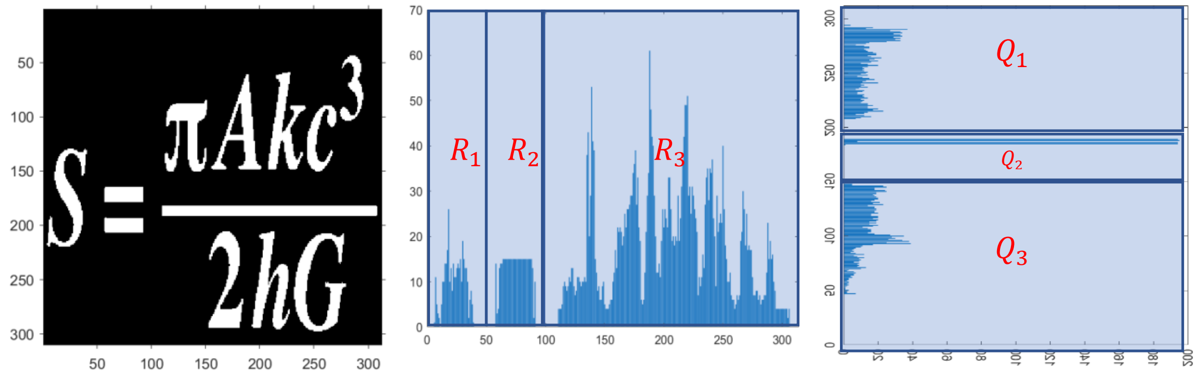
Binary image segmentation can be mathematically defined as the process of partitioning a set of pixels, B, in a binary image into multiple segments or regions, represented by a collection of non-empty disjoint subsets $P = \{P_1, P_2, \dots, P_n\}$. The partition P is defined such that the union of all subsets P_i is equal to B, meaning that every pixel in the binary image is assigned to exactly one subset in the partition P. This process can be represented by a mapping, $f: B \to P$, where for each pixel $b \in B$, $f(b)$ is the unique subset $P_i \in P$ to which b belongs, representing the segment or region of the binary image to which pixel b belongs.

Several methods can be used to perform binary image segmentation including thresholding [11], Connected Component Analysis (CCA) [27], edge detection [28], region growing [29], watershed algorithm [30] and machine learning-based methods [31]. The choice of the method depends on the nature of the image, the desired output and the computational resources available.

In this paper, the segmentation process was carried out based on the horizontal and vertical distances that divide the main components of the equation. Horizontal and vertical projections can be easily obtained by counting the number of 1 pixel for each bin in the vertical and horizontal directions, respectively. The projections are calculated by summing the values of the binary image B_{ij} along the rows H_i and columns V_j respectively, as described in the following equations

$$H_i = \sum_{j=1}^{M} B_{ij} \text{ where } (i = 1,2, \dots, N). \tag{10}$$

$$V_j = \sum_{i=1}^{N} B_{ij} \text{ where } (j = 1,2, \dots, M). \tag{11}$$



(a) **(b)** **(c)**

Figure 6: Vertical and horizontal projections of the binary image (a) Original image (b). The histogram of the vertical projection (c). The histogram of the horizontal projection of region R_3.

As a demonstration, Figure 6(b) shows that the histogram of the vertical projection of the equation is divided into three separate regions R_1,R_2 and R_3, while Figure 6(c) shows that the histogram of the horizontal projection of R_3 divides it into three regions Q_1, Q_2 and Q_3, and it is clear that they represent the numerator, fraction line and denominator of the fractional part of the equation, respectively. Algorithm 3 describes the steps of the segmentation stage.

Algorithm 3: Binary Image Segmentation
Input: Binary Image **Output**: Regions with their dimensions Step 1: Calculate the vertical projection of the binary image, $\boldsymbol{V_j}$. Step 2: Use $\boldsymbol{V_j}$ to calculate the number of regions in which the image can be divided. Step 3: For each region in the image, a. Calculate the horizontal projection, $\boldsymbol{H_i}$. b. Use $\boldsymbol{H_i}$ to calculate the number of sub-regions in which the region can be divided. c. If the number of sub-regions is one, i. Extract the dimensions of the region. ii. Explore the next sub-region. d. If the number of sub-regions is greater than one, i. Repeat the above steps for the sub-region. Step 4: Return the regions with their dimensions.

6. Post-Processing

In the projection and extraction of regions, certain characters are composed of multiple parts and thus split into multiple regions. To ensure precise character recognition and analysis, merging these regions is crucial in obtaining the complete character. Examples of such characters are $=$, $\div$, $\equiv$, $\geq$ and $\leq$. Consequently, addressing this issue is necessary for achieving accurate results.

6.1. Linking Characters with More Than One Pieces

During the segmentation stage, symbols like $=$, $\div$, $\equiv$, $\geq$ and $\leq$ can be divided into two or more pieces. For example, the symbol ($=$) can be split into two consecutive ($-$) symbols, as shown in Figure 7. It is important to reunite these segments to form a single entity.
These symbols have a common attribute: they contain either a dash ($-$), a dot ($\cdot$), or both. A classification system was created to determine which symbols should be fused together, which is based on several features such as the segment's width and height compared to the average values of the other symbols in the expression, the ratio of width to height of the segment, the area of the segment, and the distance between the centers of different segments. The average width and height were computed from the mainstream after removing any outlier values [32].

Figure 7 illustrates the process of image segmentation and linking to reconnect characters that have been split into multiple segments.

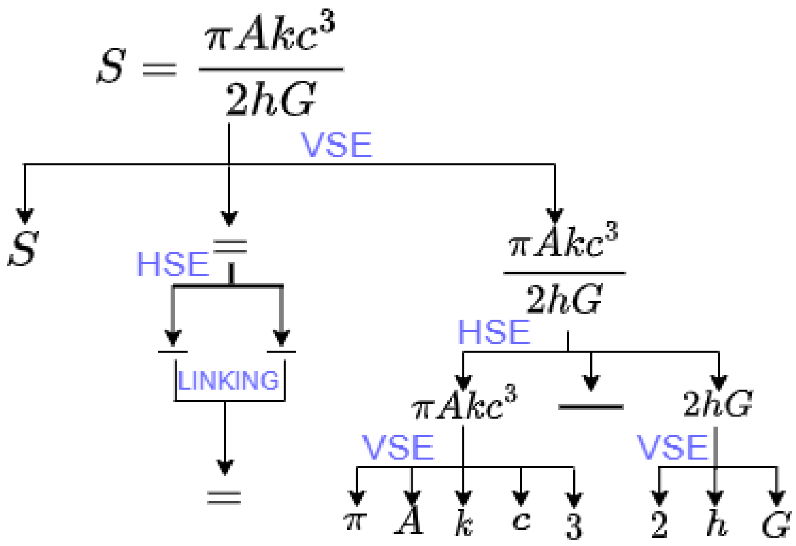


Figure 7: An illustration of image segmentation and linking characters composed of multiple pieces.

7. Experimental Results

Figure 8 displays six sample images that have been skewed at varying angles with corresponding the skewness angle noted below to each image. These images were selected to evaluate the performance of the two proposed deskewing methods.

Using the set of scanned images of mathematical expressions shown in Figure 8, the performance of skew correction was evaluated in terms of accuracy and speed by comparing the two proposed methods and the widely used Hough transformation method. The results are presented in Tables 1 and 2. Table 1 measures the error in detecting the actual skewness angle and Table 2 measures the processing time.

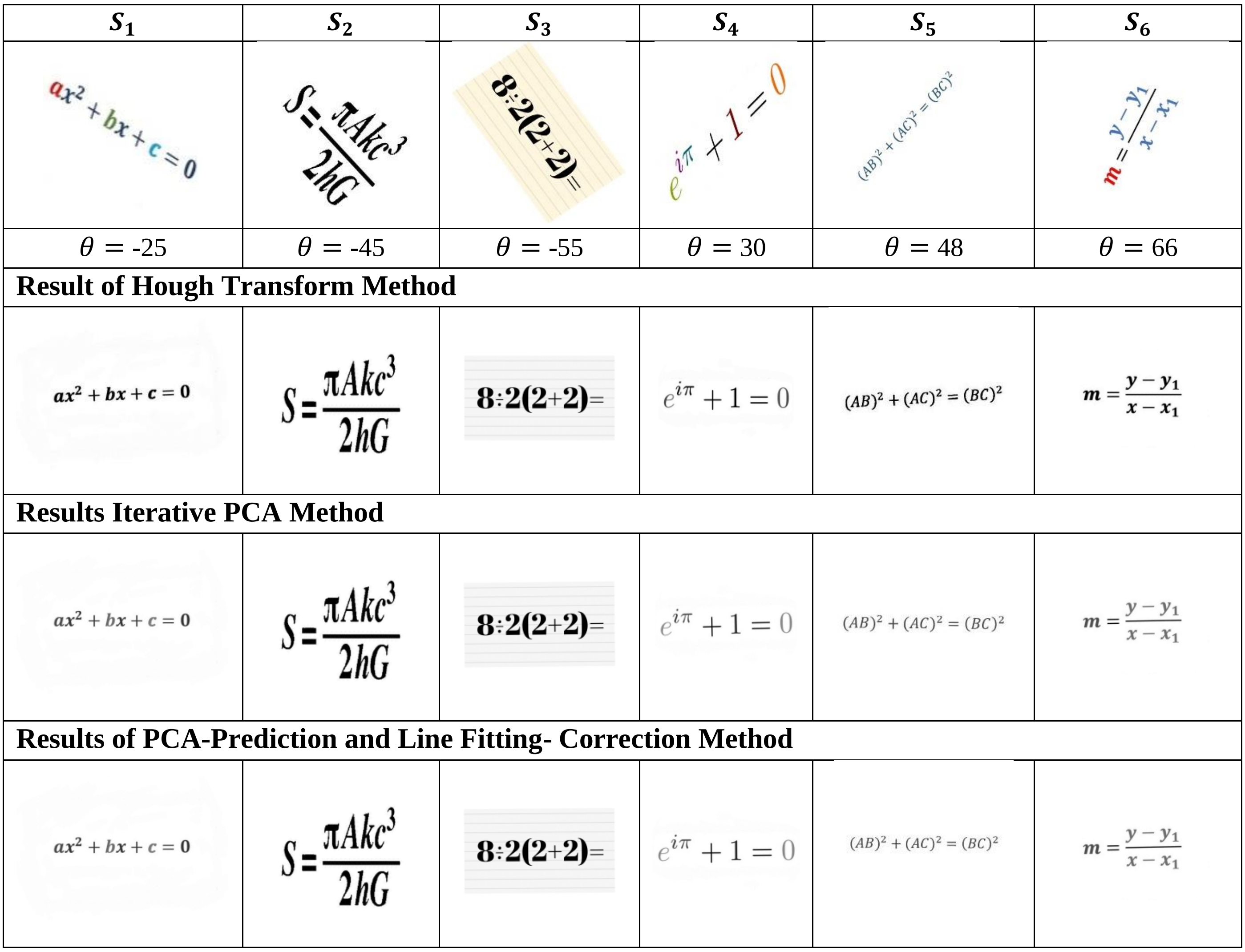


Figure 8: Skewed image samples at various skew angles with the deskew results

Table 1: Error in detecting skewness angle for scanned mathematical expressions.

Image	Skewness Angle	The error in detection skewness angle per method		
		Hough Transform	Iterative PCA	PCA-Prediction and Line Fitting-Correction
S_1	-25	2	0.057	0.102
S_2	-45	0	0.045	0.045
S_3	-55	0	0.031	0.432
S_4	30	0	0.050	0.521
S_5	48	2	0.061	0.288
S_6	66	0	0.055	0.633
Average Error		**0.667**	**0.049**	**0.336**

Table 2: Processing time comparison for skew correction methods

Image	**Processing time per method**		
	Hough Transformation	**Iterative PCA**	**PCA-Prediction and Line Fitting-Correction**
S_1	0.029	0.042	0.035
S_2	0.024	0.048	0.020
S_3	0.090	0.142	0.045
S_4	0.036	0.03	0.037
S_5	0.030	0.028	0.025
S_6	0.034	0.038	0.059
Average time	**0.0405**	**0.0546**	**0.0368**

Samples of 20 images that were used to verify the effectiveness of the proposed segmentation method are displayed in Figure 9. The rate of correct detection is presented in

Table 3: and the results of the segmentation process are shown in Figure 10.

$S=\frac{\pi Akc^3}{2hG}$	$B=\frac{\mu_0 x}{2\pi r}$	$9-3\div\frac{1}{3}+1=$	$8\div2(2+2)=$	$6\div2(1+2)$
Eq1	Eq2	Eq3	Eq4	Eq5
$-1\le\frac{x^2+y^2}{2}\le1$	$\pi\equiv\frac{22}{7}$	$N(t)=N_0e^{-\gamma t}$	$y=ax+b$	$m=\frac{y-y_1}{x-x_1}$
Eq6	Eq7	Eq8	Eq9	Eq10
$\Delta Z=h^2S$	$ax^2+bx+c=0$	$\frac{11+x}{x^3}+2x(5-x)$	$9x-7>3(3x-7u)$	$\frac{x^2+y^2}{r^2}$
Eq11	Eq12	Eq13	Eq14	Eq15
$3(x-5)=4-(x+3)$	$(AB)^2+(AC)^2=(BC)^2$	$X=2X+4$	$(X+3)^2=4$	$\theta=\pi-\emptyset$
Eq16	Eq17	Eq18	Eq19	Eq20

Figure 9: Sample images for evaluating the effectiveness of proposed segmentation method.

Table 3: Rate of correct detection for proposed segmentation method on sample images

#	Equation	Total no. of Characters	Detected Characters	Detection Rate
1	Eq1	11	11	100%
2	Eq2	9	9	100%
3	Eq3	10	10	100%
4	Eq4	9	9	100%
5	Eq5	8	8	100%
6	Eq6	12	12	100%
7	Eq7	6	6	100%
8	Eq8	11	9	81.8%
9	Eq9	6	6	100%
10	Eq10	11	11	100%
11	Eq11	6	6	100%
12	Eq12	10	10	100%
13	Eq13	15	15	100%
14	Eq14	13	13	100%
15	Eq15	8	8	100%
16	Eq16	14	14	100%
17	Eq17	17	15	88.23%
18	Eq18	6	6	100%
19	Eq19	8	6	75%
20	Eq20	5	5	100%
Average Detection Rate				**97.32%**

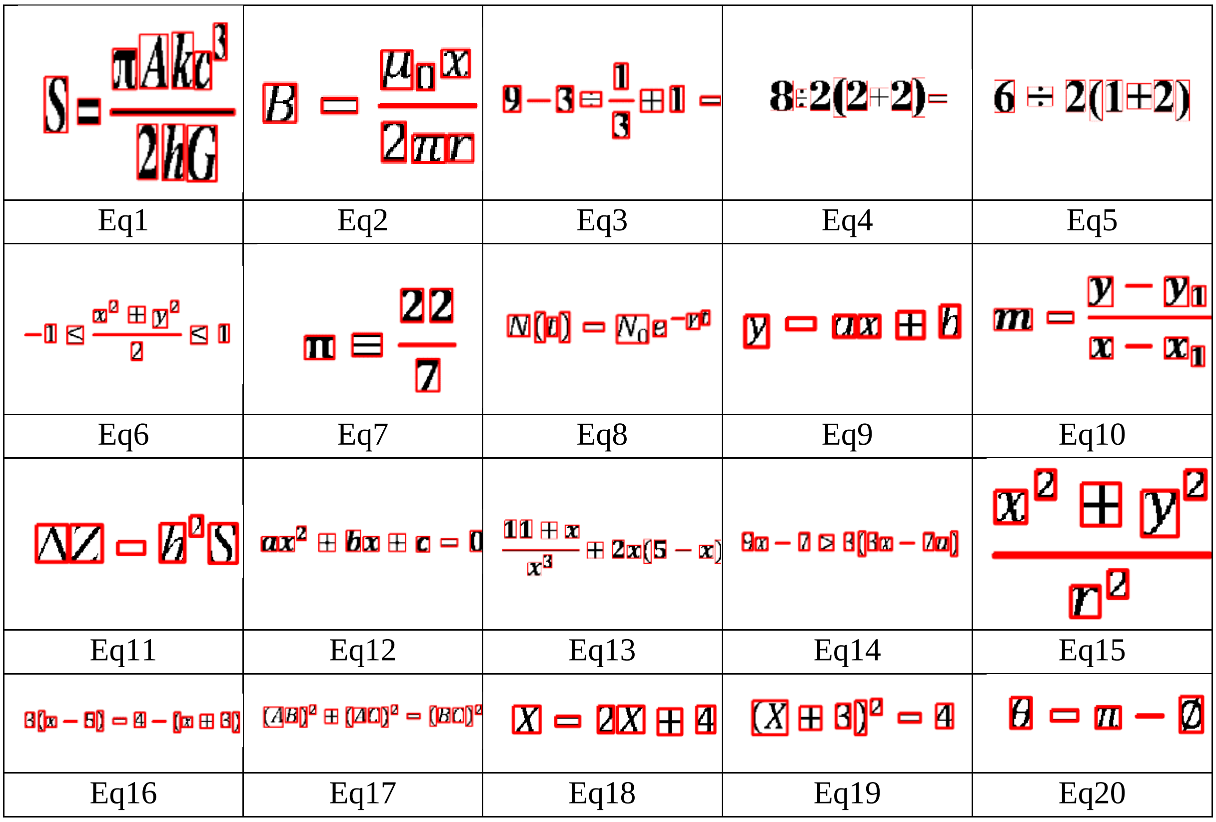


Figure 10: Results of segmentation process for sample images using proposed method

Table 4 presents a comparative performance analysis of the method introduced in this study alongside methods from previous research that is carried out by other scholars. The findings unequivocally indicate that the recommended approach outperforms the other techniques being considered in terms of efficiency.

Table 4: Comparative Performance Analysis of the Proposed Method and Previous Research Approaches.

#	References	Segmentation Accuracy
1	[33]	94.3%
2	[34]	94.04%
3	[35]	95%
4	[36]	96.4%
5	[37]	91.05%
6	Proposed method	97.32%

8. CONCLUSIONS

The results of this study demonstrate that the proposed method is effective in segmented mathematical expressions. The proposed method involves three stages, namely preprocessing, segmentation, and post-processing. The deskewing stage plays an important role to make the proposed segmentation algorithm that gives a good result. Two deskewing methods proposed in this study, namely the iterative PCA and PCA-prediction and line fitting-correction. Table 1 indicates that the two proposed methods produce better deskewing outcomes compared to the well-known Hough transformation method. On the other hand, Table 2 demonstrates that the PCA-prediction and line fitting-correction method outperforms the proposed Iterative PCA method and the Hough transformation method in terms of speed. The proposed segmentation method can segment the mathematical expressions with an average detection rate of 97.32%.

References

[1] Z. Li, Y. Li and J. Liu, "A Historical Review of Optical Character Recognition," *The Journal of Computer Science and Technology*, vol. 36, no. 4, pp. 797-819, 2021.

[2] R. Jain, A. K. Tripathi and P. K. Dwivedi, "Optical Character Recognition: A Historical Overview and Future Directions," *The Journal of Information and Knowledge*, vol. 21, no. 1, pp. 1-15, 2022.

[3] D. Zhelezniakov, V. Zaytsev and O. Radyvonenko, "Online Handwritten Mathematical Expression Recognition and Applications: A Survey," *IEEE Access*, vol. 9, pp. 38352 - 38373, 2021.

[4] B. H. Phong, T. M. Hoang and T.-L. Le, "A Hybrid Method for Mathematical Expression Detection in Scientific Document Images," *IEEE Access*, vol. 8, pp. 83663 - 83684, 2019.

[5] A. S. El-Bakry, M. S. Farag and A. M. Aly, "Deskewing Document Images Using Hough Transform and Distance Transformation," *The Journal of Imaging*, vol. 5, no. 9, p. 94, 2019.

[6] K. B. Khuwaja, M. A. Khan and A. R. Baig, "Document image skew detection and correction using edge detection and Hough transform," *Journal of Visual Communication and Image Representation*, vol. 63, pp. 360-370, 2019.

[7] H. Z. Rehman and S. Lee, "Automatic Image Alignment Using Principal Component Analysis," *IEEE Access*, vol. 6, pp. 72063 - 72072, 2018.

[8] Y. Xie, H. Lu, W. Lu and Y. Jia, "End-to-End Recognition of Printed Mathematical Expressions with Graph Convolutional Networks," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1825-1837, 2019.

[9] G. Goswami and P. Agarwal, "Deep Learning-based OCR for Printed Mathematical Expressions," in *International Conference on Recent Advances in Information Technology* (RAIT), 2020.

[10] F. G. Mohammed1, W. A. Amer and U. S. Al-hasani, "Noisy character recognition technique based on moments of inertiae," *International Journal of Advancements in Research & Technology*, vol. 2, no. 5, 2013.

[11] F. A. Dawood and Z. M. Abood, "Importance of Contrast Enhancement in Medical Images Analysis and Diagnosis," *International Journal of Engineering Research & Technology* (IJERT), vol. 7, no. 12, 2018.

[12] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 63-84, 2000.

[13] J. Chen, L. Chen and X. Liu, "A Survey of Optical Character Recognition for Mathematical Expressions," *the Journal of Computer Science and Technology*, vol. 34, no. 6, pp. 1151-1173, 2018.

[14] S. K. Maity and R. N. Biswas, "Broken character recognition and restoration in optical character recognition using enhanced morphological operations," *The Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 12, pp. 12915-12924, 2021.

[15] Y. Zhang, Y. Liu and Q. Liu, "A Hybrid Method for Image Binarization based on Global Thresholding and Local Adaptive Thresholding," *Journal of Multimedia Tools and Applications*, vol. 81, no. 7, pp. 10183-10201, 2022.

[16] R. S. Starikov, V. G. Rodin, V. V. Krasnov, N. N. Evtikhiev, E. A. Kurbatova and P. Cheremkhin, "Adaptive Digital Hologram Binarization Method Based on Local Thresholding, Block Division and Error Diffusion," *Journal of Imaging*, vol. 8, no. 2, pp. 120-131, 2022.
[17] X. Zhang and Y. Zhang, "An Improved Adaptive Thresholding Algorithm for Grayscale Image Binarization," *The Journal of Imaging*, vol. 6, no. 12, p. 128, 2020.
[18] P. Ranjitha and T. D. Shreelakshmi, "A Hybrid Ostu based Niblack Binarization for Degraded Image Documents," *in 2nd International Conference for Emerging Technology (INCET), Belagavi, India*, 2021.
[19] A. A. A. Karim and R. A. Sameer, "Comparing the Main Approaches of Image Segmentation," *Iraqi Journal of Scienc*e, vol. 58, no. 4B, p. 2211–2221, 2021.
[20] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab, Hoboken, New Jersey: John Wiley & Sons*, 2011.
[21] N. J. Habeeb, "Comparative analysis of Median filter family for Removing High-Density Noise in Magnetic Resonance Images," *Iraqi Journal of Science*, vol. 60, no. 10, p. 2246–2256, 2019.
[22] M. S. Al-Tamimi and R. S. Al-Khafaji, "Finger vein recognition based on PCA and fusionconvolutional neural network," *The International Journal of Nonlinear Analysis and Applications* (IJNAA), vol. 13, no. 1, pp. 3667-3681, 2022.
[23] Y. Zhang, Y. Wang and L. Zhang, "A Fast Image Deskewing Algorithm Based on PCA," *in 12th International Conference on Software and Emerging Technologies* (ICSET), 2021.
[24] J. F. Canny, "A Computational Approach To Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligenceadient*, vol. 8, no. 6, pp. 679 - 698, 1986.
[25] R. S. Upadhyay and P. Tanwar, "Detection of Bone Fracture using Canny Edge Detection Techniques," *International Journal of Scientific Research in Science and Technology* (IJSRST), vol. 6, no. 3, pp. 126-131, 2019.
[26] G. Strang, *Linear Algebra and Its Applications (Fifth edition), Wellesley, Massachusetts: Wellesley-Cambridge Press*, 2019.
[27] B. A. Hussain and M. S. Hathal, "Developing Arabic License Plate Recognition System Using Artificial Neural Network and Canny Edge Detection," *Baghdad Science Journal*, vol. 17, no. 3, p. 0909, 2020.
[28] S. M. Ali and S. AL-ZEWARY, "New Automatic Technique for Fingerprints Recognition and Identification," *MAAS journal of Islamic science*, vol. 10, no. 2, pp. 55-60, 1997.
[29] X. Jiang, Y. Guo, H. Chen, Y. Zhang and Y. Lu, "An Adaptive Region Growing Based on Neutrosophic Set in Ultrasound Domain for Image Segmentation," *IEEE Access*, vol. 7, pp. 60584 - 60593, 2019.
[30] F. H. Mahmood and N. A. Mahmood, "Automated Methods to Segment Kidneys and Detect Tumors Using CT Images," *Iraqi Journal of Science*, vol. 58, no. 3B, p. 1555–1564, 2021.
[31] E. Rodrigues, A. Conci and P. Liatsis, "ELEMENT: Multi-Modal Retinal Vessel Segmentation Based on a Coupled Region Growing and Machine Learning Approach," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 12, pp. 3507 - 3519, 2020.
[32] Y. V. Galperin, *An image processing tour of college mathematics, CRC Press*, 2021.
[33] S. Sakshi and V. Kukreja, "Segmentation and Contour Detection for handwritten mathematical expressions using OpenCV," *in 2022 international conference on decision aid sciences and applications* (DASA), 2002.
[34] M. Sarfraz, M. J. Ahmed and S. A. Ghazi, "Saudi Arabian license plate recognition system," *in Proceedings of the 2003 International Conference on Geometric Modeling and Graphics* (GMAG'03), 2003.
[35] C. Wu, L. C. On, C. H. Weng, T. S. Kuan and K. Ng, "A Macao license plate recognition system," *in 2005 International Conference on Machine Learning and Cybernetics*, 2005.
[36] J.-M. Guo and Y.-F. Liu, "icense plate localization and character segmentation with feedback self-learning and hybrid binarization techniques," *IEEE transactions on vehicular technology*, vol. 57, no. 3, pp. 1417-1424, 2008.
[37] K.-M. Hung and C.-T. Hsieh, "A real-time mobile vehicle license plate detection and recognition for vehicle monitoring and management," *in 2009 Joint Conferences on Pervasive Computing* (JCPC), 2009.