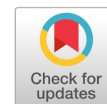


Detection of errors in the Indonesian standard *mushaf* based on pixels to support accelerated verification



Tri Wahyu Widyaningsih^{a,1}, Sarifuddin Madenda^{a,2,*}, Ravi Ahmad Salim^{a,3}, Nurma Nugraha^{a,4}

^a Gunadarma University, Depok, 16424, Indonesia

¹ tri.widyaningsih@tau.ac.id; ² sarifuddinmadenda@gmail.com; ³ ravisalim@gmail.com, ⁴ nurmanugraha01@gmail.com

* Corresponding author

ARTICLE INFO

Article history

Received October 27, 2024

Revised July 13, 2025

Accepted July 30, 2025

Available online November 30, 2025

Keywords

Indonesian Standard *Mushaf*

Pixel Matching

Block Image Mismatch Detection

Image Matching

Mushaf Verification

ABSTRACT

One effort to maintain the validity of the Al-Qur'an manuscript is the analysis and verification of the manuscript by experts (*Pentashib*). Currently, manuscript verification without translation takes 30 working days. Therefore, to support *Pentashib* in reviewing the manuscript, technology is needed to expedite the *Pentashib* process and prevent analysis errors caused by *Pentashib* fatigue. This study conducts a writing analysis of the target manuscript by referring to the template manuscript, implementing image preprocessing stages, applying SSIM for analysis, and employing the pixel-matching method. This method examines the manuscript's writing by comparing two block images at the pixel level. Block images are produced by preprocessing the manuscript images before image-matching analysis is performed. Image preprocessing comprises: cropping the outer frame, cropping the inner frame, segmenting the page into row images, adjusting margins, aligning image sizes, segmenting rows into block images, and aligning positions between block images. Pixel value differences are calculated at the same positions across each column and row of the template and target block images. Block image positions with pixel values ≥ 200 occur in 5 consecutive columns, adjacent rows with a distance = 1, and an SSIM value ≥ 0.9 , both images meet the mismatch criteria. These findings indicate that the proposed approach provides an efficient and accurate solution for automating the verification of the Indonesian Standard *Mushaf*.



© 2025 The Author(s).

This is an open access article under the [CC-BY-SA](#) license.



1. Introduction

Data can be in the form of text, images, sound, and video that can be processed, stored, and disseminated as information [1], such data cannot be analyzed before being processed [2]. Image data processing aims to improve image quality prior to subsequent processing [3]. Digital image processing uses computerized signal processing to process digital images [4]. This study uses image data as the research object, namely the Indonesian standard Al-Qur'an manuscript [5], which was originally text data and then converted into image data. The Qur'an consists of 30 parts (*juz*), 114 *surahs*, and 6,236 verses [6]. To ensure its accuracy, every copy of the Qur'an to be published in Indonesia must undergo a verification process carried out by the "Lajnah Pentashiban Mushaf Al-Qur'an" (LPMQ) unit. This institution was established by the Ministry of Religion, with the qualifications of a verifier including: memorizing the entire Qur'an, understanding the sciences of the Qur'an, and mastering verification techniques [7]. The LPMQ conducts research, inspection, and correction of the master copy of the Quran manuscript to be published by carefully and repeatedly reading it through the verifiers to ensure no errors are found, including the Ministry of Religion's translations and interpretations [8].

The verification process for one 30-*juz Mushaf* takes approximately one month because every punctuation mark, tajwid, and verse arrangement must be carefully and repeatedly checked to ensure accuracy. Proofreaders will check the completeness, sequence, and physical condition of the manuscript before conducting further checks [9]. This proofreading process presents several challenges, including the need for significant time, high precision demands, and susceptibility to human error due to fatigue. These limitations highlight the need to adopt technologies that can accelerate and improve the accuracy of the verification process. The visual verification process of the Qur'an involves a detailed examination. Technology that can compare the text in the image of the original manuscript or trusted reference (template) with the text in the image being examined (target).

The verification of the Qur'an can be performed using image-matching technology to facilitate automatic comparison between the original manuscript and the text being examined, thereby enabling faster, more accurate detection of writing errors, letter differences, and punctuation errors. The text being discussed will be referred to as the target image (text from the publisher), and the original text could be referred to as the template image, which is the text owned by LPMQ and used by publishers as a reference in writing and compiling the *Mushaf* text. The template and target images use the same font, and the same size. The technique employed is pixel-based image matching [10], also known as pixel matching.

Image matching employs several algorithms based on image content, features, structure, color, grayscale, and texture to identify image similarities. The image feature method involves extracting feature points from each image and matching them to those in another image. This method requires less computation, but has better robustness [11]. Image matching is a widely used technique in computer vision, with various applications; it is typically framed as a classification problem with a binary match/no-match output [12].

The term describing the elements of a digital image is a pixel, which has a specific location and value [13]. Not only in images, but other research has also applied pixel language models for text classification by language [14]. The connectivity between pixels represents the region and boundaries of a digital image. To determine if two pixels are connected, it is necessary to know whether the two pixels are adjacent and have the same gray level [15]. Pixel matching is a technique in image matching that compares pixels between two images. This method can be used to identify differences or similarities between two images by comparing pixel values at corresponding positions. If the pixel difference approaches 0, the two images can be considered similar; conversely, if it approaches 255, the two images are considered dissimilar. Pixel matching is an appropriate method for this study because the target image data and template images share similarities in text form, punctuation, and character spacing. Thus, they can be matched to identify similarities and differences by calculating pixel differences, as the initial step in the image matching process using the pixel matching method.

Image matching [16][17] have been used in various studies, including in article [11], which compares the matching of two images using SIFT and GCD under different conditions, namely matching two original images, matching images after rotation, and matching with other images. SIFT is not well suited to matching manuscript images because the lettering exhibits similar visual patterns, making it difficult for SIFT's keypoint-based matching to distinguish between lines or pages accurately. This article [12] used image-matching algorithms, namely SIFT, SURF, and ORB, to calculate the similarity between matched images and classify them into predetermined similarity categories. *Mushaf* manuscripts often exhibit similar letterforms, such as ب, ت, ث, so SIFT, SURF, and ORB will fail to distinguish between visually similar images. Other studies have used template-matching techniques to recognize letters using OCR (Optical Character Recognition) applications by calculating the degree of similarity between two images. The correspondence point value is calculated from the minimum error, which indicates the level of similarity [18]. The template-matching technique is unsuitable for this research because it matches a small template to a large image by shifting it and selecting the optimal position based on correlation values, so that images are considered similar solely because they are visually similar, even when they may differ. This technique may be more appropriate for image datasets than for text datasets [19]. Other

research uses image-matching methods to compare images and generates an average match value based on changes in image capture angle, scale, and rotation [20]. Another study employs image matching via feature matching with a Convolutional Neural Network (CNN) [21]. Using the template-matching method, Kiew *et al.* [22] developed an image recognition application for vehicle tracking based on license plates. Subsequent research employs template matching to enhance the minimum distance classifier [23].

Previous research has used the same data as our study and employed image matching. Among these is the recognition of Hijaiyyah letters using template matching with Optical Character Recognition (OCR). In that article, it was stated that the research worked well for letters with unique geometric shapes, such as ل and ق . However, there were still errors for similar letters, such as ح , ح , and خ [24], so the recognition of Hijaiyyah letters using template matching was not entirely successful. Another similar study involves extracting text from videos using OCR. The research focuses on 28 Arabic letters and 15 test images; however, if font variations are present, the dataset must be modified [25], even though a single Arabic letter can exhibit multiple calligraphic variations, necessitating repeated adjustments to the dataset. Another article employed template matching to convert the *Mushaf* page image into a binary image, then searched for keywords in the binary image using word image matching with Normalized Cross Correlation [26]. This matching aims to identify keywords in a *Mushaf* page image, rather than to detect writing errors in the image. Previous research characterized ancient document images composed of Arabic and Latin characters using image matching, fractal dimension, and image retrieval methods. The objective of this research was to classify page images as ancient Arabic or Latin documents [27], not to identify their content. This study [28] evaluated various feature-extraction and classification approaches for Arabic OCR systems over the past 10 years on an Arabic dataset. The evaluation results show that SIFT achieves the highest accuracy (94.28%), whereas other techniques, including commercial systems, have lower accuracy. SURF is faster but also has lower accuracy (90.46%). This research primarily compared the performance of several feature-extraction and classification methods, rather than analyzing writing errors in the *Mushaf* manuscript. Another study performed Arabic word recognition on encrypted digital documents for document management, but the new study was conducted on 10 document images containing 200 words using SURF feature extraction, achieving an accuracy of 96% [29], not for analyzing writing errors in page images.

From the studies described, no analysis of errors in the *Mushaf* manuscript has been conducted; only a few recognitions of Arabic letters and words have been performed using feature extraction and machine-learning-based OCR. Our research builds on prior work by matching images of *Mushaf* manuscript pages using pixel matching. This involves matching two images on each target page and template page. This matching process begins after image acquisition, preprocessing, and pixel-difference calculation, and proceeds to image-mismatch analysis. The researchers chose the pixel-matching technique because they had previously tested several methods for analyzing writing mismatches in manuscript images, and it proved the most suitable for implementation. This study contributes to the detection of script errors in the *Mushaf* by introducing an automated, efficient approach based on pixel-level matching between page images.

2. Method

This study employs the pixel-matching method to analyze discrepancies in writing by comparing template and target images based on pixel values and positions. If two block images differ slightly, they will be detected as different; therefore, the accuracy for detecting writing errors must be 100%. Meanwhile, modern methods such as machine learning and deep learning are not sufficient with only 100% accuracy. If the accuracy value is very high, further review and analysis are needed, such as research criticizing the high accuracy in disease diagnosis using machine learning algorithms [30], which states that maximum accuracy does not guarantee reliability and can even be obtained through infeasible experiments. Therefore, it is necessary to report the minimum and average accuracy values to assess model stability, as well as the importance of transparency and comprehensive evaluation. Another study

[31] states that high accuracy does not guarantee reliability and is prone to errors, so it is not enough to have a well-trained model but also requires a fault-tolerant architecture.

Image matching has various image matching techniques, such as template matching, SIFT, ORB, SURF, and pixel matching. SIFT, ORB, and SURT are not used because they are easier to detect features with high contrast rather than letter images that have very similar structures and patterns. This study also analyzed image matching of *Mushaf* manuscripts using template matching; however, the results were inaccurate because the sliding-window approach is not suitable for *Mushaf* manuscripts, which exhibit similar, nearly identical writing styles. The sliding window is a stage in which a small template is matched to a larger image by shifting it and finding the optimal position. For *Mushaf* page images with similar letter shapes, false matches will be found.

The research stages are shown in Fig. 1, beginning with planning, data acquisition, image preprocessing, pixel calculation on the image block, image mismatch analysis, and image matching analysis results. Planning, in the form of problem identification, was carried out by interviewing sources, including the head of the LPMQ institution and the *Pentashib*, to obtain information on the current *tasbih* system and the time required for *pentashiban* for several manuscript types. Data acquisition is the stage of collecting template and target data, which initially consists of master files *per-juz* in PDF format converted into image data per page in PNG format. Template data refers to the official reference manuscript owned by LPMQ, and target data refers to publisher-provided data to be analyzed.

2.1. Image Preprocessing

Image processing is the processing of digital images with computerized image input and output [15]. The image preprocessing stage prepares the template and target images before matching. The input data consists of template and target images in PNG format, converted from color to grayscale [13]. Next is the adjustment of the frame design, as each page of the Quran has an outer frame design with various variations in shape, size, and color. To facilitate the image matching analysis process, it is necessary to crop [32] the outer frame on each page so that the research focuses on the writing of verses on each page, using Equation (1). The cropping process of the outer frame starts from the top left ordinate to the bottom right, from the top row y to the bottom row $y + h$, and from the leftmost column x to column $x + w$ dengan h is the image height w is the image width.

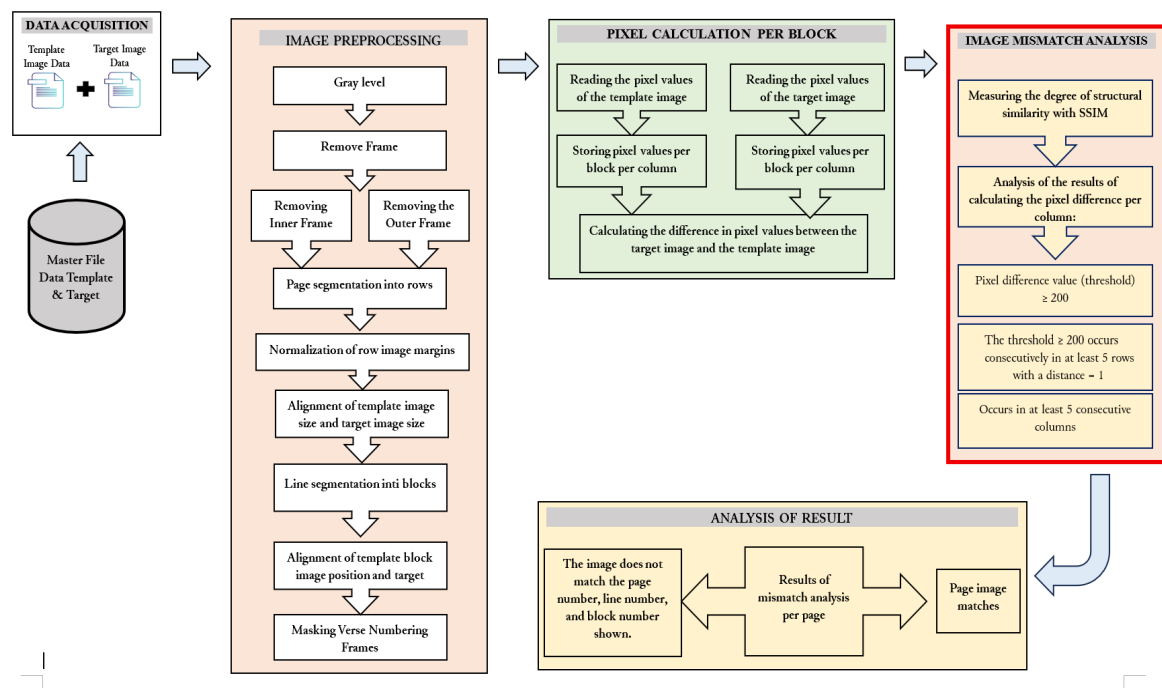


Fig. 1. Research stages

Not only is there an outer frame, but there is also an inner frame in the manuscript as a separator between letters. Equation (2) detects the image area of the lines. Equation (3) detects the empty area between two images, and equation (4) detects the empty areas above and below.

$$\text{Crop Region} = \text{Image} [y: y + h, x: x + w] \quad (1)$$

$$\text{contour} : h_{cnt} > 10 \text{ and } w_{cnt} > \frac{w}{2} \quad (2)$$

where h_{cnt} represents the total number of lines in the image, corresponding to its vertical resolution, while w_{cnt} denotes the number of columns, reflecting the horizontal resolution. In addition, w indicates the overall width of the analyzed image, typically expressed in pixels, and is used as a reference for spatial measurements and further image-processing operations.

$$y_{empty} = y_1 + h_1, h_{empty} = y_2 - (y_1 + h_1) \quad (3)$$

where y_1 represents the vertical starting position (row coordinate) of the first image row, indicating where the top of image row 1 is located in the overall image. The variable h_1 denotes the vertical position or height reference associated with the second image row, which helps define the relative placement between consecutive image rows. Meanwhile, y_2 specifies the vertical starting position (row coordinate) of the second image row. To describe the gap between these two rows, y_{empty} indicates the vertical starting position of the empty region located below image row 1, while h_{empty} represents the height of this empty area separating image rows 1 and 2. Together, these parameters precisely define the vertical layout of image rows and the intervening empty space within the analyzed image.

$$y = y_n + h_n \text{ and } h = H - (y + h) \quad (4)$$

where $y = 0$ and $h = y_1$ measure the height of the empty area from the top of the image to the beginning of the first row. Meanwhile $y = y_n + h_n$ and $h = H - (y + h)$ are used to calculate the empty space at the bottom after the last image is finished.

Each normal page has fifteen lines. After cropping the frame on each page, the page will be segmented into lines. Segmentation can use OCR techniques [33], but in this study, it was obtained through several processing stages because the page images have varying numbers of lines. The segmentation steps include horizontal projection to calculate the intensity in each line. If the projection value is high, the line is a text line; otherwise, it is a space between text lines. Horizontal projection is shown in Equation (5). Marking rows containing text is shown in Equation (6). If a row is not detected, it is divided into 15 rows as shown in Equation (7). Equation (8) shows how the template row image is aligned with the target row image.

$$\text{projection} [y] = \sum_x \text{binary} (x, y) \quad (5)$$

$$\text{is_text}[y] = \begin{cases} \text{True, if } \text{projection} [y] > 2 \\ \text{False, others} \end{cases} \quad (6)$$

$$y_i = i * \frac{H}{15} \quad (7)$$

$$\text{Template}_{\text{resized}} = \text{Resize} (\text{Template}_{\text{original}}, (W_{\text{target}}, H_{\text{target}})) \quad (8)$$

Research on various line detection and line segmentation methods has been extensively conducted [34], although in this study we have not used modern segmentation techniques; however, projection-based segmentation has been able to perform line segmentation effectively.

After adjusting the size [35] of the template and target images, both images have the same width and height. This is followed by line segmentation into blocks to facilitate matching the two images, as the

smaller image size makes it easier to analyze pixel values, pixel differences, and the positions of different image areas. Equation (9) shows line segmentation into blocks.

$$\begin{aligned}x_{end} &= W - i * w_{block} \\x_{end} &= W - (i + 1) * w_{block} \\block_i &= img[0:H, x_{start}:x_{end}]\end{aligned}\tag{9}$$

After image preprocessing, once the lines have been segmented into block images, it is necessary to align the positions [36], of the template and target block images so that they are in the correct positions. Equation (10) shows the alignment of the positions of the template and target block images before matching.

$$\begin{aligned}img_{shifted}(x, y) &= img2(x - dx, y - dy) \\diff(x, y) &= |img1(x, y) - img2(x - dx, y - dy)| \\score &= \sum_{x,y} (diff(x, y) * mask(x, y))^2\end{aligned}\tag{10}$$

The final stage of image preprocessing is masking [37] the verse numbering frame. This is necessary because each publisher uses a different verse-numbering frame design, similar to the outer frame on each page, which also varies. Therefore, masking must be performed on the verse numbering frame area in both block images so that the matching process does not need to be performed. The masking steps for the verse numbering frame are outlined in equation (11).

$$\begin{aligned}A &= area(contour) \\Ratio &= \frac{\max(M A, ma)}{\min(M A, ma)} \\500 < A < 6000 \text{ and } 0.7 < \frac{M A}{ma} < 1.2\end{aligned}\tag{11}$$

2.2. Pixel Matching

The image matching stage uses template and target block images as input data to be compared. The image rows are segmented into 5 blocks to simplify the pixel analysis process. If the number of block segments per row exceeds 5, too many letters and punctuation marks will be cut off in that row, resulting in inaccurate error analysis. If the number of blocks is less than 5, there will be too many pixel columns per block, making pixel analysis per block more difficult. Dividing the row into 5 blocks results in an average of 201–203 columns per block.

After obtaining the pixel values in the block image, the pixel difference is calculated using Equation (12).

$$Pixel\ difference = |P1(x, y) - P2(x, y)|\tag{12}$$

where $P1(x, y)$ denotes pixel value of template, while $P2(x, y)$ represents the pixel value of the target.

2.3. Measuring block image similarity

Before matching images based on the pixel difference between two block images, it is necessary to visually assess the structural similarity between the two template and target block images using the Structural Similarity Index Measure. The SSIM formula for two images is given in Equation (13). SSIM is used to improve the accuracy of detecting the input signature against the reference signature, thereby improving the accuracy of matching two signature images [38]. Other studies have also used SSIM to evaluate the accuracy of overlapping-region calculations from phase correlation during large-image

stitching [39]. In another article, it is stated that SSIM is a technique that closely approximates human visual perception compared to similar methods across various medical imaging modalities [40]. In this study, SSIM is used as an initial step in measuring the similarity between two block images. However, since SSIM has the drawback of maintaining a high value even when there are small differences between two images, a more in-depth analysis using pixel difference values is necessary.

$$SSIM(x, y) = [l(x, y)]^\alpha * [c(x, y)]^\beta * [s(x, y)]^\gamma$$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{13}$$

2.4. Analysis of block image mismatches

The discovery of indications of writing errors, if they meet the criteria, including having a pixel difference value ≥ 200 . This value is repeated across 5 columns within a row and occurs in adjacent rows with a distance of 1. Fig 10 and Fig 12 show the results of the analysis between block images that have indications of writing errors. This stage is explained in more detail in the results and discussion.

3. Results and Discussion

3.1. Grayscale

The process of removing the inner and outer frames, as well as converting RGB colors to gray levels, can be seen in Fig 2.

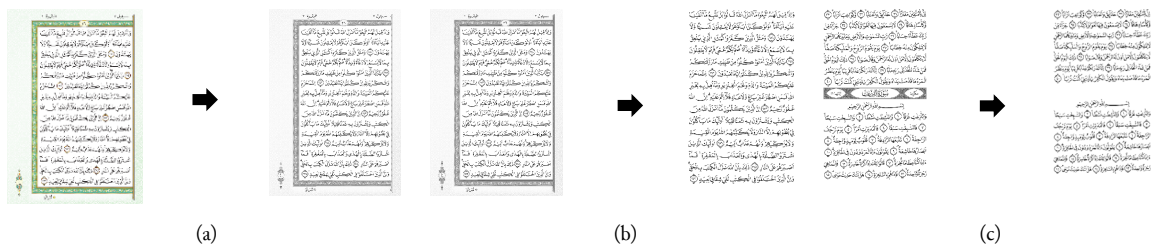


Fig. 2. (a) RGB to gray level conversion results (b) Removing the outer frame (c) Removing the inner frame

Fig 2(a) shows the results of conversion to gray level in the template image, (b) is an example of the results of removing the outer frame, and (c) shows cropping of the inner frame.

3.2. Removing frames and page segmentation

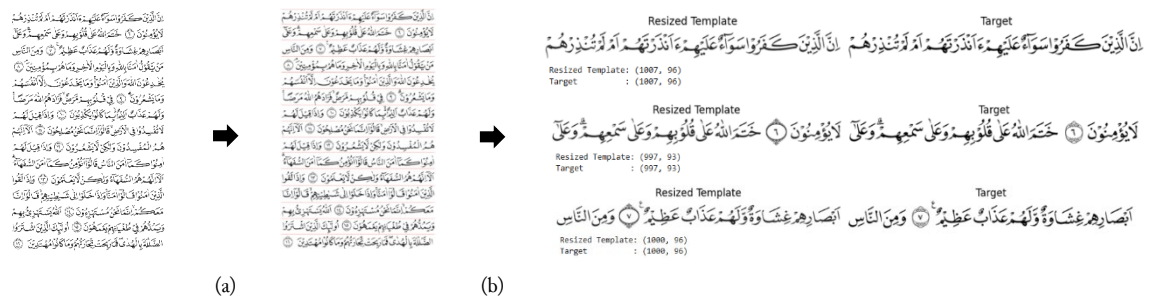


Fig. 3. (a) Page segmentation into rows (b) Alignment of template image size to target image.

Page segmentation into lines is shown in Fig 3(a) and image size adjustment is shown in Fig 3(b). In the *Ustmany* type of *Mushaf* manuscript, one page consists of 15 lines, so page segmentation is necessary to divide the page into 15 lines. Size alignment is performed on the template image relative to the target image, as the target image is smaller.

3.3. Segmentation of lines into blocks

Line segmentation into blocks is shown in Fig 4. This process aims to segment lines into blocks. One page comprises 15 lines; thus, there are 75 blocks in the template and target images.

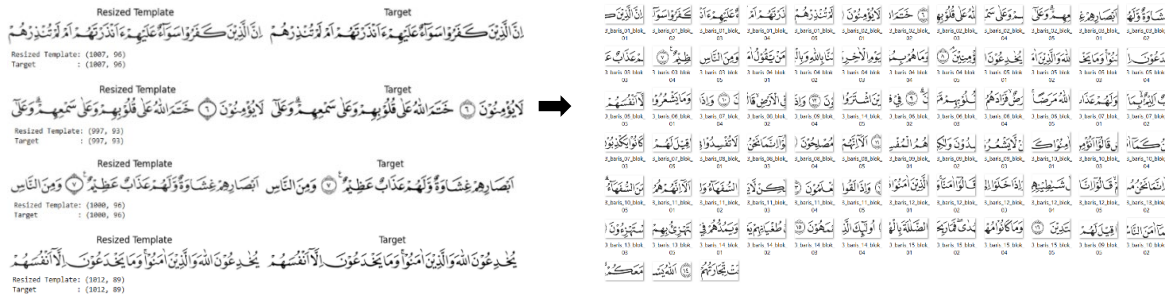


Fig. 4. Results of row to block segmentation

3.4. Alignment of block image positions

The alignment stage between the template block image and the target is shown in Fig 5, because even though they look the same, their positions are different. This will cause matching errors. Therefore, it is necessary to align the positions of both block images.



Fig. 5. Position alignment process of block images

3.5. Masking

An example of the masking results is shown in Fig 6. This process aims to obscure the verse-numbering frame. This is necessary because the verse numbering frame of each Mushaf manuscript has a different design. Therefore, it is necessary to mask these objects to avoid requiring matching.



Fig. 6. Position alignment process of block images

3.6. Pixel Analysis

The pixel value differences between the template and target block images are shown in Fig. 7. The results of the pixel difference calculation for the two images are presented in column 1. The figure explains the calculation of the pixel difference between the image on page 54, row 5, column 1. The block image consists of 201 columns and 103 rows.

An example of similarity calculation results using ssim and image mismatch analysis results is shown in Fig 8. If the SSIM value is ≥ 0.9 , the two images have structural similarities. The analysis was performed on page 100, rows 1 to 15, where each row consists of image blocks 1 to 5.

After determining the similarity between two block images, pixel analysis is performed by computing pixel difference values between them. The results of the pixel difference calculations on the block images will be analyzed further by looking at the columns and rows that have pixel difference values ≥ 200 if the threshold value occurs at the limit of 5 columns and 5 rows in succession, with a distance between them = 1. In Fig 9, there is a block image that shows a mismatch, as seen in the difference in punctuation marks in the blue box area. The details of the difference can be seen through the visualization of the pixel difference values in the block image shown in Fig 10.

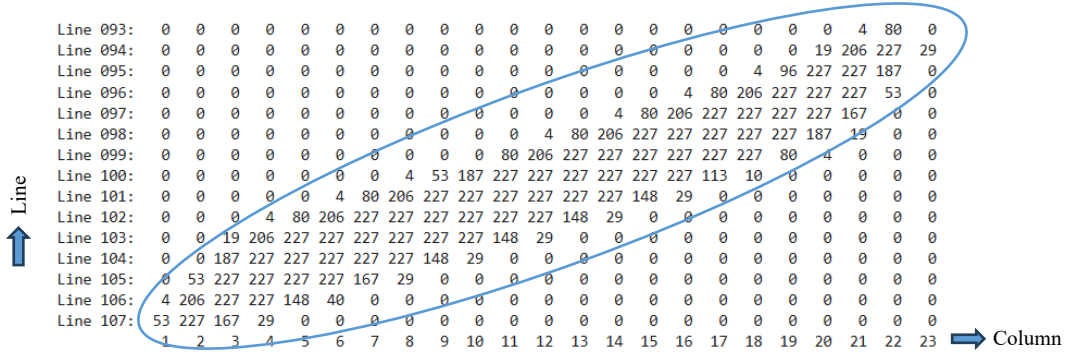


Fig. 10. Visualization of pixel locations on block images

From Fig 9 and Fig 10, further analysis is needed to determine whether the block image meets the criteria for writing inconsistency. The pixel difference threshold of ≥ 200 occurs consecutively in columns 3 to 15 and rows 99 to 104 with a distance of 1. The results of this analysis can be seen in Table 1.

Table 1. Analysis of image mismatch results on page 54, row 5

| Block 5 | Pixel Difference Detection Results ≥ 200 (column, row) | Position Mismatch (row, column) |
|---------------------|--|---------------------------------|
| عَاللهِ لَا يُحِبُّ | 3: 104, | 3: 104, |
| | 4: 104, | 4: 104, |
| | 5: 103, | 5: 103, |
| | 6: 103, | 6: 103, |
| | 7: 102, | 7: 102, |
| | 8: 102, | 8: 102, |
| | 9: 101, | 9: 101, |
| | 10: 101, | 10: 101, |
| | 11: 100, | 11: 100, |
| | 12: 100, | 12: 100, |
| عَاللهِ لَا يُحِبُّ | 12: 99, | 12: 99, |
| | 14: 99, | 14: 99, |
| | 15: 98, | 15: 98, |
| | 17: 97, | 17: 97, |
| | 19: 96, | 19: 96, |

On page 54, line 5, block 1, an image mismatch was found, indicating a writing error. The program identified the writing error as follows: columns 3 to 19 of the target image showed a positional mismatch from position 98 to 104.

Image mismatch detection also occurred on page 54, line 6, block 5, as shown in Fig. 11. The visual dissimilarity is marked with a blue box. The area where the pixel difference occurred is shown in Fig. 12.

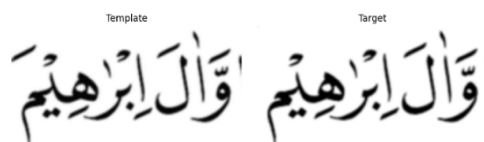


Fig. 11. Writing mismatch detection

Fig. 11 shows areas with indications of discrepancies in writing, as pixel differences exceeding the threshold occur in rows 18 to 31 and columns 1 to 22, indicated by blue circles. However, further analysis is needed to determine whether these meet the criteria for indicating writing errors. The pixel analysis of the block image in Fig. 11 is shown in Fig. 12. The blue-circled region comprises pixels in rows 17 to 31 and columns 1 to 22. This area meets the criteria for a mismatch in the block image on the 5th

Image mismatch detection also occurs on page 110, line 6, block 5, as shown in Fig. 14. The visual dissimilarity is marked with a blue box, indicating a shift of two points on the waqaf. The visualization of the area where the pixel difference occurs is shown in Fig. 15.

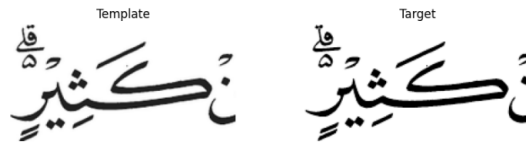


Fig. 14. Writing error detection

Fig 14 shows the area where differences in writing are indicated, with the total difference in pixel values located around that area, as shown in Fig 15.

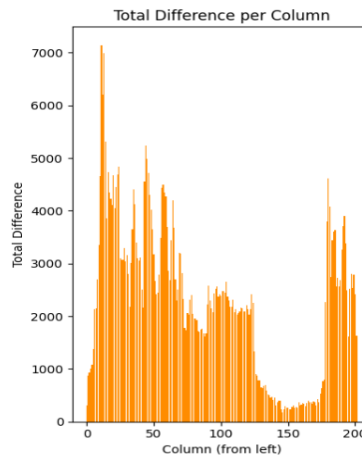


Fig. 15. Total number of pixel value differences per column

An example of a block image that meets the writing error criteria has already been described. The following is an example of the results of writing analysis on a block image that matches the template image and the target shown in Fig. 16, where the ssim value shows ≥ 0.9 , there is an area that has a pixel value difference of ≥ 200 , but is not located in 5 consecutive columns and is not in adjacent rows.

| | | |
|--|---|---|
| | ➔ | SSIM Score (refined): 0.9694 ✳ Pixel Difference Detection Results ≥ 200 : Column 34: line 97, Column 51: line 92, ✖ [UNDETECTED] No 5 consecutive columns found. |
| | ➔ | SSIM Score (refined): 0.9730 ✳ Pixel Difference Detection Results ≥ 200 : Column 200: line72-74, line 80, ✖ [UNDETECTED] No 5 consecutive columns found. |
| | ➔ | SSIM Score (refined): 0.9630 ✳ Pixel Difference Detection Results ≥ 200 : ✖ [UNDETECTED] No 5 consecutive columns found. |
| | ➔ | SSIM Score (refined): 0.9759 ✳ Pixel Difference Detection Results ≥ 200 : ✖ [UNDETECTED] No 5 consecutive columns found. |

Fig. 16. Image of blocks that do not meet the criteria for writing errors

The analysis was conducted on three juz of a manuscript submitted to the publisher and on randomly selected block images, particularly those identified as incorrect by the proofreader. To date, the block images declared incorrect have been identified as incorrect, and the correct block image samples have been identified as correct.

3.7. Efficiency of manuscript image verification

The evaluation was conducted using the start and stop functions to measure the compilation time [41] of the image mismatch analysis algorithm. The test was conducted on several manuscript pages, and the results are presented in Table 3.

Table 3. Image matching compilation time

| Number of Pages | Page | Runtime | |
|-----------------|--|-------------|--------|
| | | millisecond | second |
| 1 | 42 | 1952 | 1.95 |
| 5 | 42, 43, 44, 45, 46 | 9525 | 9.53 |
| 10 | 42, 43, 44, 45, 46, 47, 48, 49, 50, 51 | 20193 | 20.19 |
| 20 | 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61 | 45490 | 45.49 |

The trial was conducted on 1, 5, 10, and 20 pages of the Mushaf. One juz consists of 20 pages, so for 30 juz, it can be estimated that it will take 45.49 seconds * 30 = 1,364.7 seconds, which is equivalent to 22.7 minutes. Currently, proofreaders require 30 working days to examine and analyze the 30 juz of the Quran manuscript without translation. The application of technology can accelerate the proofreading process and assist proofreaders by providing recommendations for pages, lines, and positions that meet criteria for writing inconsistencies.

Accuracy is measured as the proportion of analyses that are correct, calculated from the counts of true positives, true negatives, false positives, and false negatives. The data analysis was carried out on 4 pages, with no writing errors: pages 26, 54, 110, and 330. The total number of blocks across 4 pages is 300. The image of not matched detected not matched (TN) appears on page 54, line 5, block 1; page 110, line 6, block 1; and page 330, line 6, block 5. The image of matched detected (TP) is on 4 pages, minus 3 blocks; obtained 297, with TN and TP values, an accuracy of 100% is obtained.

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \quad (14)$$

$$\begin{aligned} Accuracy &= \frac{297+3}{297+0+3+0} \times 100\% \\ &= \frac{300}{300} \times 100\% = 100\% \end{aligned}$$

4. Conclusion

Image matching in the Uthmanic Standard *Mushaf* Al-Qur'an manuscript, using pixel-matching techniques, can be employed to analyze writing errors. With this approach, the analysis process has been accelerated, and publisher manuscripts can be checked against LPMQ-owned manuscripts. The implementation of the matching technique involves comparing the template block image and the target image based on pixel difference values with a predetermined threshold, applied to five consecutive columns and rows adjacent to each other with a distance of 1, and measuring structural similarity using the SSIM technique, thereby enabling the detection of writing errors in the block image. Analyzing 30 juz takes 1,364.7 seconds or approximately 22.7 minutes. This is faster than the manual analysis process, which takes 30 working days. The image-mismatch analysis stage in the *Mushaf* writing manuscript successfully identified writing errors and indicated the positions of mismatches in the incorrect data test, and correctly identified that the template and target block images matched in the correct data test. For further research, it would be beneficial to analyze block image data across all pages of the *Mushaf* and other Mushaf manuscripts, and to compare these results using different image preprocessing techniques.

Declarations

Author contribution. All authors contributed equally.

Funding statement. This research was funded by a postgraduate research grant from the Ministry of Education, Culture, Research, and Technology in 2024.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

- [1] J. Li, "Research on Multimedia Information Processing System based on Data Mining Technology," in *2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS)*, IEEE, Jul. 2023, pp. 877–883. doi: [10.1109/ISCTIS58954.2023.10213025](https://doi.org/10.1109/ISCTIS58954.2023.10213025).
- [2] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 91–99, 2022, doi: [10.1016/j.gltp.2022.04.020](https://doi.org/10.1016/j.gltp.2022.04.020).
- [3] A. Tuppada and S. D. Patil, "Data Pre-processing Issues in Medical Data Classification," in *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)*, IEEE, Sep. 2023, pp. 1–6. doi: [10.1109/NMITCON58196.2023.10275855](https://doi.org/10.1109/NMITCON58196.2023.10275855).
- [4] X. He, Y. Chen, Y. Fan, X. Kong, and Z. Cai, "A Reversible Multimedia Representation Method and Its Applications in Multimedia Processing," *IEEE Access*, vol. 11, no. May, pp. 62436–62448, 2023, doi: [10.1109/ACCESS.2023.3288020](https://doi.org/10.1109/ACCESS.2023.3288020).
- [5] Z. Abidin, "Mengetahui Spesifikasi Tiga Mushaf Al Quran Standar Indonesia," 2020. Available at: <https://tafsirquran.id/mengetahui-spesifikasi-tiga-Mushaf-al-quran-standar-indonesia/>
- [6] F. Hoerulloh, D. T. Murdiansyah, and K. Lhaksmana, "Juz Classification of English-Translated Qur'anic Verses Using Support Vector Machine," in *2025 IEEE International Conference on Artificial Intelligence for Learning and Optimization (ICoAILO)*, 2025, pp. 372–377. doi: [10.1109/ICoAILO66760.2025.11156072](https://doi.org/10.1109/ICoAILO66760.2025.11156072).
- [7] LPMQ, "Lajnah Pentashihan Mushaf Al-Qur'an Kementerian Agama RI," 2023. Available at: [https://lajnah.kemenag.go.id/info-lpmq/#:~:text=Lajnah Pentashihan Mushaf Al-Qur'an \(LPMQ\) adalah,hasil kajian LPMQ%3B dan pengelolaan](https://lajnah.kemenag.go.id/info-lpmq/#:~:text=Lajnah Pentashihan Mushaf Al-Qur'an (LPMQ) adalah,hasil kajian LPMQ%3B dan pengelolaan)
- [8] F. Lukman, *The Official Indonesian Qur'an Translation: The History and Politics of Al-Qur'an dan Terjemahnya*. Cambridge: Open Book Publishers, 2022. doi: [10.11647/OBP.0289](https://doi.org/10.11647/OBP.0289).
- [9] M. Hanafi et al., "Evaluasi Kepatuhan Penerbit Al-Qur'an terhadap Mekanisme Penerbitan, Pentashihan, dan Peredaran Mushaf Al-Qur'an di Indonesia," *Subuf*, vol. 14, no. 2, pp. 245–263, 2021, doi: [10.22548/shf.v14i2.691](https://doi.org/10.22548/shf.v14i2.691).
- [10] T. S. Arulananth, P. Chinnasamy, J. C. Babu, A. Kiran, J. Hemalatha, and M. Abbas, "Edge detection using fast pixel based matching and contours mapping algorithms," *PLoS One*, vol. 18, no. 8, p. e0289823, 2023, doi: [10.1371/journal.pone.0289823](https://doi.org/10.1371/journal.pone.0289823).
- [11] Z. Ding, S. Qian, Y. Li, and Z. Li, "An image matching method based on the analysis of grey correlation degree and feature points," in *NAECON 2014 - IEEE National Aerospace and Electronics Conference*, IEEE, Jun. 2014, pp. 157–162. doi: [10.1109/NAECON.2014.7045795](https://doi.org/10.1109/NAECON.2014.7045795).
- [12] T. Lu, "Improving Image Matching Accuracy with Keypoint Features and Feature Fusion," in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, IEEE, Nov. 2021, pp. 513–520. doi: [10.1109/ICFTIC54370.2021.9647145](https://doi.org/10.1109/ICFTIC54370.2021.9647145).
- [13] R. Gonzalez and R. E. Woods, *Digital Image Processing 3rd Edition*, 3rd ed. 2008. [Online]. Available at: https://sde.uoc.ac.in/sites/default/files/sde_videos/Digital Image Processing 3rd ed. - R. Gonzalez, R. Woods-ilovepdf-compressed.pdf
- [14] I. Kesen, J. F. Lotz, I. Ziegler, P. Rust, and D. Elliott, "Multilingual Pretraining for Pixel Language Models," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2025, pp. 29582–29599. doi: [10.18653/v1/2025.emnlp-main.1504](https://doi.org/10.18653/v1/2025.emnlp-main.1504).

- [15] R. C. Gonzalez, R. E. Woods, and P. Hall, *Digital Image Processing*. Pearson Education India, 2001. Available at: [Google Scholar](#).
- [16] Q. Huang, X. Guo, Y. Wang, H. Sun, and L. Yang, "A survey of feature matching methods," *IET Image Process.*, vol. 18, no. 6, pp. 1385–1410, May 2024, doi: [10.1049/ipr2.13032](#).
- [17] S. Verykokou and C. Ioannidis, "Image Matching: A Comprehensive Overview of Conventional and Learning-Based Methods," *Encyclopedia*, vol. 5, no. 1, p. 4, Jan. 2025, doi: [10.3390/encyclopedia5010004](#).
- [18] J. Mukherjee, "A Deep Neural Network Based Holistic Approach for Optical Character Recognition of Handwritten Documents," *SN Comput. Sci.*, vol. 5, no. 4, p. 347, 2024, doi: [10.1007/s42979-024-02713-8](#).
- [19] R. Szeliski, *Computer Vision*, 2nd ed. in Texts in Computer Science. Cham: Springer International Publishing, 2022. doi: [10.1007/978-3-030-34372-9](#).
- [20] S. Ji, C. Zeng, Y. Zhang, and Y. Duan, "An evaluation of conventional and deep learning-based image-matching methods on diverse datasets," *Photogramm. Rec.*, vol. 38, no. 182, pp. 137–159, 2023, doi: [10.1111/phor.12445](#).
- [21] W. Dang et al., "A Feature Matching Method based on the Convolutional Neural Network.," *J. Imaging Sci. Technol.*, vol. 67, no. 3, 2023, doi: [10.2352/J.ImagingSci.Technol.2023.67.3.030402](#).
- [22] L. C. Kiew, A. J. M. Muzahid, and S. F. Kamarulzaman, "Vehicle Route Tracking System based on Vehicle Registration Number Recognition using Template Matching Algorithm," in *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, 2021, pp. 249–254. doi: [10.1109/ICSECS52883.2021.00052](#)
- [23] J. Xu, C. Yang, and J. Chen, "A customized template matching classification system," *J. Korean Phys. Soc.*, vol. 85, no. 10, pp. 867–882, 2024, doi: [10.1007/s40042-024-01182-9](#).
- [24] Z. Osman, L. Hamandi, R. Zantout, and F. N. Sibai, "Automatic processing of Arabic text," in *2009 International Conference on Innovations in Information Technology (IIT)*, IEEE, Dec. 2009, pp. 140–144. doi: [10.1109/IIT.2009.5413793](#).
- [25] A. K. J. Saudagar, H. V. Mohammed, K. Iqbal, and Y. J. Gyani, "Efficient Arabic text extraction and recognition using thinning and dataset comparison technique," in *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, IEEE, Jan. 2015, pp. 1–5. doi: [10.1109/ICCICT.2015.7045725](#).
- [26] T. Faisal and S. AlMaadeed, "Enabling indexing and retrieval of historical Arabic manuscripts through template matching based word spotting," in *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, IEEE, Apr. 2017, pp. 57–63. doi: [10.1109/ASAR.2017.8067760](#).
- [27] N. Zaghden, R. Mullot, and A. M. Alimi, "Characterization of ancient document images composed by Arabic and Latin scripts," in *2011 International Conference on Innovations in Information Technology*, IEEE, Apr. 2011, pp. 124–127. doi: [10.1109/INNOVATIONS.2011.5893801](#).
- [28] S. Alghyaline, "Arabic Optical Character Recognition: A Review," *Comput. Model. Eng. Sci.*, vol. 135, no. 3, pp. 1825–1861, 2023, doi: [10.32604/cmcs.2022.024555](#).
- [29] F. A. Abdul-Aziz, Q. M. Hussein, and M. S. Al-Taei, "Arabic Printed Word Recognition For Encrypted Document Management," 2023, Thess, Tikrit University. available at: [Google Scholar](#).
- [30] R. Patgiri, S. Nayak, T. Akutota, and B. Paul, "Machine Learning: A Dark Side of Cancer Computing," *Int'l Conf. Bioinforma. Comput. Biol.*, pp. 92–98, 2019, doi: [10.48550/arXiv.1903.07167](#).
- [31] L. Myllyaho, M. Raatikainen, T. Männistö, J. K. Nurminen, and T. Mikkonen, "On misbehaviour and fault tolerance in machine learning systems," *J. Syst. Softw.*, vol. 183, no. January, p. 111096, Jan. 2022, doi: [10.1016/j.jss.2021.111096](#).
- [32] Yuhandri, S. Madenda, E. P. Wibowo, and - Karmilasari, "Object Feature Extraction of Songket Image Using Chain Code Algorithm," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 1, p. 235, Feb. 2017, doi: [10.18517/ijaseit.7.1.1479](#).

- [33] C. Wibawa and D. T. Anggraeni, "Comparison Of Image Segmentation Method In Image Character Extraction Preprocessing Using Optical Character Recognition," *J. Tek. Inform.*, vol. 4, no. 3, pp. 583–589, Jun. 2023, doi: [10.52436/1.jutif.2023.4.3.956](https://doi.org/10.52436/1.jutif.2023.4.3.956).
- [34] I. Rabaev and M. Litvak, "Recent advances in text line segmentation and baseline detection in historical document images: a systematic review," *Int. J. Doc. Anal. Recognit.*, May 2025, doi: [10.1007/s10032-025-00526-w](https://doi.org/10.1007/s10032-025-00526-w).
- [35] Y. Deng, X. Hu, D. Teng, B. Li, C. Zhang, and W. Hu, "Dynamic adjustment of hyperparameters for anchor-based detection of objects with large image size differences," *Pattern Recognit. Lett.*, vol. 167, pp. 196–203, 2023, doi: [10.1016/j.patrec.2023.02.019](https://doi.org/10.1016/j.patrec.2023.02.019).
- [36] K. Lee, S. Lee, and K. M. Lee, "Auto-regressive transformation for image alignment," pp. 1–16, 2025, doi: [10.48550/arXiv.2505.04864](https://doi.org/10.48550/arXiv.2505.04864).
- [37] W. Song and I. Deng, "A Hybrid Architecture Combining CNN, LSTM, and Attention Mechanisms for Automatic Speech Recognition," in *2025 11th International Conference on Computing and Artificial Intelligence (ICCAI)*, 2025, pp. 285–292. doi: [10.1109/ICCAI66501.2025.00052](https://doi.org/10.1109/ICCAI66501.2025.00052).
- [38] M. R. Raigonda and Shweta, "Signature Verification System Using SSIM In Image Processing," *J. Sci. Res. Technol.*, vol. 2, no. 1, pp. 5–11, Jan. 2024, doi: [10.61808/jsrt79](https://doi.org/10.61808/jsrt79).
- [39] Z. Tang *et al.*, "A Fast Image Stitching Algorithm Based on Texture Classification and Improved SIFT," *IEEE Access*, vol. 12, no. August, pp. 124183–124208, 2024, doi: [10.1109/ACCESS.2024.3443111](https://doi.org/10.1109/ACCESS.2024.3443111).
- [40] V. Mudeng, M. Kim, and S. Choe, "Prospects of Structural Similarity Index for Medical Image Analysis," *Appl. Sci.*, vol. 12, no. 8, p. 3754, Apr. 2022, doi: [10.3390/app12083754](https://doi.org/10.3390/app12083754).
- [41] L. K. Singh, M. Khanna, H. Garg, and R. Singh, "Emperor penguin optimization algorithm-and bacterial foraging optimization algorithm-based novel feature selection approach for glaucoma classification from fundus images," *Soft Comput.*, vol. 28, no. 3, pp. 2431–2467, 2024, doi: [10.1007/s00500-023-08449-6](https://doi.org/10.1007/s00500-023-08449-6).