

Automated fracture aperture quantification in rock outcrops using image processing technique: The OpenFraQ algorithm

Maryam Esmaeili ^{*}, Tommaso Beni, Giovanni Gigli

Department of Earth Sciences, University of Firenze, Via La Pira 4, 50121 Firenze, Italy

ARTICLE INFO

Keywords:

Fracture aperture quantification
Rock mass characterisation
Contour-based segmentation
Automated fracture trace extraction
OpenFraQ algorithm

ABSTRACT

Understanding and quantifying rock fractures is one of the main goals in mining, geotechnical engineering, and engineering geological assessment. This study presents OpenFraQ, an automated image processing algorithm designed to extract and analyse open fracture aperture and relative geometries from 2D images. Unlike previous automatic or semi-automatic image processing approaches that only represent fractures as trace lines or segmented points, OpenFraQ introduces a novel geometric method that characterises fractures as both contour areas and overall traces, allowing discrimination between open fractures and non-open cracks. Using contour-based segmentation, edge detection, and width estimation methods, OpenFraQ effectively detects complex and irregular fracture traces, providing reliable estimates of the fracture aperture and tracing geometry. Validation against manual tracing and Hough transform confirms its accuracy and efficiency, particularly in processing large datasets and challenging discontinuity networks. Across test samples, OpenFraQ achieved an average precision of 63.34%, a recall of 63.05%, and an F1-score of 62.92%. The algorithm offers a practical tool for rapid geometrical fracture characterisation, enhancing our understanding of fracture networks, which serves as the starting point for further engineering geological analyses, such as stability assessment, stress and strain distribution, and rock mass permeability evaluation.

1. Introduction

Fractures are among the most pervasive discontinuities in the Earth's crust, strongly influencing rock mass stability, deformation, and fluid flow. They primarily result from natural stresses, including tectonic forces, pressure variations, thermal effects, and changes in fluid pressure [1–4]. The spatial distribution of fractures indicates clustering, which is vital for DFN (Discrete Fracture Network) models and numerical simulations [5]. Mechanically, the aperture refers to the physical gap between the walls of the fracture. Hydraulically, it is defined as the effective open portion of the aperture that allows fluid flow, reflecting the roughness characteristics of the fracture [6]. Rock fractures can be identified as traces on surfaces such as outcrops, cores or mine walls [7]. Surveying rock mass DFN helps understanding deformation sequences, rock rheology, fluid flow, geomechanical properties and fracture-structural relationships [8,9].

In general, apertures fall into one of three categories: geometric, mechanical and hydraulic [10,11]. According to the literature [3, 11–14], the examination of fractures and voids in rocks can be

categorised into two primary areas: laboratory investigations and field assessments. These approaches may be either destructive or non-destructive, depending on whether they modify the visible part of the fracture by methods like resin injection under ultraviolet (UV) light [11,15] or removal of weathered material [12] or not. All these methods can be supplemented by analysing images of the samples or the samples themselves for final measurements. Additionally, rock fracture imaging encompasses various methods such as optical imaging, laser scanning, CT (Computed Tomography) scanning, radar, ultrasonic, and infrared techniques. Integrating multiple methods enhances the accuracy of fracture mapping [15,16]. Also, destructive, precise techniques such as resin injections, which are useful in laboratory settings for studying rock cores, are not applicable in field investigations [17,18]. Furthermore, features that are regarded as noise, like tiny cracks, in slope studies, may still provide important information when analysing core samples. Discontinuities or cracks visible on the surface that are insignificant in large outcrop studies may be critical for laboratory core analysis [3]. Similarly, precise shape recognition, which is vital in hydromechanics applications, might be less relevant for slope stability assessments.

^{*} Corresponding author.

E-mail address: maryam.esmaeili@unifi.it (M. Esmaeili).

<https://doi.org/10.1016/j.rineng.2026.109043>

Received 18 September 2025; Received in revised form 7 January 2026; Accepted 7 January 2026

Available online 9 January 2026

2590-1230/© 2026 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Considering these facts helps in identifying and removing noise from the results in fracture geometric studies. Furthermore, manual methods are susceptible to human error, require significant time, and are often impractical in areas that are difficult to access [19].

Zhao & Wang (2006) [4] developed a new method for accurately measuring rock fracture aperture using digital image processing. Their method involves dividing the fracture aperture into segments based on corner points using a corner segment algorithm and then measuring the width of each segment using the best-fit rectangle method. The study by Xu et al. (2021) [12] utilised the measuring ruler dispersion-tangent middle axis (MRD-TMA) method to accurately measure fracture apertures in fresh granite, eliminating weathered surface effects destructively. While image processing assists in preprocessing, the measurements are still performed manually, making the process semi-automatic. In another laboratory study, resin-injected rock samples were analysed under UV light using fractal theories and image processing techniques, including thresholding, Steger's line detection, and Sobel's edge detection. The study developed a fully automatic algorithm, but its application is limited to controlled laboratory conditions [11]. Despite significant advancements in machine learning, especially deep learning, obtaining a comprehensive training dataset that captures the complexity and diversity of various fractures remains challenging. A dataset that yields good results in one scenario may fail in another with different rock types and stone textures [14]. Furthermore, in some cases, a sufficiently large and representative data set may not be available at all. In such scenarios, image processing-based methods can be highly effective and essential, especially in hard-to-access areas. Obtaining characteristics of fractures and their apertures through image processing and photogrammetry techniques offers a less time-consuming and more cost-effective alternative [20]. Techniques involving image processing, including binarisation, segmentation, clustering, and border detection, have been employed to identify fracture shapes [21]. In some cases, these detected shapes are segmented, vectorised, and post-processed for detailed analysis across various environments [22]. The choice between modelling the exact fracture shape [23] or representing it as a simpler form (such as a trace vector or occupied area) depends on the specific investigation objectives.

This paper specifically develops the Open Fracture Quantification algorithm, OpenFraQ, for the automatic characterisation of open fractures using imaging. Employing a geometric approach proves particularly effective in hard-to-access regions across various fields such as mining, geohazard assessment, and geoengineering projects. While some existing methods focus on analysing individual and simple fractures, the presented algorithm simultaneously detects multi-branch, multifracture, and irregular shapes, evaluating their properties and ranking them based on area and aperture width. The workflow is fully automated, remote, and inherently non-destructive, generating quantitative open fracture features to model fracture aperture as contours (openness) and fracture traces as lines or polylines for further analysis. This approach emphasises geometric analysis, capturing the overall fracture trace, shape, and width estimation through advanced image processing techniques. To the best of the authors' knowledge, the combined use of the bounding box technique for fracture extraction and characterisation, together with overall trace detection, is not widely discussed in the existing literature.

Unlike previous automatic or semi-automatic image processing approaches that represent fractures only as trace lines [14,24–26] or segmented points through machine learning methods [16,27–29] OpenFraQ introduces a novel geometric method that characterises fractures both as contour areas and as overall traces. This enables discrimination between open and non-open cracks, as well as identification of fracture boundaries, fracture areas, and their overall traces. This dual representation enables the robust quantification of complex, irregular fracture openings, which extend beyond simple linear features prevalent in natural rock masses and are challenging to characterise. Byun et al. (2021) [14] developed a CNN (Convolutional Neural

Network) approach for rock fracture segmentation, achieving a precision of 0.532, a recall of 0.614, and an F1-score of 0.570. While direct comparison with other methods may not be exact due to differences in datasets and evaluation criteria, these results highlight that fracture detection in complicated geological and geotechnical settings remain a challenging task. OpenFraQ advances the field by improving validation parameters and enabling efficient processing of high-resolution 2D images, with compatibility for projected or orthorectified images from 3D models [27,29], allowing results to be overlain or integrated with three-dimensional fracture trace maps [30].

The validity of the approach is confirmed both qualitatively and quantitatively, using visual inspection, manual assessments, and a modified Hough transform method, a widely used image processing tool for detecting fractures and their geometries [22,31,32]. However, further improvements are still needed to address challenges posed by severe weathering, the presence of vegetation, and other sources of significant noise that are discussed in the methodology section. Since this algorithm is designed to characterise open fractures, it may not detect fractures that lack visible evidence of aperture or erosion. These less prominent fractures are considered less significant for the survey's purpose compared to open or eroded fractures. The paper assesses the developed method using a publicly available dataset (Byun et al., 2021) for fracture detection problems, thereby minimising the bias that can arise from selecting ideal or proprietary datasets, as well as one real outcrop image, with a concentration on openness to further validate its practical applicability.

2. Materials and Methods

This section details the dataset used and the methodology implemented for defining open fractures in 2D images, as well as the approach employed to validate the results obtained from the OpenFraQ algorithm.

2.1. Materials

A subset of a publicly available dataset (<https://github.com/Montherapy/Rock-fracture-segmentation-with-Tensorflow>) from Byun et al. (2021) [14] was used for evaluation. From this dataset, images without shadows or lower vegetables are selected. While it is possible to mask vegetables or reduce shadow effects using image processing techniques, doing so may introduce noise and impact the results. In addition to the benchmark dataset, one real outcrop image with visible open fractures, collected from an abandoned stone quarry in the Tuscany region of Italy, was included to further validate the method's practical applicability. The algorithm was implemented and executed in MATLAB R2024b [33].

2.2. The OpenFraQ algorithm

OpenFraQ is engineered to analyse and visualise the apertures of rock fractures and their traces using 2D input images through a fully automated workflow. Open fractures are defined as distinct, continuous features on a rock surface with an aperture width exceeding a user-defined threshold, recognised by significant differences in pixel intensity. User-adjustable parameters, such as sigma and Canny thresholds, along with optional preprocessing, are employed to exclude weathered or ambiguous edges, ensuring consistent identification of physically open fractures. The process begins by loading an image, which is then converted to greyscale. To minimise noise in the interval image $I(x, y)$, Gaussian smoothing, a linear filtering technique using a Gaussian kernel $G(x)$ (Eq. 1), is applied, where σ is the standard deviation [34,35].

$$G(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 - y^2}{\sigma^2}\right) \quad (1)$$

Edge detection follows, using the gradient magnitude (Eq. 2), which

highlights intensity changes. This is performed using the Canny algorithm [36], which allows fine-tuning through user-defined thresholds. Specifically, the Canny method employs two thresholds: a lower threshold m_1 to identify potential edges, and a higher threshold m_2 to define strong edges [37]. This two-step thresholding process, known as hysteresis thresholding, adjusts the sensitivity of edge detection, emphasising significant features while reducing noise. The resulting output is a binary image where pixels are classified as foreground (1) or background (0) (Eq. 3).

$$\text{Gradient magnitude} = \sqrt{\frac{\partial I}{\partial x} + \frac{\partial I}{\partial y}} \quad (2)$$

$$\text{Edge} = \text{Hysteresis threshold}(\text{Gradient Magnitude}(I), m_1, m_2) \quad (3)$$

Edges are enhanced with morphological operations: dilation (expanding boundaries) and closing (filling gaps and smoothing contours) [38,39]. These steps, controlled by the size of the structural element (Se) used in morphological operations, enhance the clarity of fractures prior to contour tracing. This improvement facilitates more accurate and effective detection of fracture boundaries. Open rock fractures are characterised by contour detection within the image, while open fractures are considered as close contours. During the contour extraction process, connected components are labelled to identify individual fracture regions. This connectivity for pixels $M(x, y)$ and $N(x', y')$ can be described in two ways: four-connectivity, where a pixel is connected to another if they share an edge, defined mathematically as (Eq. 4), and 8-connectivity (Eq. 5), where connection occurs if they share an edge and a corner [35]:

$$\{(x', y') | |x' - x| = 1, |y' - y| = 0 \text{ or } |x' - x| = 0, |y' - y| = 1\} \quad (4)$$

$$\{(x', y') | |x' - x| \leq 1, |y' - y| \leq 1 \text{ and } (x', y') \neq (x, y)\} \quad (5)$$

The best-fit rectangle method effectively classifies shapes, such as crushed aggregates, in image analysis. It is simple, rotationally invariant, and accurately captures shape features such as elongation and angularity [40]. The bounding box, which represents the smallest rectangle enclosing a shape or object in two-dimensional space, is calculated for each detected contour [41,42]. This involves determining bounding box widths, which are important for analysing and visualising the shapes and sizes of rock apertures. The contour tracing approach is utilised to delineate the boundaries of labelled components, beginning from a boundary pixel and following the edge until returning to the initial pixel. Mathematically, the binary image can be represented as $I(x, y)$, where $I(x, y) = 1$ for object pixels and $I(x, y) = 0$, for background pixels. Each labelled contour is represented by a cell array as (Eq. 6):

$$B[i] = \{(y1, x1), (y2, x2), \dots, (yj, xj)\} \quad (6)$$

Here, (yj, xj) are the coordinates of the j -th boundary pixel of the i -th contour. This methodology provides valuable geometric information regarding the fractured regions within the image, enabling effective analysis of fracture structures [43–45]. Then, properties of each contour are calculated, including area, perimeter, and bounding box width. The bounding box width is defined as the width of the smallest rectangle that completely encloses the contour. Mathematically, the bounding box and its height and width can be determined by identifying the minimum and maximum x and y coordinates as described in Eq. 7-13:

$$\text{MINx} = \text{MIN}_{i=1}^n (xi) \quad (7)$$

$$\text{MAXx} = \text{MAX}_{i=1}^n (xi) \quad (8)$$

$$\text{MINy} = \text{MIN}_{i=1}^n (yi) \quad (9)$$

$$\text{MAXy} = \text{MAX}_{i=1}^n (yi) \quad (10)$$

$$\text{Bounding Box} = (\text{MINx}, \text{MAXx}, \text{MINy}, \text{MAXy}) \quad (11)$$

$$\text{Width} = \text{MAXx} - \text{MINx} \quad (12)$$

$$\text{Height} = \text{MAXy} - \text{MINy} \quad (13)$$

Contours shorter than a user-defined minimum length (mcl) are discarded. Generating the geometric representation provides a straightforward metric for assessing fracture openness and is particularly useful in applications such as feature extraction and object recognition, as it summarises the contours' size, orientation, and shape. The method involves identifying the two furthest points within each contour using Euclidean distances to determine the longest segment, which serves as the main middle line or trace of the contour, while, if necessary, the midpoint of this segment is also plotted to indicate the simple trace of the fracture. In complex contours, the parameter p specifies the number of middle points within the trace polyline, which represents a better capture of intricate features of the shape. This process transforms initial simple lines into polylines that more effectively depict complex fracture geometries. To localise these p (≥ 0) points, a binary mask isolates each contour from the background, enabling the calculation of pixel intensity variations and the selection of the most relevant points for constructing smooth polylines that accurately represent the contour shape. The resulting polylines, along with attributes such as length and orientation, are stored for further analysis. It is important to note that the objective of the algorithm is not to precisely detect the exact shape of the centerline, as has been the focus in previous research [23]. Instead, the characterising of the open fracture of outcrops uses a simple line as the trace and a basic contour to represent the opening area.

The width measurement is performed using the modified MRD-TMA method proposed by Xu et al. (2021) [12]. This approach employs the normal to the central axis of the fracture to identify boundary intersections and measures the distance between these intersection points to determine the width. Since this study adopts a geometrical rather than hydrogeological perspective, the precise shape of each fracture's central axis is less critical; instead, a general trace derived from OpenFraQ suffices. Consequently, the exact centerline is replaced by the trace line, which considers the trace's position relative to the fracture. Intersection points are accurately identified, and for each contour, only those within the contour's boundary are retained. If no intersection is found in a given direction, a fallback strategy selects the second closest boundary point along that side to ensure measurement continuity. The local width at each sampled point is calculated as the Euclidean distance between the segment point and the boundary intersection points, effectively capturing the aperture size. It is important to note that the trace line may not be completely within the fracture contour. Consequently, the final width value depends on whether intersections are found on one or both sides of the trace and on whether the segment point is inside or outside the boundary. Depending on its position, the width is calculated either as the sum of the intersection lengths or as their absolute difference (Eq. 14):

$$W = |Intersection_{B[i]}^+ \pm Intersection_{B[i]}^-| \quad (14)$$

Accurate visualisation of open fractures relies on carefully tuning parameters to minimise errors. Figs. 1 and 2 illustrate the workflow, pseudocode and parameter settings of the proposed algorithm, respectively. Likewise, the algorithm's parameters, ranges, and operations are summarised as a guide in Table 1 and employed for the tested images in the next session.

2.3. Sensitivity Analysis

A comprehensive sensitivity analysis was performed to evaluate OpenFraQ's robustness in detecting and characterising open fractures. Three key parameters, Canny edge detection thresholds, Gaussian smoothing (δ), and structural element size (Se), were systematically varied, with other settings held constant. Their influence on average contour area, contour length, and contour count was assessed (Fig. 3).

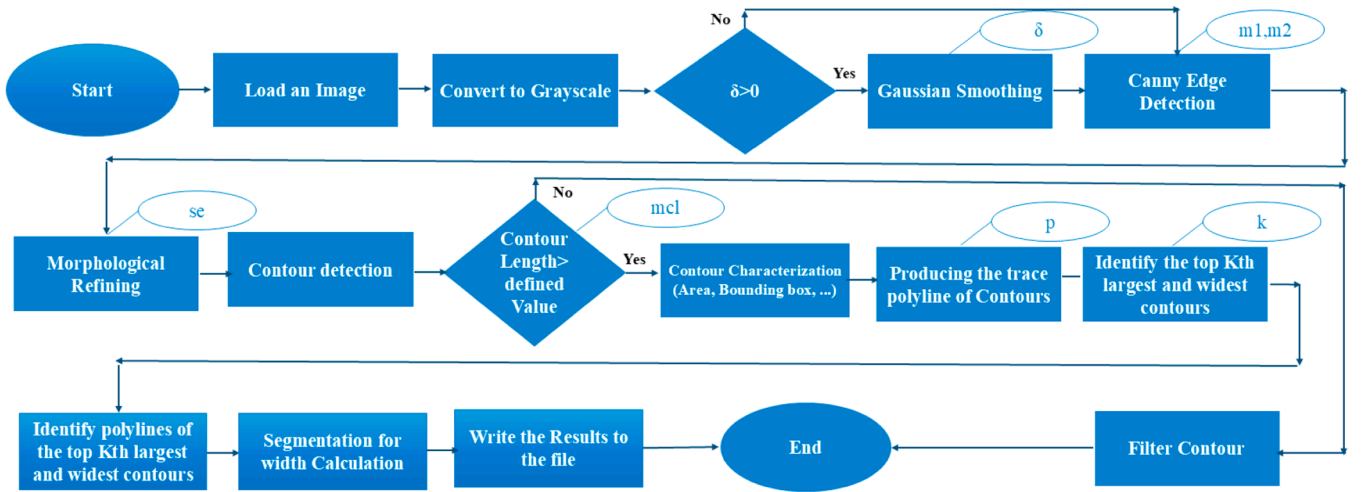


Fig. 1. Flowchart of the presented algorithm.

Input: Image I , parameters: σ , $[m_1, m_2]$, Se , mcl , k , p

Compute polyline (centerline) with p key points

- Preprocess:**
 - $I_{gray} \leftarrow \text{Grayscale}(I)$
 - $I_{smooth} \leftarrow \text{GaussianFilter}(I_{gray}, \sigma)$
 - $E \leftarrow \text{CannyEdges}(I_{smooth}, m_1, m_2)$
 - $E_{morph} \leftarrow \text{MorphologicalOps}(E, Se)$
- Extract contours:**
 - $B \leftarrow \text{FindContours}(E_{morph})$
 - For each contour b in B :
 - If $\text{Length}(b) < mcl$: skip
 - Compute $\text{Area}(b)$, $\text{Perimeter}(b)$, $\text{BoundingBoxWidth}(b)$
- Select top fractures:**
 - $\text{Widest} \leftarrow \text{TopK}(\text{BoundingBoxWidth}, k)$
 - $\text{Largest} \leftarrow \text{TopK}(\text{Area}, k)$
- For selected contours:**
 - 5. **For each polyline:**
 - For each sampled centerline point:
 - Measure aperture width via intersection with contour boundary
 - 6. **Save results:**
 - Export contour/polyline properties, aperture measurements, and figures

Output: Feature tables, images, statistics

Fig. 2. Pseudocode of the presented algorithm.

Table 1
Algorithm's parameters, ranges and operation.

Parameter	Definition	Possible Range	Suggested range	Effect of Increasing Values	Effect of Decreasing Values
δ	Gaussian smoothing standard deviation	$[0, \infty)$	$[2, 6]$	Smoother images, fewer small details	Less smoothing, more noise, finer details
$[m_1, m_2]$	Canny edge detection thresholds	$[0, 1]$	$[0.1, 0.5]$	Fewer edges (more conservative detection)	More edges (sensitive detection)
se	Structural element size for morphological operations	$[0, \infty)$	$[1, 20]$	Larger: more smoothing and gap filling	Smaller: preserve details, less smoothing
mcl	Minimum number of points for a contour to be considered	$[1, \infty)$	$[1, 100]$	Larger: filter out small contours, focus on major fractures	Smaller: include minor features, possibly noise
k	Number of top contours to analyse by width and area	$[1, \text{Number of contours}]$	$[1, 20]$	Include more contours and a broader fracture set	Include fewer contours, highlighted fracture set
p	Number of top points selected for polylines	$[1, \text{length of contour points}]$	Depending on the fracture shape	More points: detailed polylines	Fewer points: more abstract, simplified lines

Increasing Canny thresholds (with $\delta = 2$, $Se = 10$) led to fewer and shorter detected contours, while average contour area remained relatively stable, indicating robustness to this parameter (Fig. 3A–C). Adjusting structural element size ($\delta = 2$, thresholds = $[0.11, 0.15]$) increased contour area and length but decreased contour count, as larger elements merged separate fractures into continuous features (Fig. 3D–F). Varying sigma ($Se = 10$, canny thresholds = $[0.11, 0.15]$) initially increased average contour area by merging small features,

peaking before excessive smoothing reduced both area and detail (Fig. 3G–I). This peak highlights the optimal trade-off between noise reduction and feature preservation, informing parameter selection for diverse geological settings.

OpenFraQ utilises advanced image processing techniques, including Gaussian smoothing, adaptive edge detection, and morphological operations, to minimise noise and enhance the accuracy of fracture detection, even under challenging conditions such as weathered surfaces

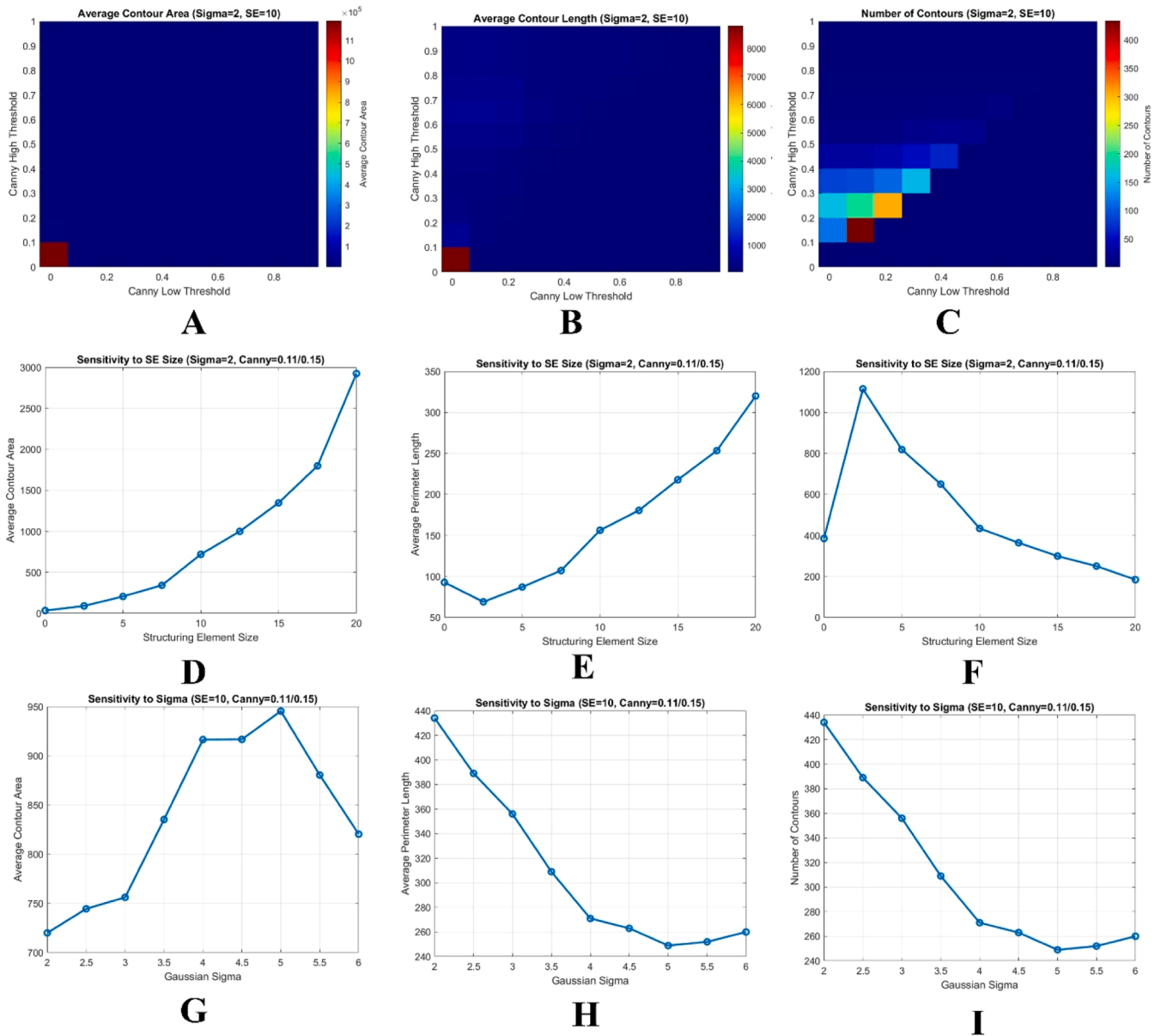


Fig. 3. Sensitivity Analysis of OpenFraQ Output Metrics to Key Parameters: Effects of 1) Canny Thresholds (A–C), 2) Structural Element Size (D–F), and 3) Sigma (G–I) on 1) average contour area, 2) average contour length, and 3) number of contours, respectively.

and variable image quality. Its adjustable parameters allow the system to be tailored to diverse scenarios, and sensitivity analyses demonstrate consistent performance across common variations. Nevertheless, in situations involving severe noise, shadowing, or extensive weathering, some artefacts or misclassifications may still occur. Overall, OpenFraQ offers a robust and flexible solution, but optimal results depend on careful parameter adjustment and, in certain cases, require images with clear, recognisable fractures, as well as additional preprocessing or manual corrections.

2.4. Validation method

Given the geometric focus of the current work, the detection of fracture traces is considered more reliable than precise measurements of openness, which can be influenced by weathering and spreading. Therefore, we employed both manual and quantitative validation methods for trace detection, along with visual comparison for openness assessment. Since validation is a crucial aspect of scientific research, two approaches were selected and implemented: a manual method and the Hough transform [46,47] for detecting and comparing traces and a separate manual method for openness validation. For the latter, precision, recall, and F1 score (Eq. 14-16) are based on the contours obtained from OpenFraQ, defined as follows:

$Precision = TP / (TP + FP) \times 100$ where TP is a true positive and FP is a false positive.

(14)

$Recall = TP / (TP + FN) \times 100$ where FN is a false negative. (15)

$F1\ Score = 2 \times (Precision \times Recall) / (Precision + Recall)$ (16)

The Hough transform is a widely used feature extraction technique in image analysis that detects simple shapes such as lines, circles, and other parametric forms by mapping geometric features into a parameter space. This process enables the identification of these shapes within the original image, with its line detection capability being particularly valuable in applications like fracture detection in geoscience images [48–51]. The transform works by converting edge points from the image space (x, y) into a parameter space (ρ, θ), where shapes like lines can be recognised through prominent peaks in the accumulator array. Overall, the Hough transform is a powerful and commonly employed method for detecting rock fractures and other features in close-range imaging [19]. The manual detection was performed by a geologist, who identified the most prominent open fracture traces in each image using lines or simple polylines. The geologist was briefed on the goal of this trace detection, which involves geomorphological characterisation of open fractures, before starting. This process was carried out using Inkscape software, ensuring consistency by displaying all images at the same magnification level, avoiding over-zooming as recommended by Healy et al. (2016) [22].

FracPaQ2D [22] is an open-source toolbox that accepts two main types of input files: tab-delimited ASCII text files containing fracture trace nodes and graphical image files of fracture traces. The software utilises MATLAB™ scripts to analyse various fracture attributes, such as orientations, lengths, intensity, density, and connectivity, while also estimating permeability using a parallel plate model. FracPaQ2D uses a modified Hough transform approach, which has been enhanced by removing short segments and filling gaps, to identify co-linear pixel patterns in images, thereby generating fracture traces. The authors emphasise that node files derived from user-traced fracture data typically provide greater detail and accuracy than image files processed

through the Hough transformation. They strongly recommend preprocessing input images with image detection techniques before using this toolbox or employing a manual approach. This toolbox, which implements a modified version of the Hough transform and includes various image processing functions, is used for validating and comparing the traces obtained from the OpenFraQ function.

The mentioned preprocessing function outlines the steps for preparing image files for analysis with FracPaQ2D, employing the modified Hough transform method to detect fracture traces. It starts by loading an image file and retrieving its attributes. If the image is not already in greyscale, it converts it to that format. A median filter is then applied to the greyscale image to minimise noise. The function computes the difference between the filtered image and the original greyscale image before using the Sobel operator to identify edges. Subsequently, morphological operations, including closing and thinning, are performed to refine the edge representation. In the final step, the processed image is inverted and saved as a TIFF file, preparing it for further analysis with FracPaQ2D.

3. Results

In this section, five test images from a publicly available dataset [14] are used to evaluate the performance of the OpenFraQ algorithm. For comprehensive validation, fracture traces detected by FracPAQ, along

with manually traced references, are also provided and compared. Subsequently, when the fracture shape is known, precise width calculation can be achieved as the final objective.

3.1. Trace and fracture detection

Image segmentation typically integrates both region-based and edge-based algorithms to assess the shape and size of rock fractures [15]. Several adjustable parameters in the OpenFraQ algorithm play a critical role in fine-tuning the aperture detection process for analysing rock fractures. The δ parameter controls the degree of Gaussian smoothing applied to the greyscale image, which helps reduce noise and improve edge detection effectiveness. The $[m_1, m_2]$ parameter allows for customised thresholds in the Canny edge detection method, enabling flexibility in sensitivity to edge variations. The se parameter determines the size of the structuring element used in morphological operations, influencing the extent of dilation and closing applied to the detected edges. Additionally, the mcl parameter filters out contours that are too short, ensuring that only significant features are retained for analysis. The k parameter specifies the number of top contours (both widest and largest) to be highlighted, allowing the user to focus on the most prominent fractures in the image. Finally, based on the complexity of the contour shape, p controls the number of points selected for constructing polylines, if necessary. Together, these parameters provide a means to tailor the algorithm's performance based on specific image characteristics and analysis goals. Adjusted parameters for test images (Figs. 4, 10, 12, 14, 16) based on trial and error are depicted in Table 2:

The results of the function implementation for Image Test 1 (Fig. 2A), with parameters from Table 2 presented in Fig. 2, showcase all identified contours (Fig. 2B), which are presumed to represent open fractures; however, this may lead to an overestimation in some cases. Because of this, selecting the top k highlighted open fractures is important for more accurate analysis. Fig. 2C displays all the middle lines of these contours, while Fig. 2D illustrates the three widest contours (open fractures), with the red colour representing the first and the green colour representing

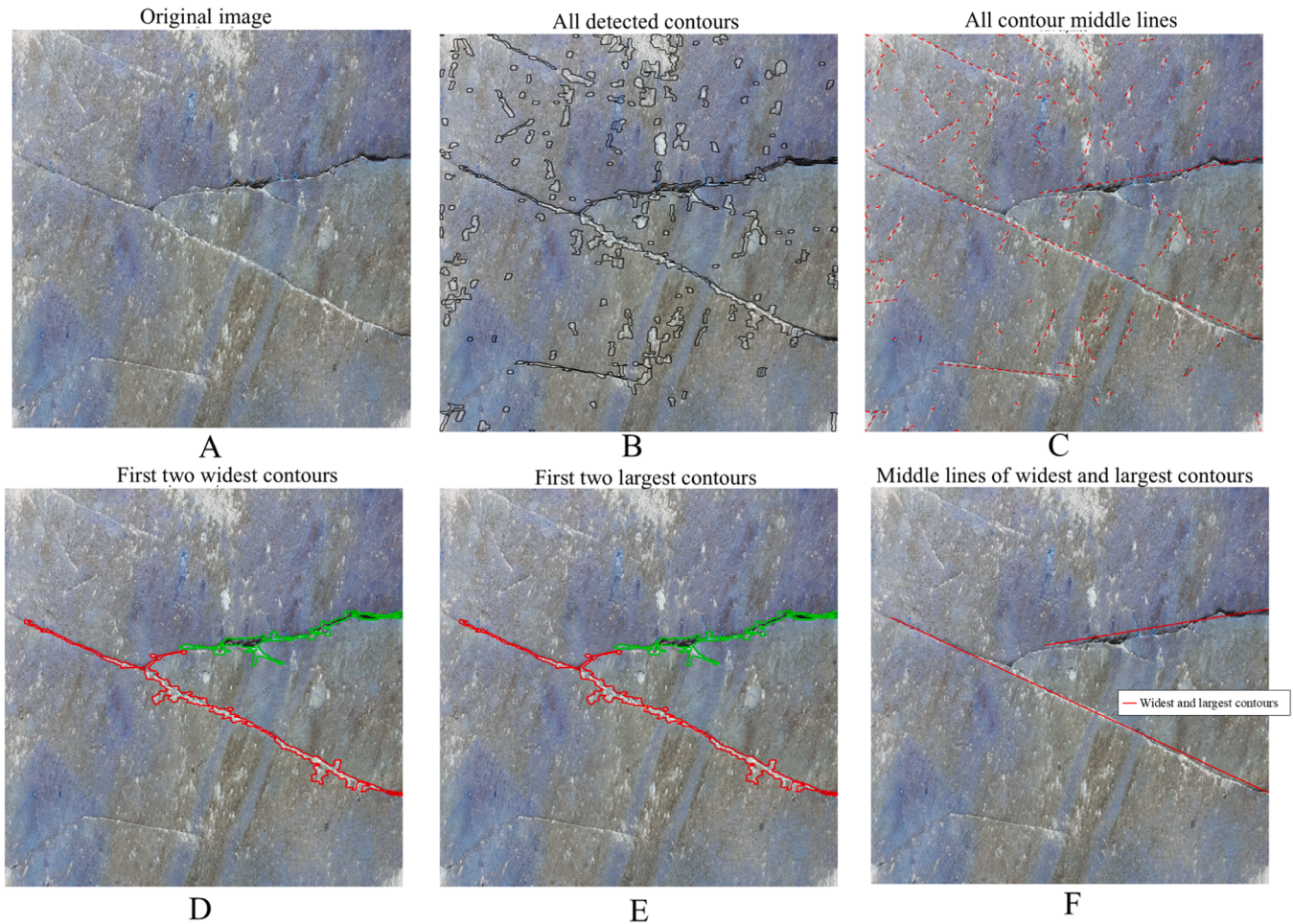


Fig. 4. OpenFraQ results for sample 1; A: original image; B: all detected contours (open fractures); C: all contour middle lines; D: first two widest contours (red second, green second); E: first two largest contours (red second, green second); F: middle lines related to the first two widest and largest contours.

Table 2

Adjusted parameter for OpenFraQ.

Parameters	δ	$[m_1, m_2]$	se	mcl	k	p
Value for Sample 1	2	[0.1, 0.3]	10	10	2	0
Value for Sample 2	6	[0.1, 0.3]	10	10	2	0
Value for Sample 3	3	[0.3, 0.5]	10	10	3	0
Value for Sample 4	6	[0.1, 0.3]	10	10	3	1
Value for Sample 5	5	[0.15, 0.35]	20	10	1	1

the second. In Fig. 2E, following the same colour order, the first three largest contours (open fractures) are presented, along with their middle lines (traces) in Fig. 2F. However, as the traces are similar, it is important to note that different traces can be detected for the largest and widest open fractures, depending on geomorphological properties.

To validate the results, it is necessary to compare them with an algorithm that detects traces as lines, as this method is more practical for further mathematical geomorphological analysis and modelling. The modified Hough transform was implemented in FracPaQ2D through preprocessed original samples (Fig. 5A) (using the preprocess tool that is suggested with the FracPAQ toolbox) using both default parameters and adjusted parameters (as shown in Table 3) obtained through trial and error, with the obtained Hough transform parameters ρ and Θ illustrated in Figs. 6 and 7, respectively. In the Hough transform, ρ denotes the distance from the origin to the closest point on a line, while Θ specifies the angle of the line concerning the horizontal axis. These parameters together facilitate the representation of lines in polar coordinates by

mapping Cartesian space points to the Hough parameter space. This process aids in identifying straight lines in images by locating the convergence of ρ and Θ , thus, allowing the discovery of local maxima in the parameter space. In conclusion, the bright yellow points in the ρ and Θ plots indicate the lines in the original image space (Figs. 6 and 7).

Iterative trial and error in adjusting the modified Hough transform parameters indicates that beyond a certain point, defined as the optimal parameter set, no further improvements are observed in the results. During trial and error, it was observed that significantly increasing the Hough transform threshold and gap-filling parameters caused the loss of the principal trace patterns. Conversely, setting these parameters too low resulted in very short lines (lacking geomechanical analysis value), as well as the detection of false traces and noise amplification.

Figs. 5B and 5C show the modified Hough transform outputs on the preprocessed image (Fig. 5A), using default and adjusted parameters. Both outputs exhibit overall shape detection; however, the result obtained with default parameters consists of numerous small lines that are ineffective for identifying fracture attributes such as orientation, length, intensity, and density. Additionally, the excessive noise from scattered tiny lines unrelated to fractures poses a significant challenge. Conversely, adjusting the parameters yields a clearer result that aligns more closely with the study's objective of automatically identifying prominent fracture traces, but disconnectivity and extra lines still exist in trace identification, and truly representative traces are not available. The OpenFraQ algorithm somewhat mitigates this issue. Fig. 8 compares the prominent fracture traces obtained via the modified Hough transform, OpenFraQ, and manual tracing.

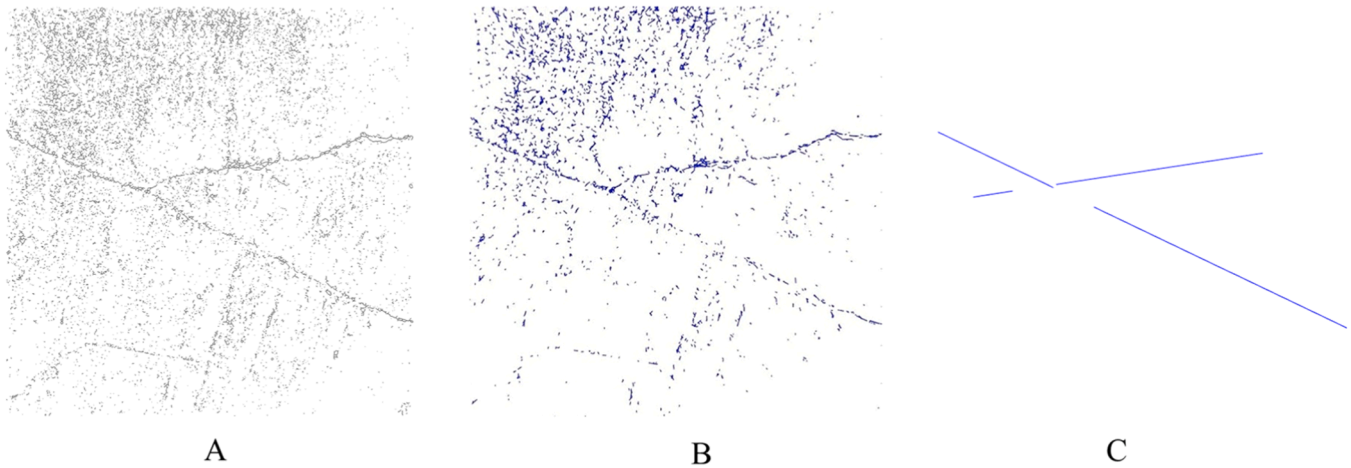


Fig. 5. Implementation of preprocessing and modified Hough transform by FracPaQ2D; A: Preprocessed image; B: lineament extraction with default parameters; C: lineament extraction results with adjusted parameters.

Table 3
Adjusted parameter for FracPaQ2D algorithm.

Parameters	Number of Hough Peaks	Hough Threshold	Merge gaps less than	Discard lengths less than	Median filter size (preprocess)
Default value	1000	0.33	5	3	25×25
Value for sample 1	500	0.8	20	3	25×25
Value for sample 2	500	0.8	100	5	25×25
Value for sample 3	500	0.7	70	5	25×25
Value for sample 4	500	0.84	60	3	25×25
Value for sample 5	500	0.9	90	5	25×25

Revisiting the properties of contours obtained from OpenFraQ enables an assessment of whether the identified number of dominant open fractures (k) is realistic. All contours detected in Test Image 1, along with their corresponding label numbers, are displayed in Fig. 9A, while Figs. 9B and 9C display the distributions of contour areas and bounding widths. The area and bounding box width distributions strongly support $k=2$, although a more lenient interpretation suggests $k = 3$. The same

procedure was applied to the other samples (Figs. 18–21), further validating the selection of k .

The geometric aperture is defined as the width of a fracture measured perpendicular to its walls, representing the extent of the opening across the fracture. It is equivalent to the thickness of veins [52] and dykes [53,54]. So, modelling a fracture as a rectangle enables the extraction of the fracture's aperture area (S) and its longitudinal vector (Length Vector, L) based on previously recorded data. Although estimating the average width ($W_{ave} = S/L$) is possible, this value may not be entirely accurate since the width varies along the fracture. Nonetheless, it provides a useful approximation of the fracture aperture, particularly in areas that are difficult to access. Moreover, the bounding box provides an estimate of a single fracture and its sub-branches' spread. The overall dominant aperture characteristics, including trace length, aperture area, bounding width, and perimeter, for image test 1 and other tested images are summarised in Table 4. While these measurements are expressed in pixels, they can be converted to real-scale units.

The same procedure that was followed for test image 1 was followed for test images 2–5 (Figs. 10, 12, 14, 16A) with parameters from Table 2 and results presented in Figs. 10, 12, 14, 16(B–F). In cases where P is not equal to zero, the location of the middle points is visualised too, as shown in Figs. 12C and 14C. For validation of the results, the overall traces obtained from OpenFraQ are visualised and compared with the modified Hough transform and manual results in Figs. 11, 13, 15, and 17. Additionally, the highlighted fractures, along with their labels,

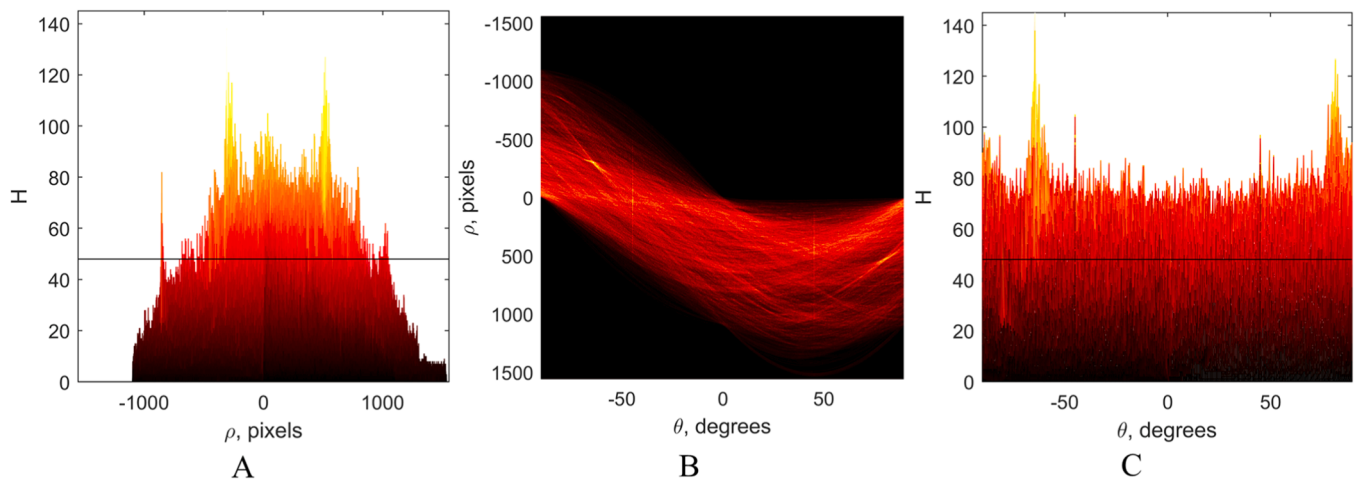


Fig. 6. Hough peaks by FracPaQ2D implementation with default parameters; A: Side view (θ); B: side view (θ); C: plan view (ρ, θ).

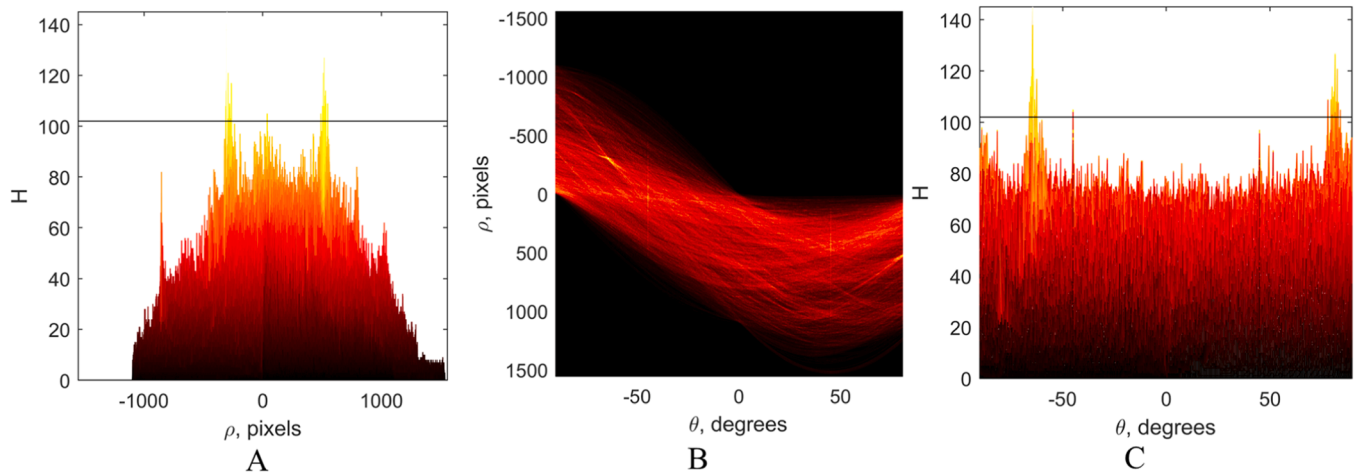


Fig. 7. Hough peaks by FracPAQ2D implementation with adjusted parameters; A: side view (Θ); B: side view (Θ); C: plan view (ρ, Θ).

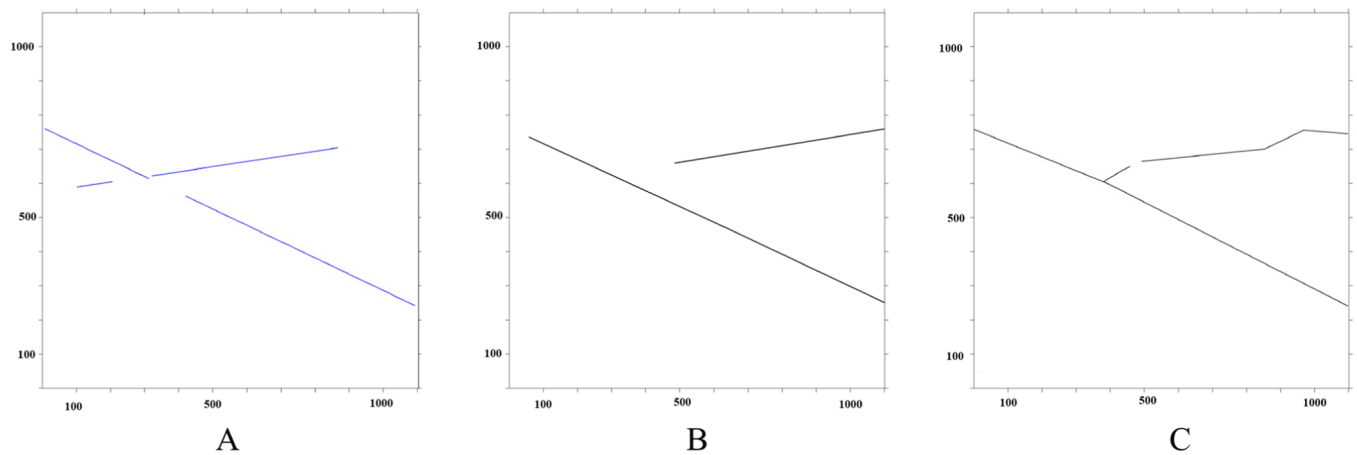


Fig. 8. Sharp fracture trace detection for sample 1; A: FracPaQ2D; B: OpenFraQ algorithm; C: manual approach.

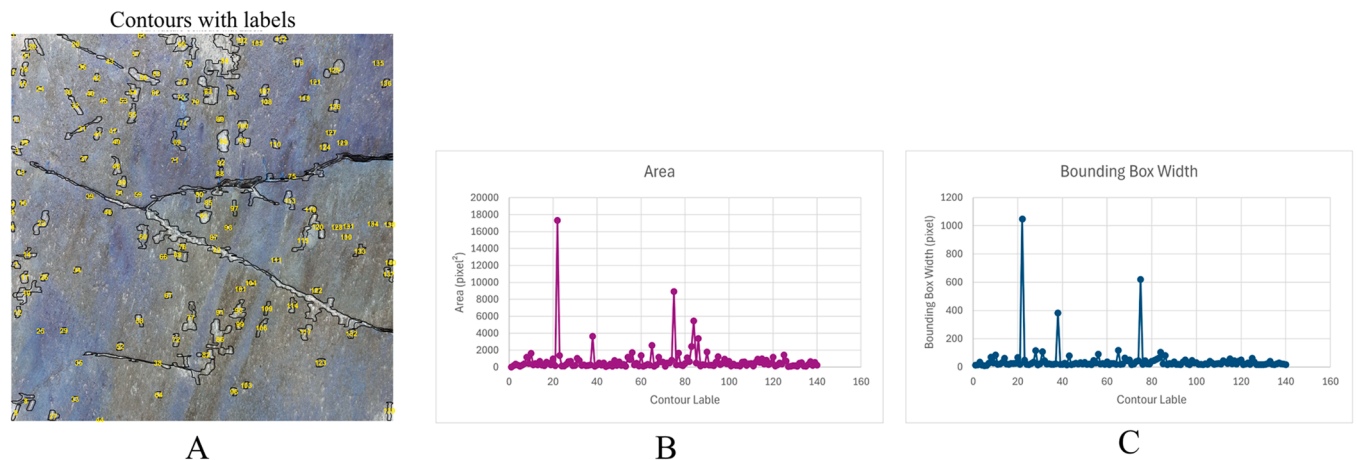


Fig. 9. Analysis of OpenFraQ Results for sample 1: A: highlighted fractures and their labels; B: area of detected contours; C: bounding box width of detected contours.

areas, and bounding box widths, are displayed in Figs. 16-19, confirming the realistic value of k for each tested image. However, this value can be adjusted depending on the specific objectives or desired level of precision in the research. For instance, in some cases, such as test image 3 (Fig. 12), the widest aperture-related traces do not correspond to the longest traces, or they may be partially overlapped, as seen in test image

4 (Fig. 14). Details of the Hough transform process for sample images 2-5 (Figs. 10, 12, 14, 16A) are omitted to maintain brevity. Besides, manual detection methods, as described in the methodology section, are also employed.

Fig. 8 shows the results of implementing OpenFraQ, the modified Hough transform, and manual trace detection for test image 1. The

Table 4

Single fractures' aperture characterisation with OpenFraQ.

Sample Number	Aperture Label	Trace Length (pixel)	Aperture area (pixel ²)	Aperture area (pixel ²)/Trace length (pixel)	Bounding width (pixel)	Area Perimeter (pixel)	Average of modified (MRD-TMA) width (pixel)
Sample 1	ID22	1141.8	15.17	8.41	1048	3764.8	15.7
	ID75	742.7	8922.5	12.01	619	2559.5	29.84
Sample 2	ID6	329.3	6864.5	20.84	237	1220.2	16.84
	ID26	1003.1	21,926.5	21.85	732	3105.8	19.24
Sample 3	ID5	315.8	888.5	2.8	316	715.3	5.08
	ID24	230	3159	13.73	57	543.1	26.34
	ID25	75	99	1.32	76	154.1	7.8
	ID26	259.6	3931	15.14	55	730.2	35.39
	ID30	187.2	3031.5	16.19	65	508.3	21.24
Sample 4	ID54	130.8	996.5	7.61	131	340.1	22.41
	ID2	273.3	17,113.5	62.61	172	2221	40.82
	ID15	828.66	26,271	31.7	777	5079	37.95
	ID63	187.4	7593	40.51	81	1181.4	90.19
Sample 5	ID17	1918.90	85,895	44.76	677	3940.81	79.49

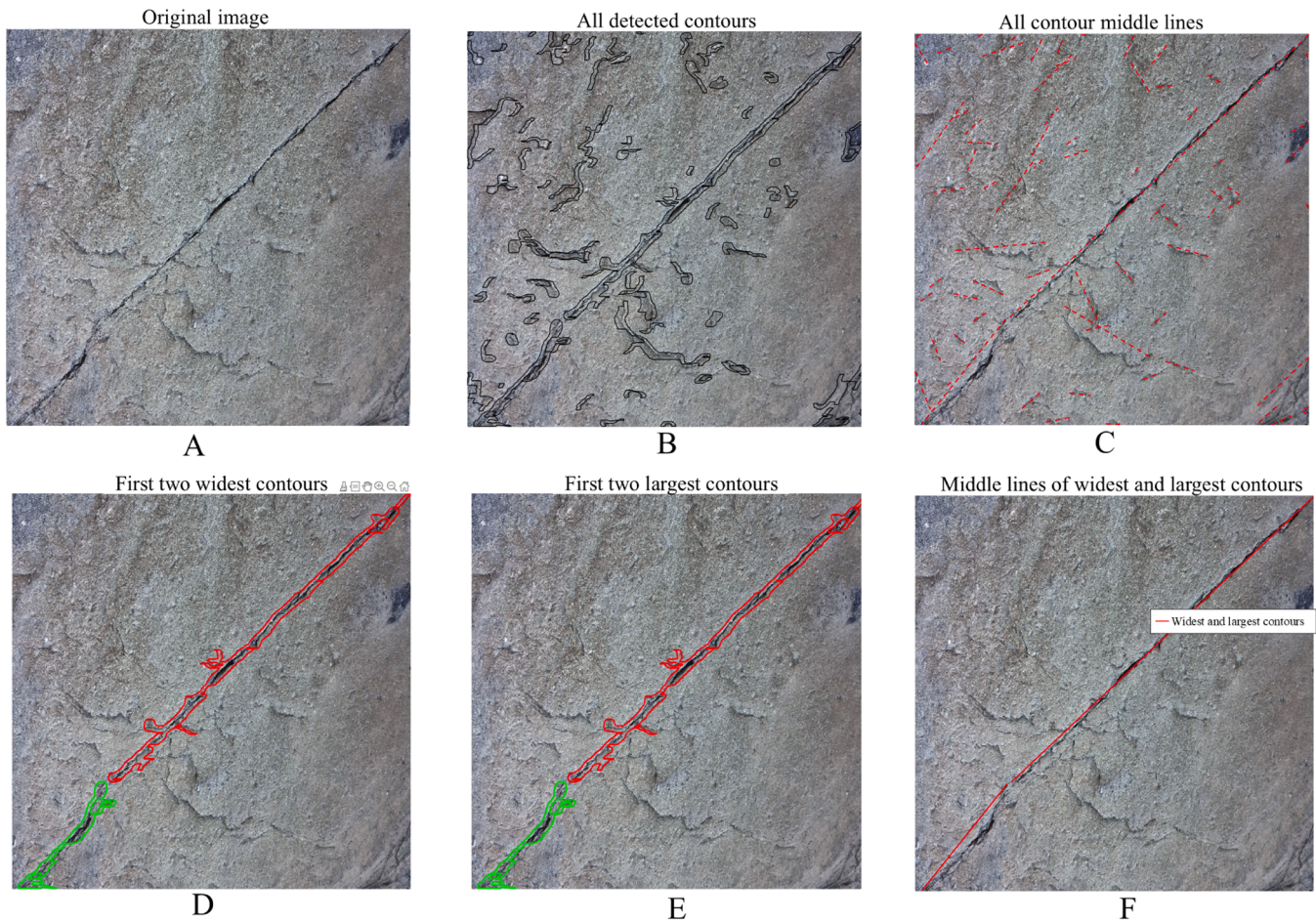


Fig. 10. OpenFraQ Results for sample 2; A: Original image; B: all detected contours (open fractures); C: all contour middle lines; D: First two widest contours (red second, green second); E: First two largest contours (red second, green second); F: Middle lines related to the first two widest and largest contours.

output of OpenFraQ closely resembles the manual detection but in a simpler form, whereas the modified Hough transform produced extra lines and missed some parts of the trace. As shown in Fig. 11, the modified Hough transform applied to test image 2 did not detect the bending of the overall trace, while the OpenFraQ implementation successfully identified it, producing a result like the manual drawing of this test image. In contrast, the modified Hough transform results for test image 3 in Fig. 13 show satisfactory overall trace detection, but it appears that distinguishing between more open fractures and tightly open fractures was not feasible with the modified Hough transform and

manual approach. In comparison, OpenFraQ successfully differentiated between them, which can be considered either a strength or a weakness depending on the specific objectives of the project.

In another examination of test image 4 (Fig. 15), it is evident that a fracture in the bottom right of the image was missed by the modified Hough transform, while an extra line in the weathered upper-left area was detected by OpenFraQ. These results suggest that, in future work, incorporating constraints specifically for weathered parts could improve accuracy. Overall, the traces detected by OpenFraQ are more consistent with the manual detection than those from the modified Hough

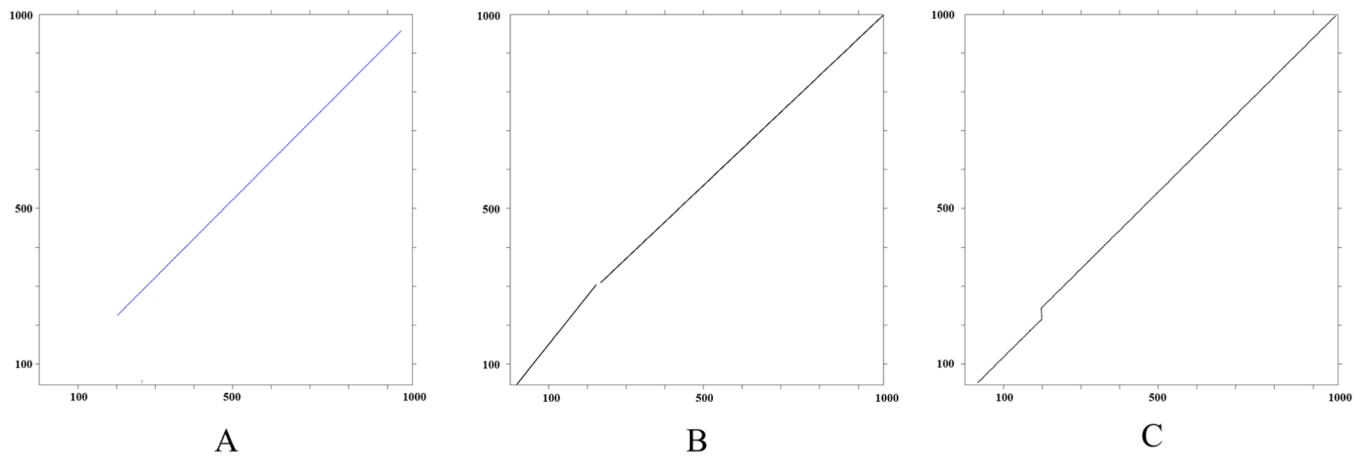


Fig. 11. Sharp fracture trace detection for sample 2; A: FracPaQ2D; B: OpenFraQ algorithm; C: manual approach.

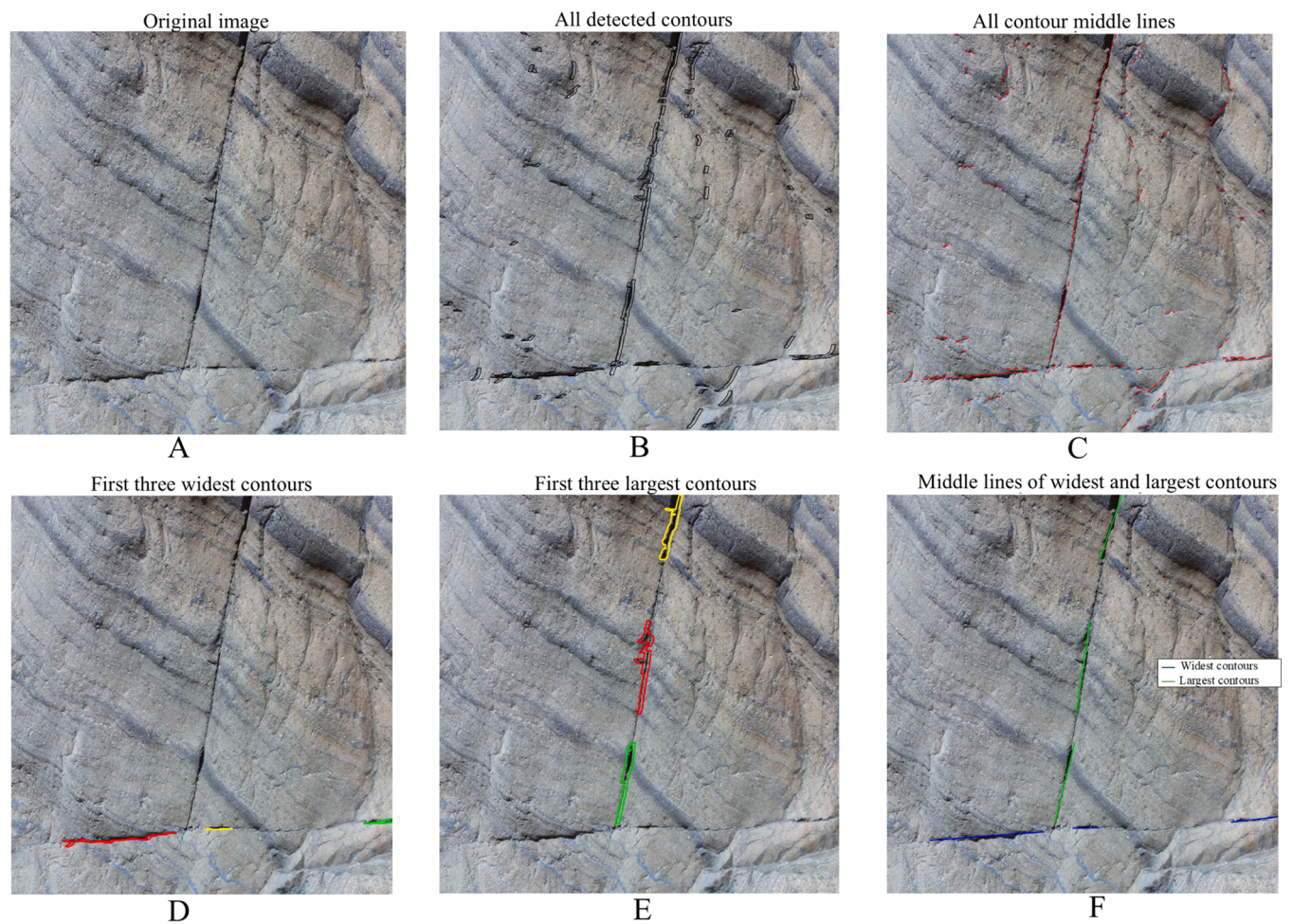


Fig. 12. OpenFraQ Results for sample 3: A: original image; B: all detected contours (open fractures); C: all contour middle lines; D: first two widest contours (red second, green second, yellow third); E: first two largest contours (red second, green second, yellow third); F: middle lines related to the first two widest and largest contours.

transform. Test image 5, which features an irregular open fracture, was processed using both OpenFraQ and the modified Hough transform. The comparison with manual trace detection is shown in Fig. 17, which clearly shows that the modified Hough transform failed to detect the overall trace, only capturing some parts of the aperture wall. In addition, despite the presence of vegetation cover in the image, the inclusion of a middle line ($p=1$) somewhat mitigated this error source, resulting in a

final detection that closely resembles the manual tracing.

3.2. Width calculation

Existing aperture calculation methods, including the equivalent circle, equivalent ellipse, and Ferret techniques, aim to estimate the aperture by determining the diameter of an equivalent circle or ellipse or

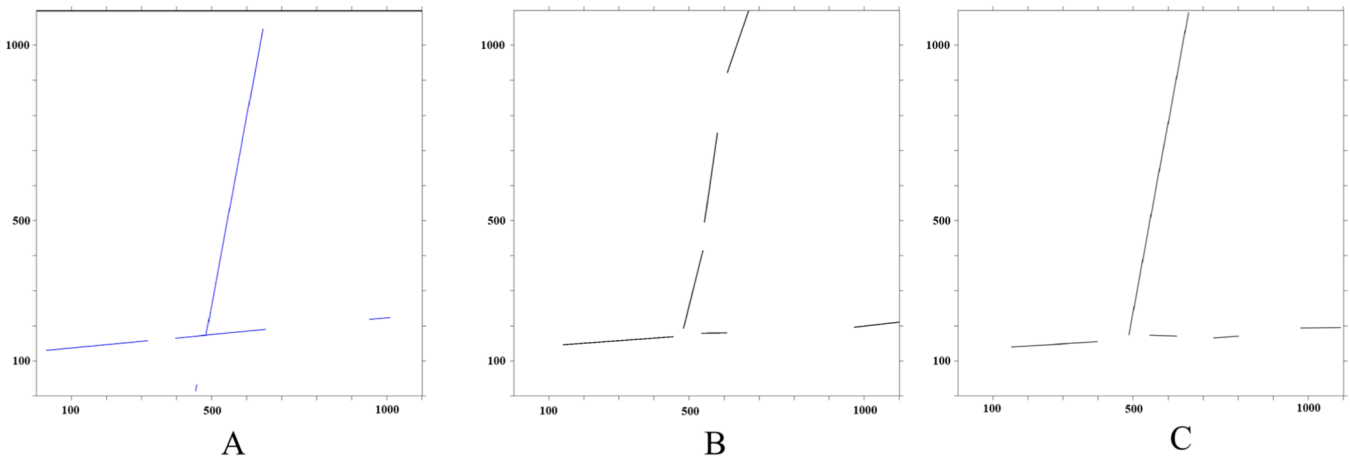


Fig. 13. Sharp fracture trace detection for sample 3; A: FracPAQ2D; B: OpenFraQ Algorithm; C: manual approach.

Table 5
Implementation duration for applied methods.

	FracPAQ2D	Proposed Algorithm	Manual Operation
Sample 1	17(s)	20.79(s)	160(s)
Sample 2	14.34(s)	10.51(s)	135(s)
Sample 3	19.2(s)	15.6(s)	122(s)
Sample 4	19.44(s)	24.02(s)	145(s)
Sample 5	20.77(s)	18.47(s)	182(s)

Table 6
Algorithm performance metrics: precision, recall, and F1 score by samples.

Sample	Precision (%)	Recall (%)	F1 Score (%)
Sample 1	60.18	63.00	61.56
Sample 2	80.11	64.71	71.59
Sample 3	50.17	65.19	56.70
Sample 4	51.21	41.14	45.63
Sample 5	76.17	75.52	75.85

the dimensions of an equivalent rectangle that encloses the fracture at selected points along its length with specified spacing [4,15,40]. While these methods can effectively depict the width distribution alongside the fracture trace, they are primarily suitable for fractures with simple, regular shapes. They are less effective for irregularly shaped fractures [15]. For the obtained results, the modified (MRD-TMA) method was applied to the highlighted fractures in the tested sample images. While the detected traces are assumed to be the middle line, the results are presented in Figs. 22-26 as width distribution charts. These results visually align well with the observed fractures. Besides, Table 4 presents the average modified (MRD-TMA) width for each tested fracture. For simpler fracture shapes, this value closely matches the aperture area/trace length ratio, whereas for more irregular shapes, the difference between the two values increases.

3.3. Operation time

The computational platform utilised for implementation comprised a workstation with the following specifications: an Intel® Core™ i7-8700 processor operating at 3.20 GHz with six cores and twelve threads; 32 GB of DDR4 RAM; an NVIDIA GeForce RTX 2070 GPU with 8 GB of GDDR6 memory; and a dedicated virtual memory of 17.4 GB. These hardware specifications facilitated efficient processing for image analysis tasks. Manual trace detection for the top k dominant traces is performed on all samples using Inkscape software. While the implementation of FracPAQ2D and OpenFraQ is carried out in MATLAB R2024b, a comparison of the processing times for the FracPAQ and

OpenFraQ implementations, as well as manual trace detection, is presented in Table 5. However, the processing times for the modified Hough transform and OpenFraQ are very similar, but both methods outperform manual detection in terms of speed.

3.4. Openness Validation

To assess OpenFraQ's performance, its results were compared to geologist-annotated ground truth images using precision, recall, and F1-score metrics. Figure 27 compares manual fracture detection (green), OpenFraQ results (red), and overlapping regions (yellow) across five representative samples. Table 6 presents the precision, recall, and F1 scores for each sample, along with the average performance. OpenFraQ's effectiveness varied depending on sample characteristics. Sample 1 (Fig. 27A) demonstrated strong performance across all metrics, while samples 2 and 5 (Figs. 27B and 27E) achieved the highest F1 scores (71.59% and 75.85%, respectively). Notably, Sample 2 also attained the highest precision (80.11%), attributed to its simple fracture geometry. In contrast, Sample 3 (Fig. 27C) showed high recall (65.19%) but lower precision (50.17%), indicating successful detection of actual fractures alongside a higher rate of false positives, partly due to missed narrow cracks classified as non-open fractures. Sample 4 (Fig. 27D) recorded the lowest scores (precision 51.21%, recall 41.14%, F1 score 45.63%), primarily due to misclassification of weathered rock surfaces as fractures.

3.5. Real case study

Sample 6 (Fig. 28B) shows an exposed rock outcrop (Fig. 28A) from an abandoned quarry in Tuscany, Italy, distinguished by a prominent, difficult-to-access open fracture on its surface. High-resolution images were obtained using a calibrated Canon 550D DSLR camera (Fig. 28C), positioned parallel to the rock face to ensure accurate spatial measurement. Unlike other samples, Sample 6 features a physical ruler and reference letters placed on the outcrop (Fig. 28D), enabling precise calibration of pixel-to-real-world scale for fracture width measurement.

Table 7
Algorithm performance metrics: precision, recall, and F1 score by samples.

Sample	Precision (%)	Recall (%)	F1 Score (%)	Manual average width (cm) of target area, employing modified (MRD-TMA)	OpenFraQ average width (cm) of target area, employing modified (MRD-TMA)
Sample 6	60.18	63.00	61.56	8.52	11.6

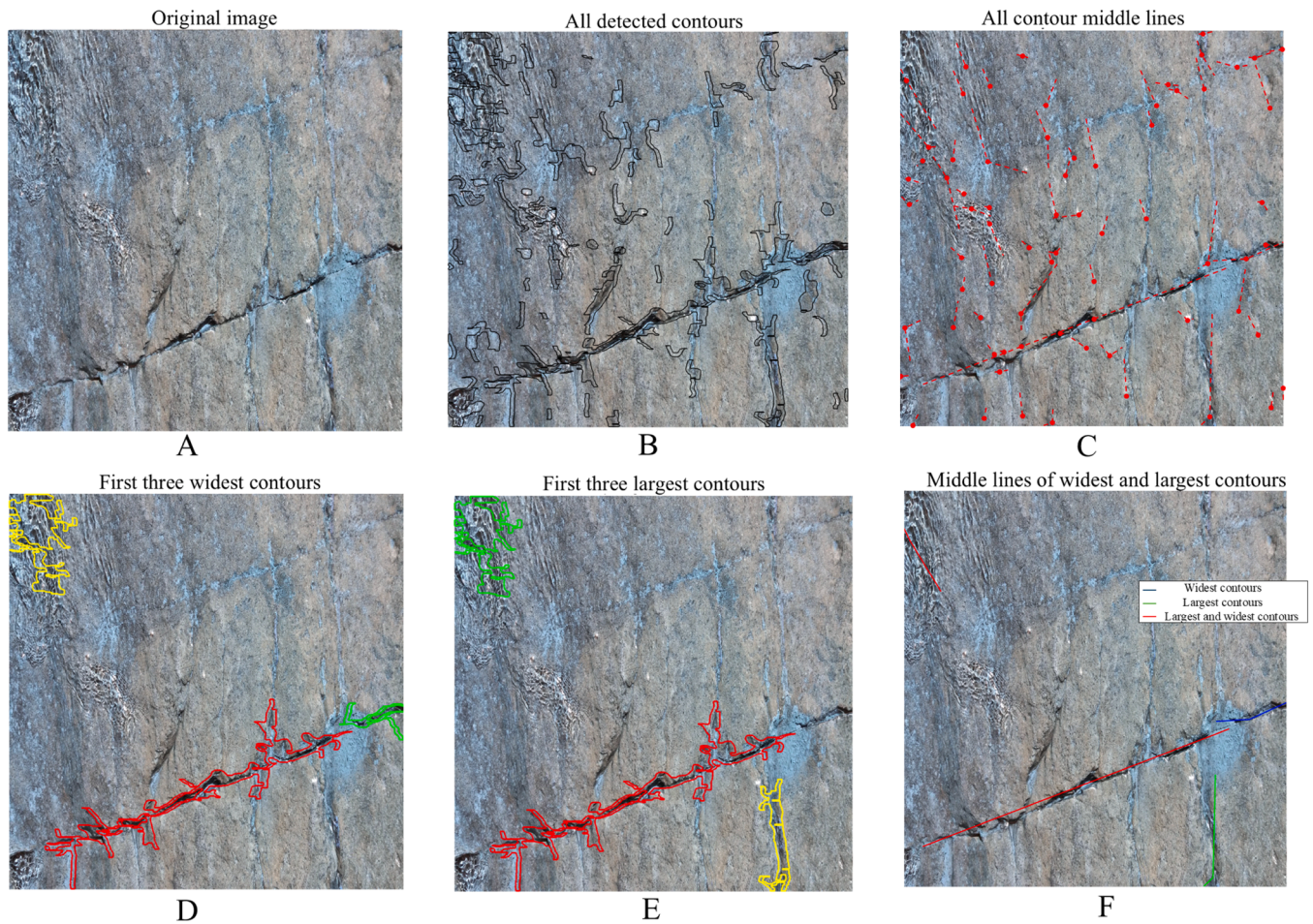


Fig. 14. OpenFraQ Results for sample 4; A: original image; B: all detected contours (open fractures); C: all contour middle lines; D: first two widest contours (red second, green second, yellow third); E: first two largest contours (red second, green second, yellow third); F: middle lines related to the first two widest and largest contours.

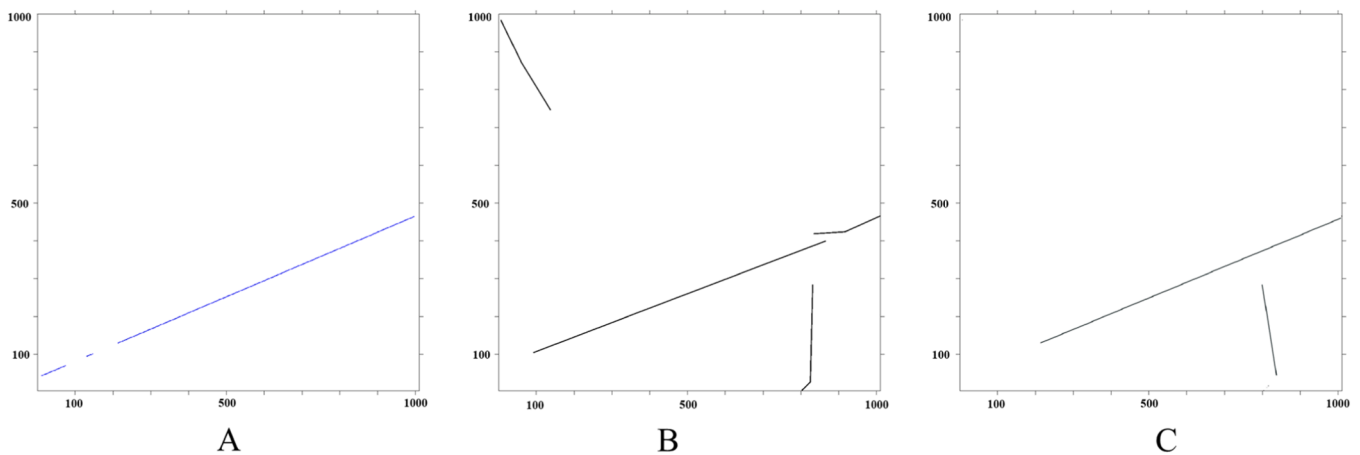


Fig. 15. Sharp fracture trace detection for sample 4; A: FracPaQ2D; B: OpenFraQ algorithm; C: manual approach.

A geologist manually quantified fracture openness by digitising the image, tracing the fracture, and using the reference markers for calibration to determine exact aperture values. This process established the pixel size within the selected window as 2.60 mm.

For automated analysis, OpenFraQ was set to extract the predominant fracture within the selected sample window (Fig. 28B). Due to the extensive area, limited number of fractures, and significant noise from

minor cracks and weathering, which required a higher Se value, parameter settings were optimised through sensitivity analysis and trial-and-error. The final configuration included $\sigma = 6$, $Se = 25$, $mcl = 10$, Canny edge thresholds = $[0.2, 0.4]$, $k = 3$, and $p = 1$. Fracture contours and aperture geometry were identified using contour-based segmentation and bounding box methods (Fig. 29A–E), enabling the algorithm to filter and highlight the widest and most prominent fractures. As with

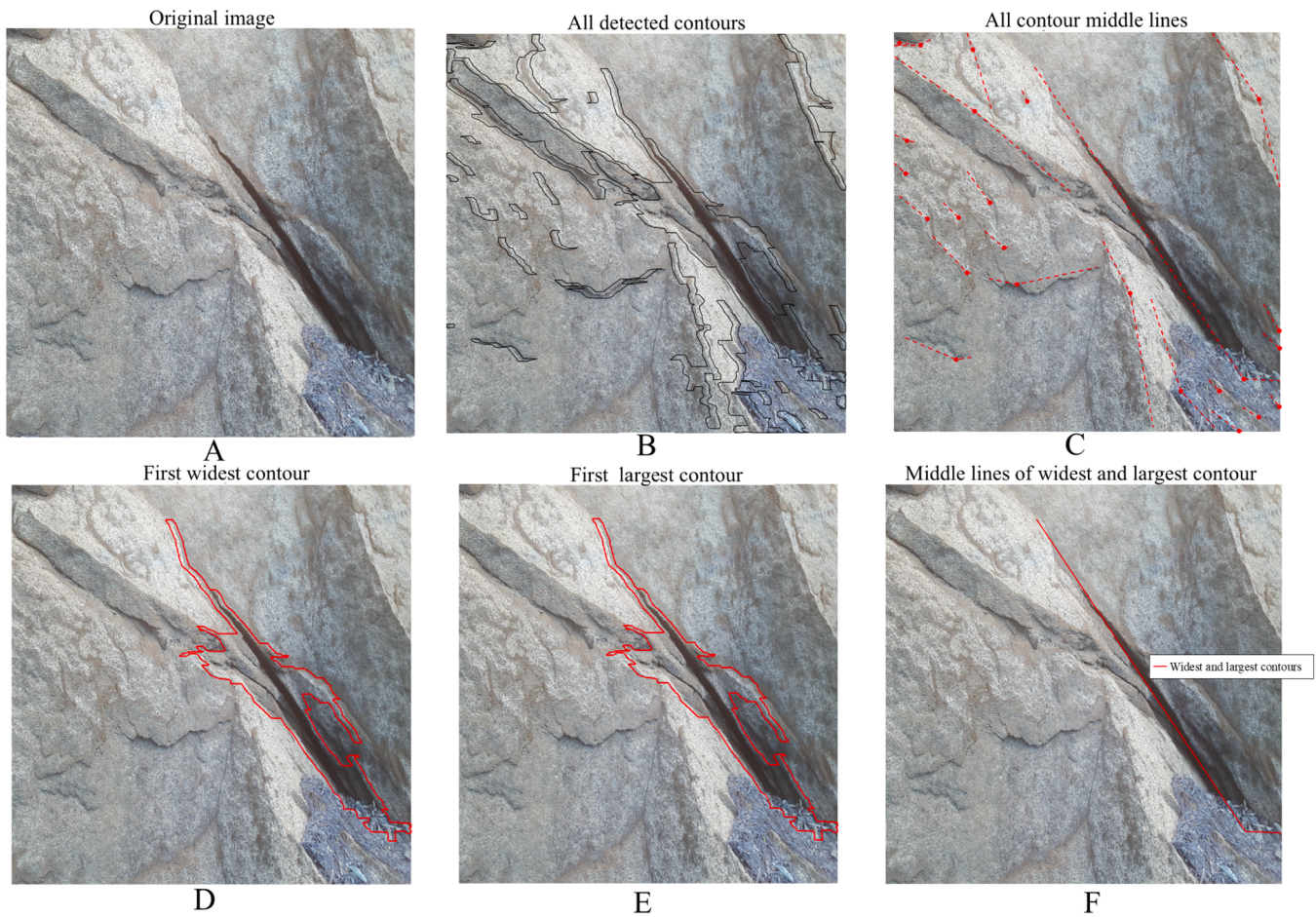


Fig. 16. OpenFraQ results for sample 5; A: original image; B: all detected contours (open fractures); C: all contour middle lines; D: first two widest contours; E: first two largest contours; F: middle lines related to first two widest and largest contours.

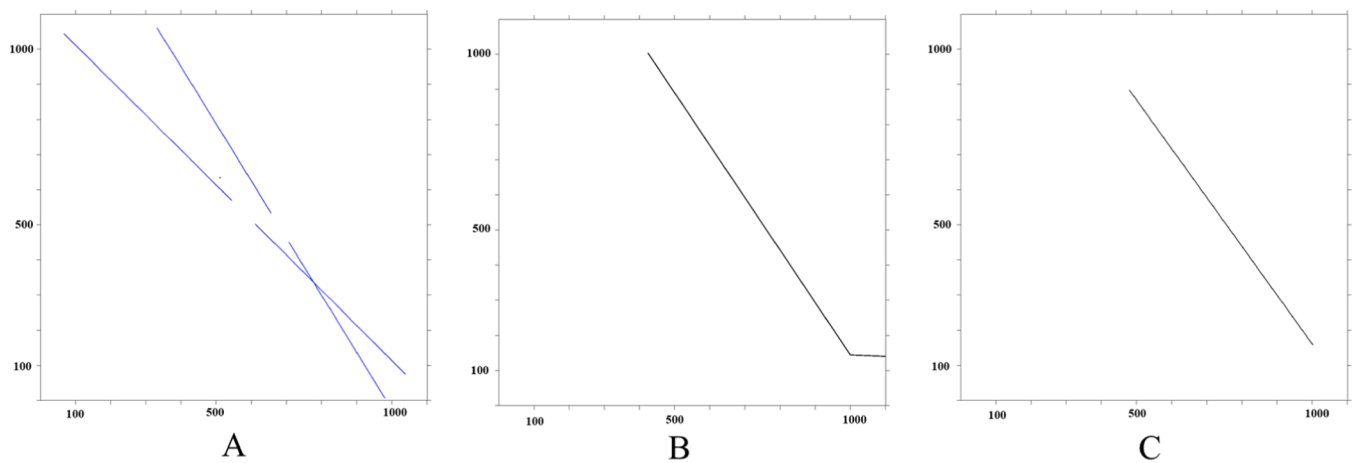


Fig. 17. Sharp fracture trace detection for sample 5; A: FracPaQ2D; B: OpenFraQ algorithm; C: manual approach.

previous samples, a geologist validated fracture openness (Fig. 29F), and the results, shown in Table 7, indicate a precision of 60.18%, a recall of 63%, and an F1 score of 61.56%, reflecting moderate accuracy of automated detection in this challenging setting.

As previously observed in sample 4 (Fig. 14D-E, upper left corner) and now in sample 6 (Fig. 29B, center of the image), weathered areas can cause misinterpretation between open fractures and weathered surfaces. To reduce this ambiguity, it is recommended to mask

weathered regions before conducting fracture characterisation. Similarly, vegetated areas should also be avoided or masked, as vegetation can obscure fracture features and further complicate accurate detection. Additionally, sample 6 (Fig. 29C, upper left with the three widest fractures) demonstrates that shadow zones and hidden corners can adversely affect fracture detection and characterisation, particularly in trace detection. To minimise these issues, images should be captured with adequate lighting, avoiding shadows and ensuring a complete,

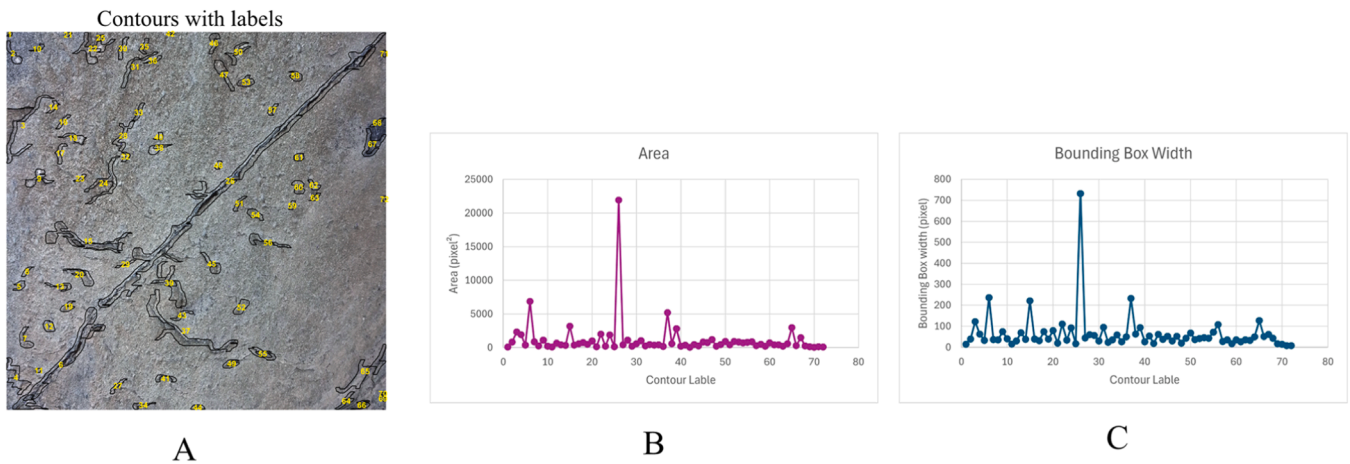


Fig. 18. Analysis of OpenFraQ Results for sample 2; A: Highlighted fractures and their labels; B: area of detected contours; C: bounding box width of detected contours.

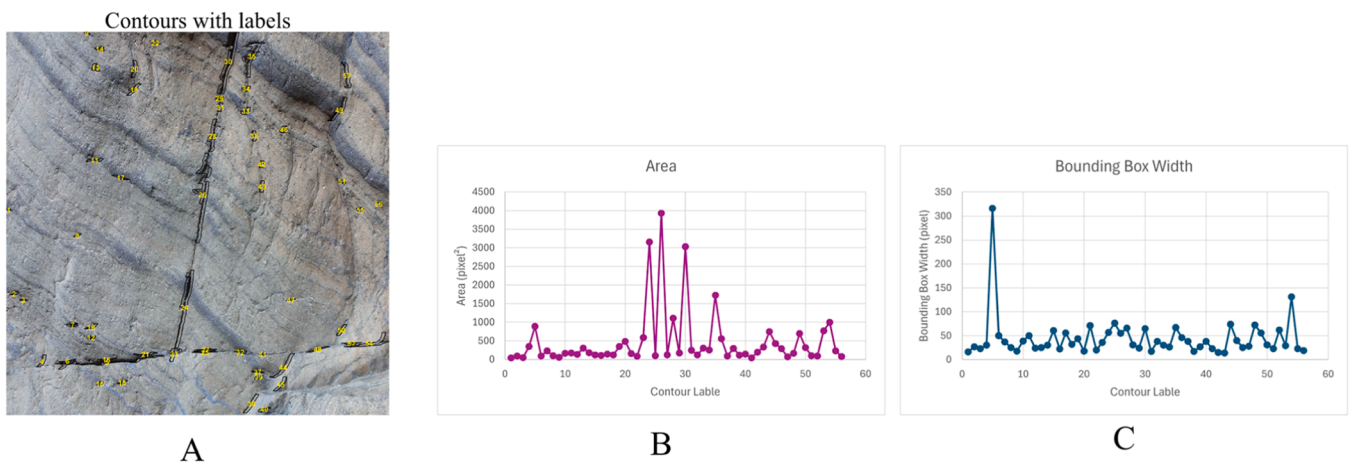


Fig. 19. Analysis of OpenFraQ results for sample 3; A: Highlighted fractures and their labels; B: area of detected contours; C: bounding box width of detected contours.

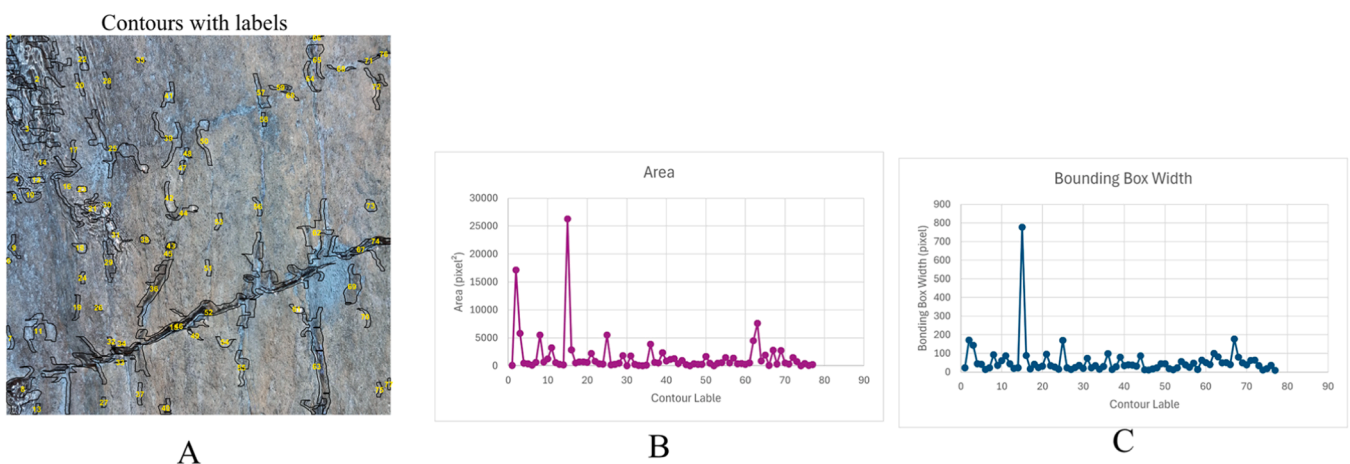


Fig. 20. Analysis of OpenFraQ Results for sample 4; A: Highlighted fractures and their labels; B: area of detected contours; C: bounding box width of detected contours.

unobstructed view of the rock surface.

The manually calculated average fracture width was estimated at 11.6 cm, while the OpenFraQ algorithm produced an estimate of 8.52 cm, approximately 36% lower, but still relatively close (Table 7). This

discrepancy may be attributed to several factors, including human error in manual measurements, misclassification of weathered areas as fractures, and variations in pixel intensity caused by shadows or lighting conditions in the images. Additionally, differences in the tolerance

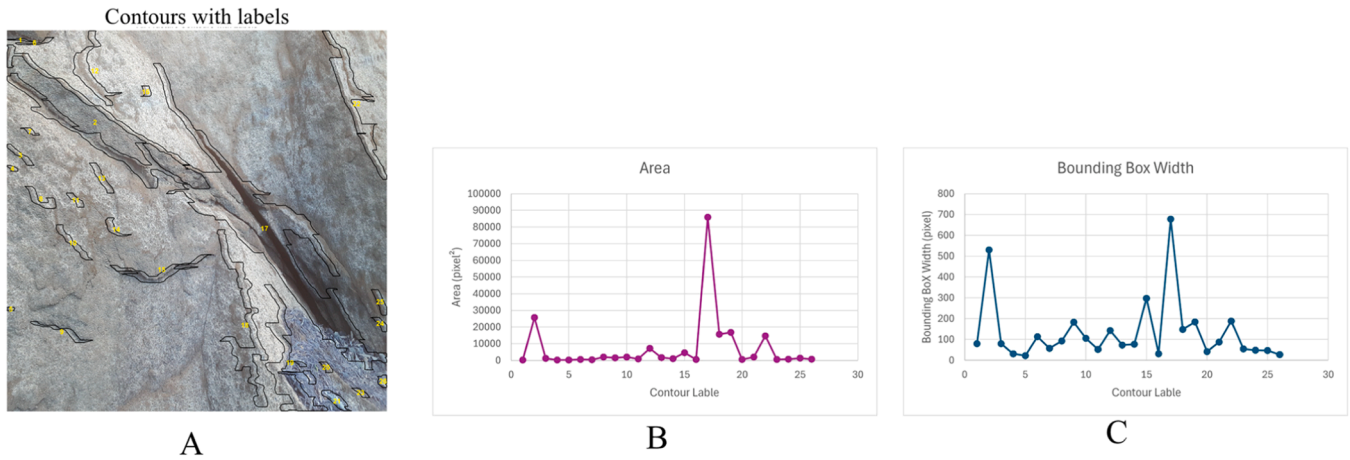


Fig. 21. Analysis of OpenFraQ Results for sample 5; A: Highlighted fractures and their labels; B: area of detected contours; C: bounding box width of detected contours.

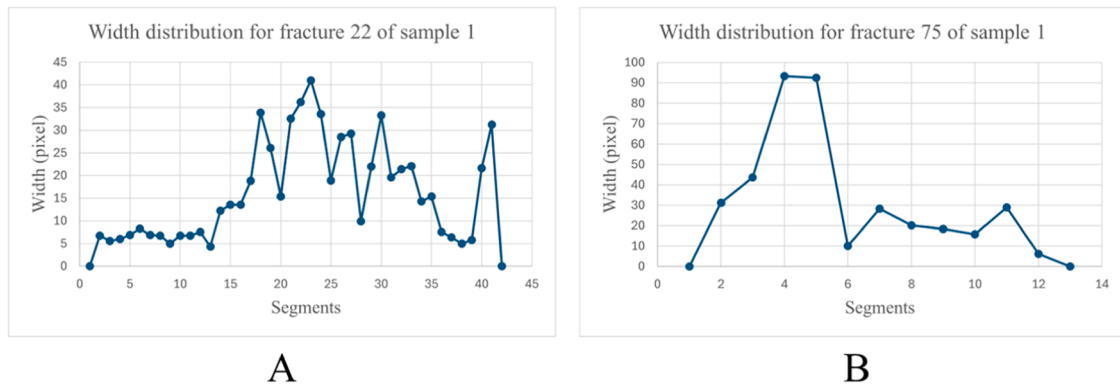


Fig. 22. Width distribution for highlighted fractures of sample 1; A: fracture number 22; B: fracture number 75.

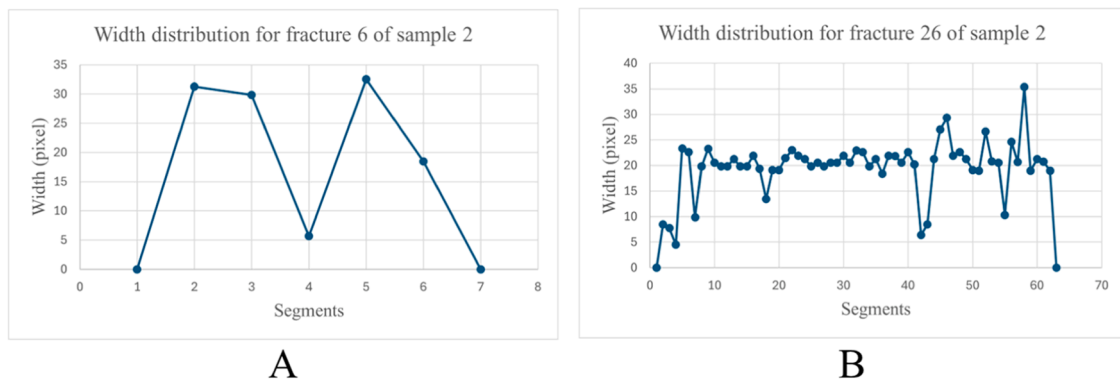


Fig. 23. Width distribution for highlighted fractures of sample 2 in a) fracture number 6, b) fracture number 26.

settings for intensity thresholds during automated analysis may contribute to the variation in results.

4. Discussion

Various algorithms, like Steger and Hessian-based methods [11,23,55], focus on extracting fracture shapes, often relying on centerline detection or adaptive histograms [21]. Even in the machine learning field, several methods have been developed for this purpose [56–58], especially by employing the CNN (Convolutional Neural Network) tool [14,59], creating a comprehensive, fully automatic training dataset

applicable across diverse environments and rock types remains challenging. Developers of automated fracture analysis tools like FracPAQ [22], QDC-2D [60], and Fractopo [61] strongly recommend that these kinds of algorithms cannot substitute for manual approaches, and that the condition of the input image and its processing affect the results. Therefore, we do not claim that our work can be applied in any environment, as other presented work does. Still, we firmly believe that this work represents a significant step forward in solving the fracture recognition problem, particularly for open fractures and geometrical studies.

Although OpenFraQ is designed to analyse multi-fracture images, a

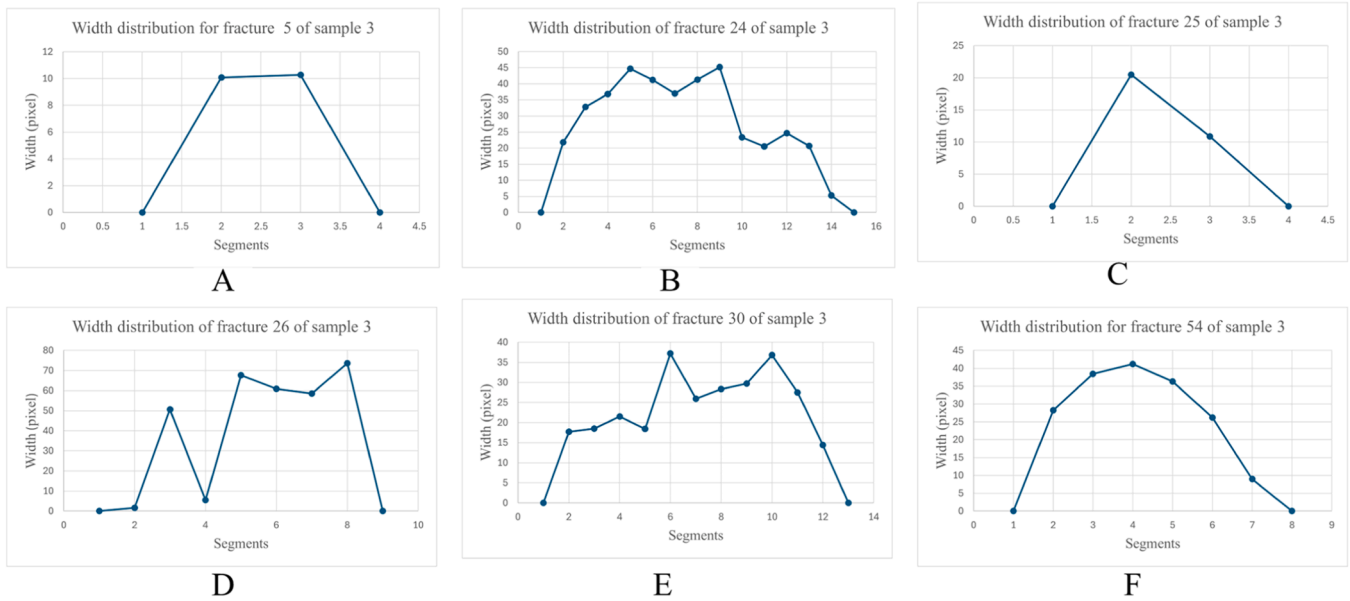


Fig. 24. Width distribution for highlighted fractures of sample 3 in a) fracture number 5, b) fracture number 24, c) fracture number 25, d) fracture number 26, e) fracture number 30, f) fracture number 54.

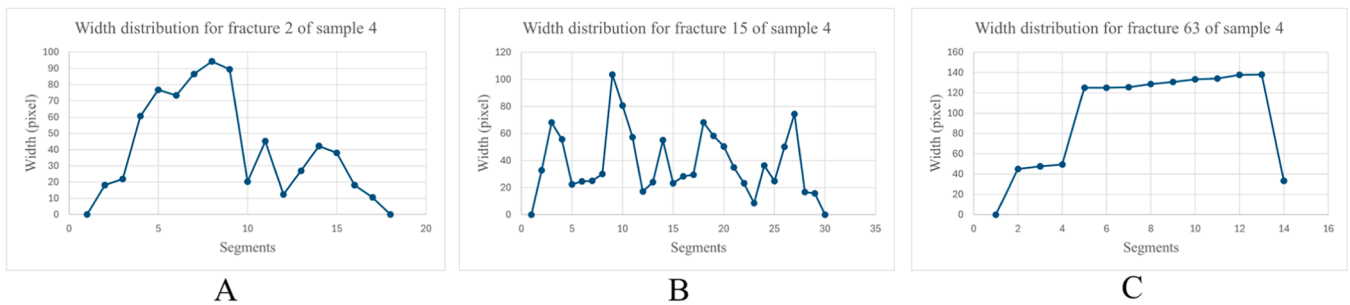


Fig. 25. Width distribution for highlighted fractures of sample 4 in a) fracture number 2, b) fracture number 15, c) fracture number 63.

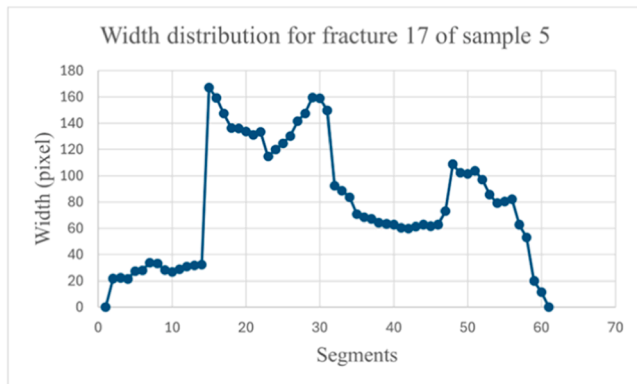


Fig. 26. Width distribution for highlighted fractures of sample 54 in fracture number 17.

highly anisotropic environment can introduce bias and produce inaccurate results, as observed in sample image 4 in this study. In this case, one possible solution is to divide the images into subimages to minimise the effects of anisotropy. Consistency in the obtained traces, particularly for long fractures with complex shapes, is not guaranteed in such a condition and may require further processing, auxiliary points [62,63], or additional conditions [23] to connect separated lines. However,

approaching a fracture first as a contour and then as a trace simplifies this issue, and this approach forms the basis of the OpenFraQ algorithm. Not all traces are equally significant; thus, establishing criteria, such as length and width, for ranking is important. Furthermore, the proposed algorithms for open fracture characterisation are primarily limited to laboratory settings and specific conditions, such as resin injection, weathering removal, and the presence of UV light, whereas OpenFraQ is intended for outcrop characterisation.

This work focuses on geometric fractures, which are particularly relevant in mining, geoenvironment, geohazard assessment, and slope stability analysis. However, the exact shape of the aperture is essential in some fields, such as hydrogeological studies [11,64], but it is not essential in other fields, like geomorphological investigation. This algorithm is designed based on geomorphological demands and emphasises the characterisation of open fractures, specifically focusing on their openness and overall trace or direction characterisation. One obvious problem with current fracture detection algorithms is that they do not distinguish between open fractures and non-open fractures, an issue that is addressed in this work.

Since this work primarily serves as a test of our approach rather than a comprehensive investigation, scale is not a focus of this study. In real cases, a transformation function could convert digital images from an image coordinate system to a physical coordinate system, ensuring accurate measurements while considering perspective errors. Hence, the overall perpendicular direction of the lens relative to the fracture, along with clear, minimally noisy images, is preferred. Including a scale object

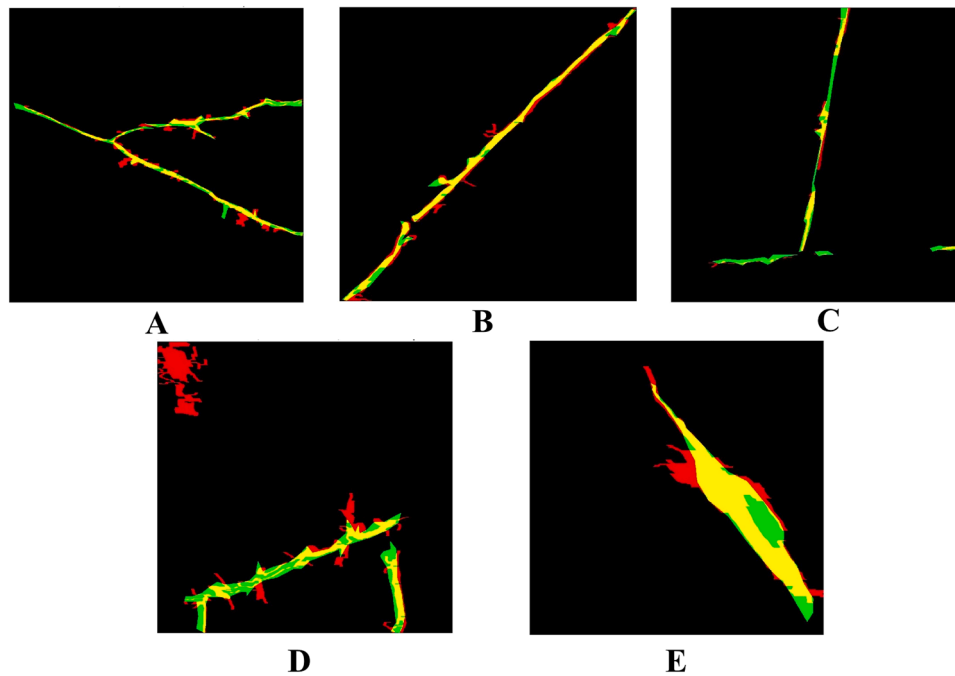


Fig. 27. Comparison of open fracture detection by OpenFraQ and manual annotation in (a) sample 1, (b) sample 2, (c) sample 3, (d) sample 4, (e) sample 5, and (f) sample 6. Manual detection is shown in green, OpenFraQ results in red, and overlapping regions in yellow.

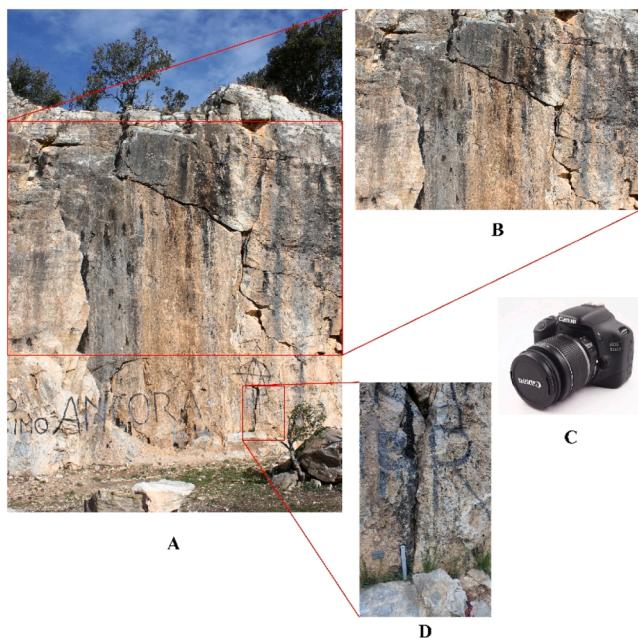


Fig. 28. Field setup and sample selection for automated fracture aperture quantification using OpenFraQ. (A) Outcrop from an abandoned mine in the Tuscany region; (B) selected window for OpenFraQ algorithm testing; (C) DSLR camera (Canon 550D model) used for image acquisition; (D) ruler placed adjacent to written letters on the outcrop for pixel size calibration.

within the image can also assist in obtaining precise measurements. Indeed, by capturing digital images of rock faces and applying photogrammetry techniques, it is possible to analyse discontinuity geometry without requiring physical access to the rock surfaces. Moreover, the proposed destructive methods suggest that for aperture detection, the weathered area should be removed; however, this is impossible in inaccessible areas. On the other hand, the contour detection-based environment of OpenFraQ enables it to include highly weathered

areas as part of the fracture, while adjusting parameters can modify the margin according to the specific purpose of the study.

The analysis indicates that OpenFraQ reliably detects the primary fracture traces in images, providing results that are consistent with manual detections, whereas the modified Hough transform in some cases produces extraneous lines or fails to capture key features. One can justify the number of significant open fractures (k) by evaluating the distribution of detected contours' areas and widths. Across multiple test images, OpenFraQ consistently outperforms the modified Hough transform in accuracy and reliability, successfully identifying complex and irregular fracture patterns, with results closely matching manual mapping, thus validating its efficacy for geomorphological analysis and modelling. Moreover, based on the bounding box concept, its width can provide an estimate of the extent of fracture influence around its width. Furthermore, reviewing the aperture definition suggests that dividing the obtained area of each fracture by its length can offer an estimate of the average width. Measuring the exact width and its variation along the middle line, as discussed in [3,11–14], is not within the scope of this work, since this study focuses on modelling rather than precise measurement. Additionally, utilising publicly available datasets minimises the bias that could arise from using overly ideal or perfect test images.

It is assumed that the input image, which contains multiple fractures, is captured in an isotropic environment, as anisotropy could influence the results. The chosen parameters may not apply to all fractures, so they may need to be adjusted for each fracture or set of fractures. Notably, the widest aperture does not always correspond to the largest fracture, offering insights into the fracture's history and the forces involved; for example, fracture orientation and aperture indicate the local stress field, with compressive forces reducing aperture and tensile forces increasing it, where compression is positive and tension negative [3]. Moreover, this algorithm considers noise issues by first preprocessing the interval image, applying a greyscale transformation and Gaussian smoothing, and then isolating each fracture separately using the bounding box method. Significant parameter adjustments can enhance preprocessing for cases with high noise levels. If additional preprocessing beyond what is described here is necessary, further measures can be implemented as recommended in standard image processing manuals [35,65,66].

Although OpenFraQ achieved higher average precision (63.34%),

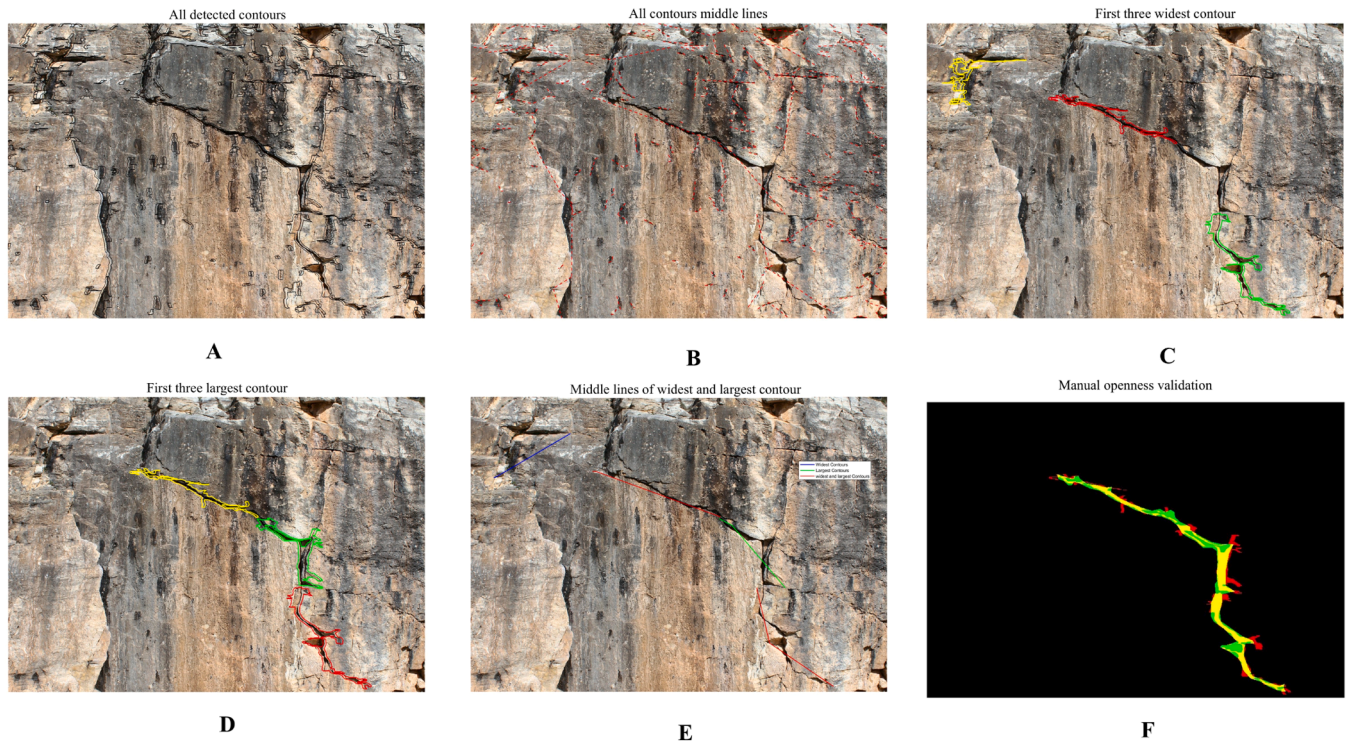


Fig. 29. OpenFraQ Results for sample 6; A: all detected contours (open fractures); B: all contour middle lines; C: first two widest contours; D: first two largest contours; E: middle lines related to first two widest and largest contours; and F: manual validation of openness results (manual detection in green, OpenFraQ results in red, and overlapping regions in yellow).

recall (63.05%), and F1-score (62.92%) compared to Byun et al. (2021), whose deep learning approach reported precision (53.2%), recall (61.4%), and F1-score (57.0%) on the same dataset with the CNN method, a true comparison is not possible. This is because the presence of vegetation in many images from the Byun dataset prevented us from using all available data, and only selected images with minimal vegetation were analysed. While validation metrics are higher with almost simple shapes, such as those in sample 2 (precision=80.11), the challenging dataset selected for this study demonstrates the algorithm's robustness but also highlights that additional progress can be made, especially in complex geological conditions. In the future, the integration of deep learning methods such as convolutional neural networks (CNNs) could further improve fracture detection by effectively handling vegetation and other complex features, thereby enhancing the algorithm's performance in challenging geological settings.

Recent advances in machine learning offer segmentation-based automated detection of fractures and non-fractures [14,16,19,27,28], as well as fracture characterization [27]. However, gaps remain in achieving more advanced characterization and discrimination between open and non-open fractures, along with upgrading to the identification of fracture boundaries, fracture areas, and exact or overall fracture traces according to project needs, especially in complex geological settings. Integrating these approaches with geometric methods like OpenFraQ may further enhance accuracy and adaptability across diverse field conditions. In future work, combining machine learning with high-resolution field imagery from UAVs (unmanned aerial vehicles) and 3D modelling workflows [29,30,67] could enable comprehensive analysis and quantification of fracture networks in both laboratory and field environments. Future research could focus on applying this algorithm to monitor fracture evolution over time, leveraging its automation to enhance accuracy, reduce processing time, and minimise errors, particularly when analysing large datasets. As demonstrated in Table 4, OpenFraQ's computational speed is comparable to that of the modified Hough transform and significantly faster than manual tracing, which

relies heavily on geological interpretation and is limited to identifying dominant fractures. Further advancements should include applying the algorithm to georeferenced cases and comparing the results with field measurements to validate its accuracy. Incorporating constraints to distinguish between allowable marginal weathered zones and precise fracture locations would also be beneficial. Developing a 3D adaptation of this method is highly recommended, as apertures are inherently three-dimensional structures, and capturing their full surfaces could provide more detailed and comprehensive insights into fracture geometry and behaviour.

5. Conclusion

- OpenFraQ is a novel automated algorithm for geometric fracture characterisation in rock outcrops, enabling rapid and robust detection using advanced image processing techniques, especially in inaccessible or hazardous areas.
- The novelty of OpenFraQ lies in its ability to quantify fractures as both contours and traces from 2D images, unlike previous automatic or semi-automatic approaches that considered fractures only as trace lines or segmented pixels.
- The algorithm uniquely distinguishes open fractures from non-open cracks, enhancing fracture detection accuracy compared to existing methods.
- OpenFraQ achieved average precision (63.34%), recall (63.05%), and F1-score (62.92%) against the ground truth.
- Qualitatively, OpenFraQ provides more accurate and consistent fracture detection than the modified Hough transform, especially for complex and irregular networks.
- Validation metrics were highest almost for simple fracture shapes, but the use of a challenging dataset demonstrated both the robustness of the algorithm and areas for further improvement in complex geological conditions.

- Future integration with deep learning techniques and 3D modelling workflows could further enhance detection accuracy and expand its applicability to more difficult scenarios.

CRedit authorship contribution statement

Maryam Esmaeili: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Tommaso Beni:** Writing – review & editing. **Giovanni Gigli:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The publication has been funded by EU - Next Generation EU Mission 4, Component 2 - CUP B53C22002150006 - Project IR0000032 – ITIN-ERIS - Italian Integrated Environmental Research Infrastructures System.

Data availability

Data will be made available on request.

References

- [1] A. Garavand, V.M. Podgornov, Hydraulic fracture optimization by using a modified pseudo-3D model in multi-layered reservoirs, *J. Nat. Gas. Geosci.* 3 (2018) 233–242, <https://doi.org/10.1016/j.jnggs.2018.08.004>.
- [2] A.A. Keller, P.V. Roberts, M.J. Blunt, Effect of fracture aperture variations on the dispersion of contaminants, *Water. Resour. Res.* 35 (1999) 55–63, <https://doi.org/10.1029/1998WR900041>.
- [3] Z. Pouraskarparast, H. Aghaei, L. Colombero, E. Masoero, M. Ghaedi, Fracture aperture: a review on fundamental concepts, estimation methods, applications, and research gaps, *Mar. Pet. Geol.* 164 (2024) 106818, <https://doi.org/10.1016/j.marpetgeo.2024.106818>.
- [4] F. Zhao, W. Wang, Rock fracture aperture characterization based on corner segment algorithm in image analysis, in: 2006 International Conference on Computational Intelligence and Security, 2006, pp. 1004–1007, <https://doi.org/10.1109/ICCIAS.2006.295414>.
- [5] T.S. Miranda, R.F. Santos, J.A. Barbosa, I.F. Gomes, M.L. Alencar, O.J. Correia, T.C. Falcão, J.F.W. Gale, V.H. Neumann, Quantifying aperture, spacing and fracture intensity in a carbonate reservoir analogue: crato Formation, NE Brazil, *Mar. Pet. Geol.* 97 (2018) 556–567, <https://doi.org/10.1016/j.marpetgeo.2018.07.019>.
- [6] C. Klimczak, R.A. Schultz, R. Parashar, D.M. Reeves, Cubic law with aperture-length correlation: implications for network scale fluid flow, *Hydrogeol. J.* 18 (2010) 851–862, <https://doi.org/10.1007/s10040-009-0572-6>.
- [7] M. Mauldon, W.M. Dunne, M.B. Rohrbach Jr, Circular scanlines and circular windows: new tools for characterizing the geometry of fracture traces, *J. Struct. Geol.* 23 (2001) 247–258, [https://doi.org/10.1016/S0191-8141\(00\)00094-8](https://doi.org/10.1016/S0191-8141(00)00094-8).
- [8] M. Antonellini, P.N. Mollema, Outcrop fracture network characterization for unraveling deformation sequence, geomechanical properties distribution, and slope stability in a flysch sequence (Monte Venere Formation, Northern Apennines, Italy), *Int. J. Earth. Sci.* 108 (2019) 735–751, <https://doi.org/10.1007/s00531-019-01685-y>.
- [9] L. Zeng, X. Tang, J. Qi, L. Gong, F. Yu, T. Wang, Insight into the cenozoic tectonic evolution of the Qaidam Basin, Northwest China from fracture information, *Int. J. Earth. Sci.* 101 (2012) 2183–2191, <https://doi.org/10.1007/s00531-012-0779-y>.
- [10] L. Jing, O. Stephansson, Fundamentals of Discrete Element Methods for Rock engineering: Theory and Applications, Elsevier, 2007.
- [11] L. Li, W. Wang, M. Wang, J. Chen, A. Khatoun, Rock aperture image acquisition and processing based on mechanics moment, Steger theory and fractal analysis, *Geoenery Sci. Eng.* 230 (2023) 212236, <https://doi.org/10.1016/j.geoen.2023.212236>.
- [12] W. Xu, X. Li, Y. Zhang, X. Wang, R. Liu, Z. He, J. Fan, Aperture measurements and seepage properties of typical single natural fractures, *Bull. Eng. Geol. Environ.* 80 (2021) 8043–8058, <https://doi.org/10.1007/s10064-021-02392-2>.
- [13] S. Zhao, Z. Zhao, A comparative study of landslide susceptibility mapping using SVM and PSO-SVM models based on grid and slope units, *Math. Probl. Eng.* 2021 (2021) 1–15, <https://doi.org/10.1155/2021/8854606>.
- [14] H. Byun, J. Kim, D. Yoon, I.-S. Kang, J.-J. Song, A deep convolutional neural network for rock fracture image segmentation, *Earth Sci. Inf.* 14 (2021), <https://doi.org/10.1007/s12145-021-00650-1>.
- [15] W. Weixing, W. Fengping, H. Xiaojun, S. Junfang, Rock fracture image acquisition using two kinds of lighting and fusion on a wavelet transform, *Bull. Eng. Geol. Environ.* 75 (2015), <https://doi.org/10.1007/s10064-015-0747-4>.
- [16] X. Zhao, X. Wang, Unsupervised domain adaptation-based fracture segmentation method for core CT images, *Expert. Syst. Appl.* 264 (2025) 125857, <https://doi.org/10.1016/j.eswa.2024.125857>.
- [17] S. Gentier, D. Billaux, L. van Vliet, Laboratory testing of the voids of a fracture, *Rock. Mech. Rock. Eng.* 22 (1989) 149–157, <https://doi.org/10.1007/BF01583959>.
- [18] E.V.A. Hakami, E. Larsson, Aperture measurements and flow experiments on a single natural fracture, *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, Elsevier, 1996, pp. 395–404.
- [19] M. Esmaeili, T. Beni, G. Gigli, V. Tofani, Rock mass exposure fracture detection through 2D close-range images using image processing techniques: a review, *Earth Sci. Inf.* 18 (2025) 1–29, <https://doi.org/10.1007/s12145-025-01980-0>.
- [20] R. Palamakumbura, M. Krabbendam, K. Whitbread, C. Arnhardt, Data acquisition by digitizing 2-D fracture networks and topographic lineaments in geographic information systems: further development and applications, *Solid Earth* 11 (2020) 1731–1746, <https://doi.org/10.5194/se-11-1731-2020>.
- [21] A. Marques, G. Racolte, D.C. Zanotta, E. Menezes, C.L. Cazarin, L. Gonzaga, M. R. Veronez, Adaptive segmentation for discontinuity detection on karstified carbonate outcrop images from UAV-SfM acquisition and detection bias analysis, *IEEe Access.* 10 (2022) 20514–20526, <https://doi.org/10.1109/ACCESS.2022.3151897>.
- [22] D. Healy, R. Rizzo, D. Cornwell, N. Farrell, H. Watkins, N. Timms, E. Gomez-Rivas, M. Smith, FracPaQ: a MATLAB™ toolbox for the quantification of fracture patterns, *J. Struct. Geol.* 95 (2016), <https://doi.org/10.1016/j.jsg.2016.12.003>.
- [23] W. Zhang, Y. Liang, Rock fracture centerline extraction based on Hessian Matrix and Steger algorithm, *KSII Trans. Internet Inf. Syst.* 9 (2015) 5073–5086, <https://doi.org/10.3837/tiis.2015.12.018>.
- [24] B. Chudasama, N. Ovaskainen, J. Tamminen, N. Nordbäck, J. Engström, I. Aaltonen, Automated mapping of bedrock-fracture traces from UAV-acquired images using U-Net convolutional neural networks, *Comput. Geosci.* 182 (2024) 105463, <https://doi.org/10.1016/j.cageo.2023.105463>.
- [25] H. Xu, S. Tang, J. Wang, B. Dong, X. Wang, K. Zhao, Y. Zhu, J. Geng, Rock fracture identification algorithm based on the confidence score and non-maximum suppression, *Bull. Eng. Geol. Environ.* 83 (2024) 213, <https://doi.org/10.1007/s10064-024-03710-0>.
- [26] S. Wang, J. Yin, Z. Pi, W. Cao, X. Cai, Z. Zhou, Automatic detection and characterization of discontinuity traces and rock fragment size distribution using a digital image processing method, *Meas.* 128 (2024) 114343, <https://doi.org/10.1016/j.measurement.2024.114343>.
- [27] Y. Ji, S. Song, W. Zhang, Y. Li, J. Xue, J. Chen, Automatic identification of rock fractures based on deep learning, *Eng. Geol.* 345 (2025) 107874, <https://doi.org/10.1016/j.enggeo.2024.107874>.
- [28] N. Li, Z. Xiong, L. Wang, B. Dai, S. Zhao, H. Ye, Q. Wang, A multi-attention deep learning network for intelligent identification of rock mass fracture in mines, *Results. Eng.* 26 (2025) 105023, <https://doi.org/10.1016/j.rineng.2025.105023>.
- [29] M. Torkan, M. Janiszewski, L. Uotinen, A. Baghbanan, M. Rinne, High-resolution photogrammetry to measure physical aperture of two separated rock fracture surfaces, *J. Rock Mech. Geotech. Eng.* 16 (2024) 2922–2934, <https://doi.org/10.1016/j.jrmge.2023.10.003>.
- [30] T.D. Seers, D. Hodgetts, Extraction of three-dimensional fracture trace maps from calibrated image sequences, *Geosphere* 12 (2016) 1323–1340, <https://doi.org/10.1130/GES01276.1>.
- [31] D. Deb, S. Hariharan, U.M. Rao, C.-H. Ryu, Automatic detection and analysis of discontinuity geometry of rock mass from digital images, *Comput. Geosci.* 34 (2008) 115–126, <https://doi.org/10.1016/j.cageo.2007.03.007>.
- [32] A. Karnieli, A. Meisels, L. Fisher, Y. Arkin, Automatic extraction and evaluation of geological linear features from digital remote sensing data using a Hough transform, *Photogramm. Eng. Remote Sens.* 62 (1996) 525–531.
- [33] MathWorks, MATLAB version 24.2.0 (R2024b). <https://www.mathworks.com/products/matlab.html>, 2024.
- [34] G. Deng, L.W. Cahill, An adaptive Gaussian filter for noise reduction and edge detection, in: 1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference 3, 1993, pp. 1615–1619, <https://doi.org/10.1109/NSSMIC.1993.373563>.
- [35] R. Gonzalez, R. Woods, Digital image processing (3rd Edition), 2006.
- [36] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8 (1986) 679–698, <https://doi.org/10.1109/TPAMI.1986.4767851>.
- [37] S. Gupta, Image edge detection: a review, *Int. J. Adv. Res. Comput. Eng. Technol.* IJARCET 2 (7) (2013). July 2013 2, <https://api.semanticscholar.org/CorpusID:212440552>.
- [38] D. Rana, S. Dalai, Review on traditional methods of edge detection to morphological based techniques, *Int. J. Comput. Sci. Inf. Technol.* 5 (2014) 5915–5920.
- [39] J.P.F. Serra, Image analysis and mathematical morphology. <https://api.semanticscholar.org/CorpusID:62066269>, 1983.
- [40] W. Wang, Image analysis of particles by modified Ferret method—Best-fit rectangle, *Powder. Technol.* 165 (2006) 1–10, <https://doi.org/10.1016/j.powtec.2006.03.017>.

- [41] A. Toshev, B. Taskar, K. Daniilidis, Shape-based object detection via boundary structure segmentation, *Int. J. Comput. Vis.* 99 (2012) 123–146, <https://doi.org/10.1007/s11263-012-0521-z>.
- [42] V. Lempitsky, P. Kohli, C. Rother, T. Sharp, Image segmentation with a bounding box prior, in: 2009 IEEE 12th International Conference on Computer Vision (ICCV), 2009, pp. 277–284, <https://doi.org/10.1109/ICCV.2009.5459262>.
- [43] P. Groves, S. Saha, P. Bajcsy, Boundary information storage, retrieval, georeferencing and visualization, Technical Report NCSA-ALG-03-0001, 2003.
- [44] B.R. Kang, H. Lee, K. Park, H. Ryu, H.Y. Kim, BshapeNet: object detection and instance segmentation with bounding shape masks, *Pattern. Recognit. Lett.* 131 (2020) 449–455, <https://doi.org/10.1016/j.patrec.2020.01.024>.
- [45] O.L.F. de Carvalho, O.A.d.C. Júnior, A.O. de Albuquerque, N.C. Santana, R. F. Guimarães, R.A.T. Gomes, D.L. Borges, Bounding box-free instance segmentation using semi-supervised iterative learning for vehicle detection, *IEEe J. Sel. Top. Appl. Earth. Obs. Remote Sens.* 15 (2022) 3403–3420, <https://doi.org/10.1109/JSTARS.2022.3169128>.
- [46] H.T. Basics, Review of Hough transform for line detection, (n.d.). <https://doi.org/10.1007/978-1-4471-4414-4>.
- [47] Notes, Standard Hough Transform I Standard Hough Transform II proposed in 1962, the Hough transform aims to estimate lines from a set of points Standard Hough transform III Standard Hough transform IV Standard Hough transform V Standard Hough transform VI radon (1962).
- [48] J. Wang, P.J. Howarth, Use of the Hough transform in automated lineament detection, *IEEe Trans. Geosci. Remote Sens.* 28 (1990) 561–567, <https://doi.org/10.1109/TGRS.1990.572949>.
- [49] O. Chutatape, L. Guo, A modified Hough transform for line detection and its performance, *Pattern. Recognit.* 32 (1999) 181–192, [https://doi.org/10.1016/S0031-3203\(98\)00140-X](https://doi.org/10.1016/S0031-3203(98)00140-X).
- [50] S. Engineering, Z. Campus, Comparison of edge detection and Hough transform techniques for the extraction of geologic features, image (Rochester, N.Y.) (2000). <https://api.semanticscholar.org/CorpusID:16374860>.
- [51] N.C. Fitton, S.J.D. Cox, Optimising the application of the Hough transform for automatic feature extraction from geoscientific images, *Comput. Geosci.* 24 (1998) 933–951.
- [52] J.M. Vermilye, C.H. Scholz, Relation between vein length and aperture, *J. Struct. Geol.* 17 (1995) 423–434, [https://doi.org/10.1016/0191-8141\(94\)00058-8](https://doi.org/10.1016/0191-8141(94)00058-8).
- [53] R.J.H. Jolly, D.J. Sanderson, Variation in the form and distribution of dykes in the Mull swarm, Scotland, *J. Struct. Geol.* 17 (1995) 1543–1557, [https://doi.org/10.1016/0191-8141\(95\)00046-G](https://doi.org/10.1016/0191-8141(95)00046-G).
- [54] D. Peacock, C. Nixon, A. Rotevatn, D. Sanderson, L. Zuluaga, Glossary of fault and other fracture networks, *J. Struct. Geol.* 92 (2016), <https://doi.org/10.1016/j.jsg.2016.09.008>.
- [55] C. Steger, An unbiased detector of curvilinear structures, *IEEe Trans. Pattern. Anal. Mach. Intell.* 20 (1998) 113–125, <https://doi.org/10.1109/34.659930>.
- [56] R. Santoso, X. He, H. Hoteit, Application of machine-learning to construct simulation models from high-resolution fractured formation, 2019. <https://doi.org/10.2118/197439-MS>.
- [57] A. Junior, G. Racolte, E. Souza, H. Domingos, R. Horota, J. Motta, D. Zanotta, C. Cazarin, L. Gonzaga da Silveira Jr, M. Veronez, Deep learning application for fracture segmentation over outcrop images from UAV-based digital photogrammetry, 2021. <https://doi.org/10.1109/IGARSS47720.2021.9553232>.
- [58] B. Kanoun, M. Cherif, I. Manighetti, Y. Tarabalka, J. Zerubia, An enhanced deep learning approach for tectonic fault and fracture extraction in very high resolution optical images, 2022. <https://doi.org/10.1109/ICASSP43922.2022.9747007>.
- [59] T. Perol, M. Gharbi, M.A. Denolle, Convolutional neural network for earthquake detection and location, *ArXiv*. 2016 (2017) 2–10, <https://doi.org/10.1126/sciadv.1700578>.
- [60] L. Liotine, C. Wolff, E. Wyser, G. Andriani, M.-H. Derron, M. Jaboyedoff, M. Parise, QDC-2D: a semi-automatic tool for 2D analysis of discontinuities for rock mass characterization, *Remote Sens.* 13 (2021) 5086, <https://doi.org/10.3390/rs13245086>.
- [61] N. Ovaskainen, fractopo: a Python package for fracture network analysis, *J. Open. Source Softw.* 8 (2023) 5300.
- [62] Y. Vasuki, E.-J. Holden, P. Kovesi, S. Micklethwaite, Semi-automatic mapping of geological structures using UAV-based photogrammetric data: an image analysis approach, *Comput. Geosci.* 69 (2014), <https://doi.org/10.1016/j.cageo.2014.04.012>.
- [63] L. Guo, J. Liao, F. Zhong, K. Dong, X. Liu, J. Tao, J. Chen, A semi-automated methodology for rock mass discontinuity trace map construction, in: 2012 International Conference on Image Analysis and Signal Processing, IEEE, 2012, pp. 1–5. <https://api.semanticscholar.org/CorpusID:14913780>.
- [64] T. Xu, F. Liang, Machine learning for hydrologic sciences: an introductory overview, *Wiley Interdiscip. Rev.: Water* 8 (2021) e1533, <https://doi.org/10.1002/wat2.1533>.
- [65] R.A. Poldrack, T. Nichols, J. Mumford, Image processing basics, *Handb. Funct. MRI Data Anal.* (2011) 13–33, <https://doi.org/10.1017/cbo9780511895029.003>.
- [66] A.K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, 1989. <https://books.google.it/books?id=3ilCPgAACAAJ>.
- [67] D. Pan, Y. Li, X. Wang, Z. Xu, Intelligent image-based identification and 3-D reconstruction of rock fractures: implementation and application, *Tunn. Undergr. Space Technol.* 145 (2024) 105582, <https://doi.org/10.1016/j.tust.2023.105582>.