Improved fracture segmentation from unwrapped drill-core images using an innovative two-stage segmentation approach

Fatimah Alzubaidi[1], *Harikrishnan* Nalinakumar[1], *Stuart* R. Clark[1], Jan Erik Lie[2], *Peyman* Mostaghimi[1], and *Ryan* T. Armstrong[1,*]

[1]School of Minerals and Energy Resources Engineering, University of New South Wales, Sydney, Australia.
[2]Lundin Energy Norway AS, Lysaker, Norway.

Abstract. Recent advances in drill-core scanning provide high-quality unwrapped core images that require similarly advanced methods for processing the produced images and extracting important information in an automatic and reliable manner. Unwrapped core images provide a detailed view of fractures that can be segmented and characterized in terms of depth, dip angle, dip direction, aperture, and roughness. Despite the availability of advanced machine learning techniques for instance segmentation, obtaining accurate segmentation of the fractures from high-resolution images is challenging. For Mask Region-based Convolutional Neural Network (Mask R-CNN), a common instance segmentation model, the capacity of the segmentation branch is limited which, in turn, limits the segmentation quality of large objects especially those with fine details, such as the fractures. We propose a two-stage segmentation approach using Mask R-CNN to improve fracture segmentation. We use two CNN models: the first model processes full-size unwrapped core images to detect and segment fractures; the second model performs a more detailed segmentation by processing smaller regions of the images that include the fractures detected by the first model. We investigate using the standard Mask R-CNN architecture and a new architecture of Mask R-CNN with a Point-based Rendering (PointRend) neural network module. The method is evaluated on ~47 m of core and results in an increase in the average intersection-over-union by 0.25 from the baseline (one-stage segmentation with standard Mask R-CNN).

1 Introduction

Unwrapped core images are two-dimensional representations of the core surface. The images are taken by scanning the core in a 360° mode, i.e., by rotating the core and capturing the surface from different angles [1–3]. Modern core scanners provide high-resolution unwrapped core images with resolution up to 40 pixel/mm and can scan approximately three meters of core in 1 minute [2]. The images provide a detailed view of the core surface and the structural features, such as fractures.

Fractures appearing in unwrapped core images are involved in a variety of fracture analysis practices [4, 5]. The fractures, especially those with high dips, are used as matching features during core re-orientation by correlating them with the fractures from borehole image logs, allowing for a low-cost and indirect core orientation method [1]. Moreover, integrating unwrapped core images and borehole image logs (acoustic or resistivity images) are necessary for accurate fracture characterization [3].

In recent work [6], fractures from high-resolution unwrapped core images were used to automatically estimate the degree of fracture roughness and mismatches between fracture edges. The authors computed the joint roughness coefficient (JRC) for each fracture based on statistical estimators. The mismatch was based on the standard deviation of fracture apertures and the correlation between fracture profiles. They validated the results according to a standard scale of the JRC [7]. The study, however, used pre-digitized fractures that were traced manually to obtain fracture profiles.

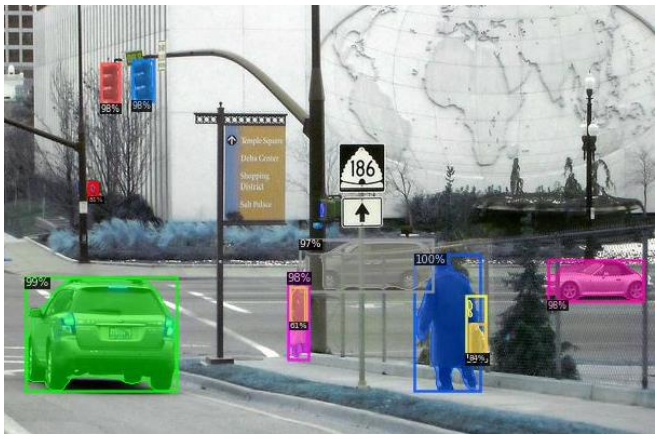
Developing automated and accurate fracture segmentation from unwrapped images can significantly facilitate the characterization process. Accurate and reliable fracture segmentation can be particularly beneficial for detailed fracture analyses, such as estimating fracture roughness or aperture sizes. Such segmentation approaches have not been developed yet due to there being limited research in this area. Previous work [7, 8] used classic image processing algorithms to identify fractures from core tray images, which provided only a limited view of the fracture surface compared to unwrapped images. In addition, several methods have been established to detect fractures from borehole image logs, which range from using edge detection and image processing algorithms to using machine learning (ML) based methods [9–12] with variable results. Thus, more research is needed to explore the feasibility of developing automated and accurate fracture segmentation from unwrapped core images.

Modern ML approaches can provide high-quality image analysis including instance or semantic segmentation [14, 15]. Instance segmentation is distinctly different from the more commonly used semantic segmentation in digital core analysis. With semantic segmentation objects of the same

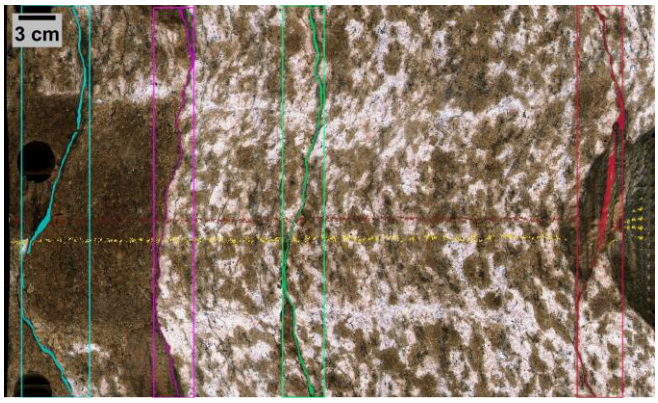
* Corresponding author: ryan.armstrong@unsw.edu.au

category are segmented as a single object while for instance segmentation objects are detected separately. An example application for instance segmentation, using the Mask Region-based Convolutional Neural Network (Mask R-CNN), is provided in Figure 1a where features in a street-level image are identified and segmented. Such technologies are an important step towards the operation of autonomous vehicles. So, can this technology be translated to core analysis applications? The aim would be to develop an instance segmentation algorithm that identifies important core structural features. The focus of our work is displayed in Figure 1b where Mask R-CNN is applied to core images for fracture identification. With this technology fractures could be segmented and characterized on a per fracture basis in an automated way. The efficacy of the method for core analysis, however, remains to be tested.

The accuracy of object segmentation by ML models depends on the size of the object compared to the resolution of the output raw mask, which is usually limited by computational and memory requirements (e.g. [14]). In high-resolution core images, the fractures appear as large objects that span the entire width of the image, but their apertures are finer features that require high-resolution images. The segmentation of such features is directly affected by the trade-off between field-of-view (computational and memory requirements) versus image resolution. We propose that the segmentation of fractures from high-resolution core images can be enhanced using a two-stage segmentation method.



(a)



(b)

Fig. 1. Original application of Mask R-CNN for natural image from COCO test dataset [15] in (a), and the application of the model for fracture segmentation from unwrapped core images in (b), showing result from two-stage segmentation using Mask R-CNN + PointRend.

Two-stage segmentation has been applied in various applications where there are multi-scale features present in an image. [16] developed a two-stage segmentation framework using two 3D U-Nets to enhance segmentation of organs from computed tomography images. The first U-Net segments the whole image and the second U-Net segments a small part of the image that includes the organ of interest detected by the first model. A similar approach was used by [17] to refine segmentation of ultrasound breast images, which resulted in an improved average Dice score, a segmentation metric [18], of up to 14% approximately. Alternatively, [19] introduced a Point-based Rendering (PointRend) module to produce high-resolution masks from semantic segmentation and instance segmentation models with only a minor increase in the memory requirement.

Herein, we evaluated the standard Mask R-CNN and Mask R-CNN with the PointRend module (Mask R-CNN + PointRend) in a two-stage segmentation approach. The first stage detects full-size fractures in the input image. The second stage provides finer segmentation by processing small regions of the image that include detected fractures. We train the models with images from two boreholes and evaluated the method on new data including images from new boreholes. The results are evaluated using the intersection-over-union (IoU) between manual and predicted segmentation of fractures in new core images.

2 Data and methodology

This section is divided into two sub-sections. Section 2.1 provides a description of the unwrapped core images used in our work and the preparation of the datasets for training and testing the segmentation models. Section 2.2 introduces the proposed two-stage procedure, the tested architectures of the instance segmentation models, the training process of the models, and the evaluation metrics used.

2.1 Data preparation

The core images were obtained from four boreholes located at the Norwegian Barents Sea and North Sea. The boreholes were 7220/6-1, 7220/11-3 16/2-17 B, and 16/2-18 S. The images were acquired from Lundin Energy [20].

Boreholes 7220/6-1 and 7220/11-3 are located on the Loppa High in the Barents Sea whereas boreholes 16/2-17 B and 16/2-18 S are from Johan Sverdrup field and are located on the Utsira High in the North Sea. Core intervals and net lengths of the images used in our work from each borehole are summarised in Table 1. The sections from boreholes 7220/6-1 and 7220/11-3 were from the Ørn formation that is dominated by carbonate rock while the sections from 16/2-17 B and 16/2-18 S were taken from the Asgard formation composes mainly of claystones, marlstones, and limestone, and from the basement group that comprises highly fractured granite [21].

Table 1. Details of the core images from each borehole, showing images resolution, depth intervals and the net lengths of the core used in the study.

Borehole	Image resolution (pixels/mm)	Core images interval (m)	Net core length (m)
7220/6-1	2.6	1149.00–1203.00	52.88
7220/11-3	9.0	1852.20–1944.00	47.19
16/2-17 B	9.0	2130.70–2140.37	5.40
16/2-18 S	9.0	2130.70–2140.37	8.21

The training and validation images were selected from two boreholes as detailed in Table 2, with an initial number of 100 images that included 204 fractures. We used data augmentation to expand this number to 1200 images that were divided into 1000 images for training and 200 images for validation. The augmentation included blurring, altering the brightness and contrast, and horizontal and vertical flipping.

For evaluating the method, the test images were selected from the four boreholes (two blocks) to include the images from boreholes 16/2-17 B and 16/2-18 S obtained from new regions and formations. The test dataset included a total of 72 images that have 160 fractures (Table 2).

Table 2. Number and core length of training, validation, and test images prior to data augmentation.

	Training & validation		Test	
Borehole	Number	Length (m)	Number	Length (m)
7220/6-1	35	33.33	17	13.86
7220/11-3	65	33.60	30	19.28
16/2-17 B	——	——	11	5.39
16/2-18 S	——	——	14	8.21
Total	100	66.93	72	46.74

Fractures in the training, validation and test images were manually labelled using the Supervisely annotation online tool [22]. The images were 811–2905 pixels high and 356–8980 pixels wide with resolution of 2.6–9 pixels/mm as shown in Table 1. To build the model for the second stage of segmentation, we created three smaller images from each fracture in the training and validation datasets following a similar procedure as explained in Section 2.2.1. The resulting images were 144–1057 pixels high and 74–3457 pixels wide.

Before inputting the images into the models during each stage of segmentation, the images were resized to the default size of Mask R-CNN (discussed in Section 2.2.3). After resizing, the full-size images (i.e., the inputs of the first segmentation stage) had an average resolution of 2.1 pixels/mm while the small images (i.e., the inputs of the second segmentation stage) had a higher average resolution of 11.2 pixels/mm.

2.2 Methodology

2.2.1 Two-stage approach

The overall procedure involved using two instance segmentation models, as displayed in Figure 2. Model-A received a full-size image and output a mask for each fracture, like the ordinary implementation of an instance segmentation model. Model-B was then used to segment small regions of the image to enhance the masks produced by model-A.

As demonstrated in Figure 2, three regions were cropped from each fracture and segmented by model-B. The cropping included an overlapping ratio of 5% of the fracture height. We also tested dividing the fracture into four regions instead of three, but it did not provide further improvement from the three-region division.

Next, the resulting high-resolution masks by model-B at each location of the fracture replaced the original mask from model-A at that location. In some cases, model-B yielded more than one mask for a single image when it included parts of nearby fractures. To decide on the correct mask for such images, we used the mask with the highest IoU with the primary mask from model-A at the corresponding location, as shown in Figure 3. See Section 2.2.4 for the definition of IoU.

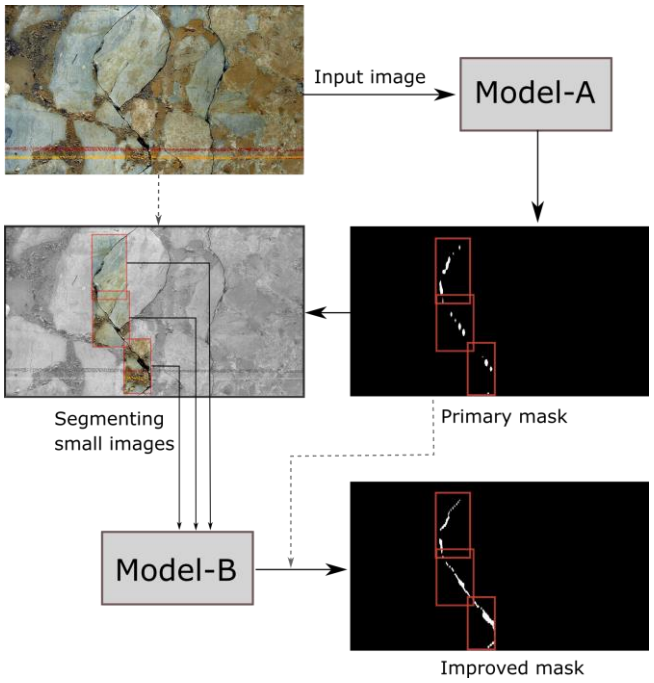


Fig. 2. Workflow of the two-stage segmentation method using two instance segmentation models (for clarity only a single continuous fracture segment is shown). Solid lines show the main flow and dashed lines show additional transitions.

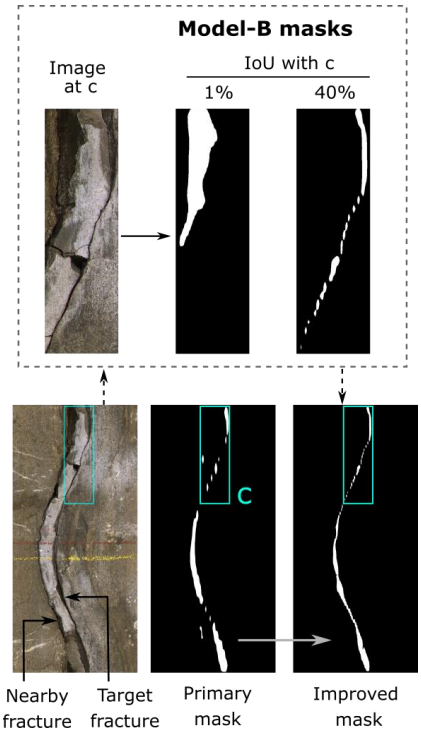


Fig. 3. Process of updating primary masks from model-A showing a case of multiple outputs by model-B. The mask with the highest IoU (40%) was selected to update region C in the primary mask instead of the mask with 1% IoU that belongs to a nearby fracture.

2.2.2 Experimented architectures of Mask R-CNN

To define the architecture of model-A and model-B, we evaluated two recognized instance segmentation architectures: (1) standard Mask R-CNN and (2) Mask R-CNN + PointRend.

The main components of the Mask R-CNN architecture are the region proposal network (RPN), a deep CNN for feature extraction (backbone), the RoIAlign layer, followed by the branches for bounding box detection, classification, and mask segmentation [14].

The purpose of the RPN is to propose candidate object regions, i.e., anchors, in the input image of different sizes and height-to-width ratios. The RoIAlign layer crops features extracted by the backbone of the proposed regions and resizes them to a uniform shape, e.g., 7 × 7 or 14 × 14 pixels^2. Then the detection and classification branches yield a classification score and bounding box coordinates for each region of interest (RoI). In parallel, the segmentation branch outputs a binary mask for each RoI.

A common architecture of Mask R-CNN includes a backbone with a feature pyramid network (FPN) [23] like what was used in this study. In this architecture, the segmentation branch receives RoIs of 14 × 14 pixels^2 and yields 28 × 28 pixels^2 masks. The branch consists of four convolutional layers, a deconvolutional layer, and an output layer.

Due to the small size of the output masks, segmentation of large objects has limited accuracy and fine details at the object boundaries are usually not segmented properly. However, increasing the capacity of the segmentation branch would require significantly more computational time and memory. Therefore, an alternative method is required.

The PointRend module was developed by [19] to refine the segmented masks generated by instance segmentation models, such as Mask R-CNN, with a minor increase in the memory requirement. In Mask R-CNN, PointRend can increase the resolution of the output masks to 224 × 224 pixels^2 with nearly 30 times less memory and computational time than what would be required by the original segmentation branch to yield the same resolution. The module predicts higher-resolution masks from both lower-resolution masks as well as features of selected points from the masks by sampling points of low uncertainty. The pixel-wise prediction is performed by a multi-layer perceptron [19].

In Mask R-CNN + PointRend, the mask head predicts 7 × 7 pixels^2 for each RoI then the PointRend module refines the prediction to 224 × 224 pixels ^{2} through multiple iterations of point sampling and mask resolution enhancement. The mask resolution in each iteration is 7^2, 28^2, 56^2, 112^2, and 224^2 pixels^2, respectively. An illustration of the mask refinement process is provided in Figure A1 of Appendix A.

For both standard Mask R-CNN and Mask R-CNN + PointRend, we used ResNet-50-FPN as the backbone. We also added height-to-width ratios of 4:1 and 6:1 to the original ratios of 2:1, 1:1, and 1:2 that define the RPN anchors in model-A to match the shapes of the fractures. The rest of the architectures were left as the defaults used in [15, 19].

2.2.3 Training implementation

We used the Mask R-CNN and Mask R-CNN + PointRend implementations from Detectron2 [24]. We started the training with pretrained models on Microsoft COCO images [25]. The input images were resized to the default size of Mask R-CNN to have a maximum of 1333 pixels on the large edge [24]. We trained the models for 30,000 iterations using a batch size of 2 and 4 images for model-A and model-B, respectively using an NVIDIA GeForce RTX 2080 Ti GPU with 11GB of memory. We defined an initial learning rate of 0.01 and this rate was halved every 10,000 iterations. All other training parameters were like the default values used in Detectron2 [24].

The training loss converged as shown in Figure 4, and the precision of the segmented validation images converged as shown in Figure 5. The details of these results are discussed in Section 3. The segmentation loss was calculated as the average binary cross-entropy loss [14]. The precision was calculated using Eq. 1, where true positives are determined by a specific IoU threshold between the objects segmented by the model and the true objects. Commonly, an average precision is calculated at different IoU thresholds (0.5–0.95).

$$Precision = \frac{true\ positives}{true\ positives\ + false\ positives}. \quad (1)$$

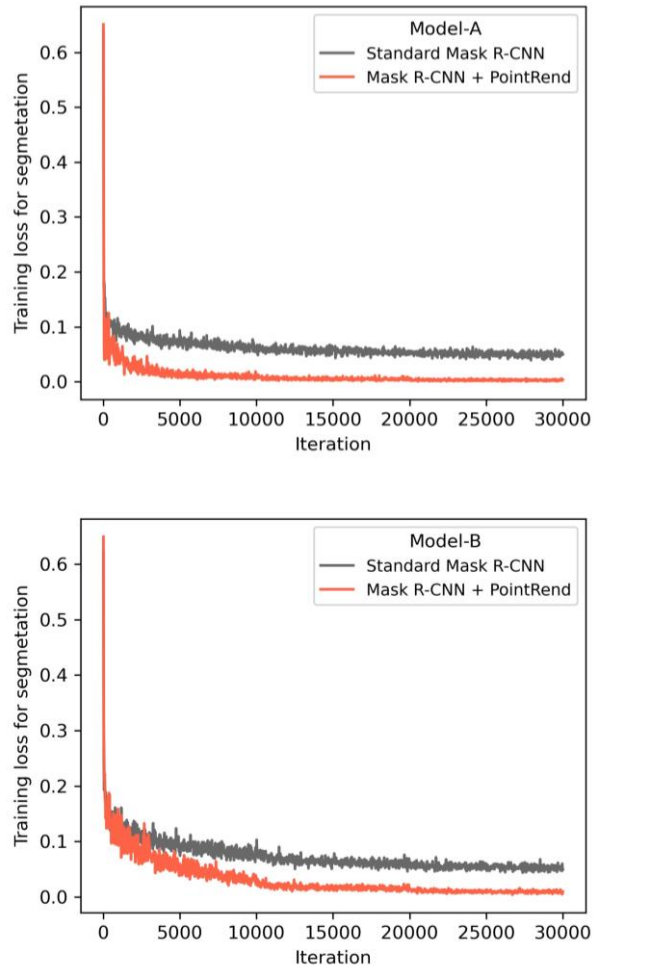


Fig. 4. Training loss for mask segmentation using standard Mask R-CNN and Mask R-CNN + PointRend architectures for model-A (top) and model-B (bottom).

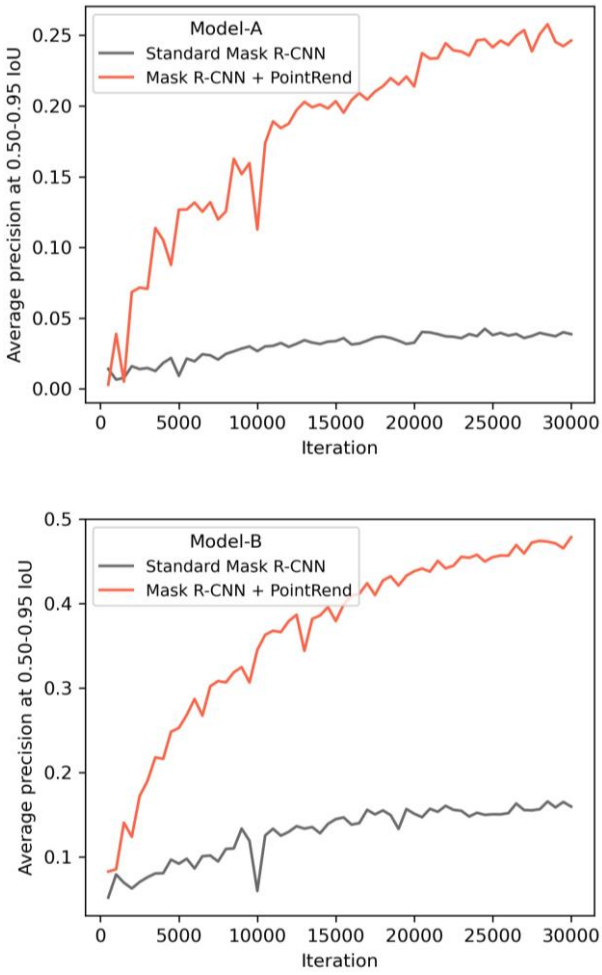


Fig. 5. Validation statistics during training of standard Mask R-CNN and Mask R-CNN + PointRend, for model-A (top) and model-B (bottom).

2.2.4 Evaluation of the two-stage method

We evaluated the fracture segmentation using the two-stage method and the PointRend module both statistically and visually. We used IoU to compare the ground truth and predicted segmentation. The IoU between two objects is defined as the ratio of the overlap area between the objects to the total area of both objects. IoU is defined in Eq. 2 where $objectA$ and $objectB$ can be ground truth and predicted masks, respectively. IoU ranges from 0 (no match) to 1 (a perfect match); for the fractures that appear in the core images as detailed thin objects, IoUs around 0.5 can indicate good predictions, as shown in Section 3.

$$IoU = \frac{objectA \cap objectB}{objectA \cup objectB}. \quad (2)$$

We evaluated the effect of mask improvement on the accuracy of fracture aperture calculation, for a group of eight sample fractures (discussed in Section 3.2). Fracture aperture was calculated from the binary masks segmented manually and by the model, following the procedure in Appendix A. To evaluate aperture calculations by the model, we computed the absolute error as the difference between aperture measurements from ground truth and predicted masks.

We also assessed the quality of the segmentation through visual inspection, including comparing the one-stage and two-stage segmentations as well as comparing standard Mask R-CNN and Mask R-CNN + PointRend segmentations.

3 Results and discussion

The training and validation statistics of segmentation in Figure 4 and 5 demonstrate the following.

(1) Mask R-CNN + PointRend had lower training loss and significantly higher validation precision than standard Mask R-CNN in both model-A and model-B, suggesting that the PointRend module can enhance fracture segmentation by Mask R-CNN.

(2) Model-B had more accurate segmentation than model-A as the segmentation precision increased from 4% for the full-size images (model-A) to 16% for the small images (model-B) in standard Mask R-CNN, and from 25% to 48% in Mask R-CNN + PointRend.

The results after integrating model-A and model-B in the two-stage method from the 72 test images (Table 2) are discussed in the next sections.

3.1 Results from IoU calculation

Based on the calculation of IoU between predicted and ground truth fractures in the test images, the two-stage method showed improvement in the segmentation from the one-stage method by both the standard Mask R-CNN and Mask R-CNN + PointRend models. The IoUs increased on average by 0.14 and up to 0.34 using standard Mask R-CNN, and on average by 0.11 and up to 0.27 using Mask R-CNN + PointRend, as shown in Table 3 and Figure 6. The segmentation of the test images from block 7220, which was used in the models training, was better than that of the images from block 16; however, the latter showed slightly more improvement by the two-stage method (Table 3). As demonstrated in Figure 6, the distributions of the IoU obtained by the two-stage method shifted towards the right side of the X-axis, i.e., higher IoUs.

Table 3. Comparison between the average IoU of 160 fractures in the test images from blocks 7220 and 16 (see Table 2), using the one-stage and the two-stage method.

Architecture	Block	One-stage method	Two-stage method
Standard Mask R-CNN	7220	0.34	0.47
	16	0.18	0.33
Mask R-CNN + PointRend	7220	0.47	0.56
	16	0.34	0.45

The method provided a significant increase in the percentage of fractures segmented with IoU greater than 0.5. The percentage increased from 7% in the one-stage method to 31% in the two-stage method using standard Mask R-CNN (Figure 6, top). Similarly with using Mask R-CNN + PointRend, the percentage of fractures with IoU > 0.5 almost doubled by the two-stage method, increasing from 33% to 62% (Figure 6, bottom).

Standard Mask R-CNN gained more improvement from the second stage of segmentation than Mask R-CNN + PointRend as the standard architecture provided poor segmentation of the full-size images in the first stage, due to its limited mask resolution, compared with Mask R-CNN + PointRend.

Overall, the best segmentation, in terms of the IoU, was obtained by the two-stage segmentation using Mask R-CNN + PointRend. It resulted in a greater than 0.5 average IoU for 160 fractures with 62% of the IoUs greater than this average. This represented a 0.25 increase in the IoU from the baseline of the one-stage segmentation using standard Mask R-CNN.

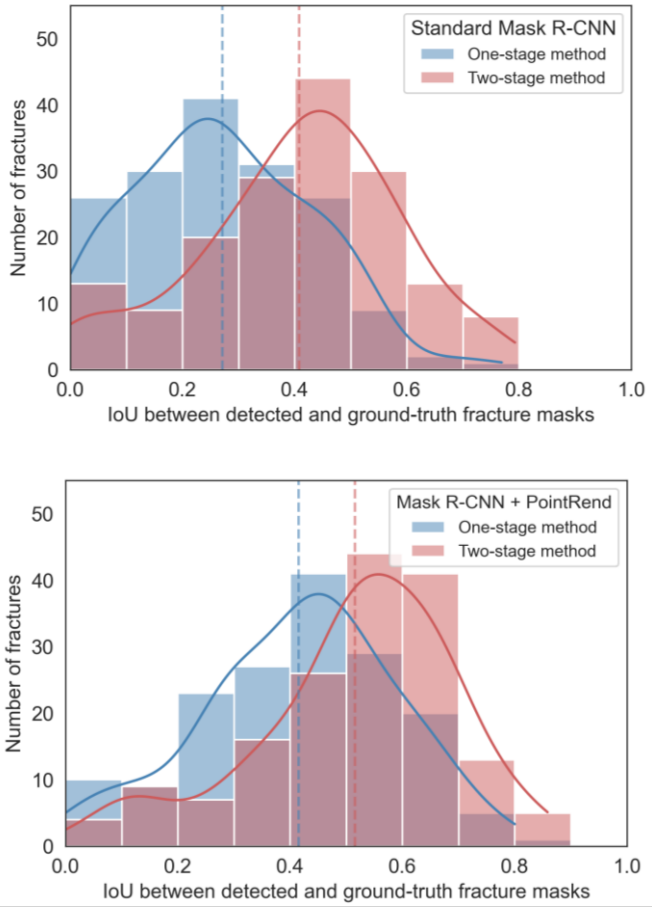


Fig. 6. Improvement in IoUs gained by the two-stage method for 160 fractures in the test images. Dashed lines show average values for each distribution. The distribution after the second stage of segmentation (in red) shifted to the right and the average IoU increased from 0.27 to 0.41 using standard Mask R-CNN (top) and from 0.41 to 0.52 using Mask R-CNN + PointRend in (bottom).

3.2 Improvement in aperture calculation

We evaluated the potential increase in the accuracy of the fracture aperture calculation that resulted from the enhanced segmentation by the two-stage method using both tested architectures. We focused on a group of eight fractures as shown in Table 4.

The third and fourth columns in Table 4 list absolute errors of aperture calculation based on the one-stage and two-stage segmentation, respectively. The errors were the average values of roughly 60 aperture measurements for each fracture (more details in Appendix A). The absolute error decreased by 0.46 mm (i.e., 46%) on average for the eight samples by using the two-stage method. Note that the image resolution of the samples was 2.6–9 pixels/mm. The absolute error decreased by 0.52 mm (i.e., 41%) and 0.40 mm (i.e., 50%) for the samples of standard Mask R-CNN and Mask R-CNN + PointRend, respectively. The results also showed that IoUs > 0.5 yielded an average aperture error of 0.61 mm compared to 0.93 mm from IoUs < 0.5.

Sample 3 showed a significant reduction in the error of 0.74 mm (i.e., 56%) following an increase in the IoU from 0.51 to 0.74. A detail demonstration for 63 aperture

measurements of Sample 3 is provided in Figure 7, indicating the improvement in each measurement using the two-stage standard Mask R-CNN. The mismatch between the ground-truth and estimated apertures have noticeably decreased in the improved segmentation (Figure 7, bottom) compared to the mismatch in the initial segmentation (Figure 7, top).

Thus, it can be concluded that enhancing fracture segmentation using the proposed two-stage segmentation approach can result in a substantial increase in the accuracy of aperture calculation.

Table 4. Absolute error for fracture aperture calculation for eight samples, showing the reduction in error after the second stage of segmentation (column 4) compared to initial results in column 3.

		Absolute error (mm)	
	Sample	One-stage method	Two-stage method
Standard Mask R-CNN	1	0.68	0.46
	2	1.81	0.93
	3	1.31	0.57
	4	0.86	0.64
Mask R-CNN with PointRend	5	0.68	0.26
	6	0.8	0.4
	7	0.87	0.4
	8	0.81	0.52

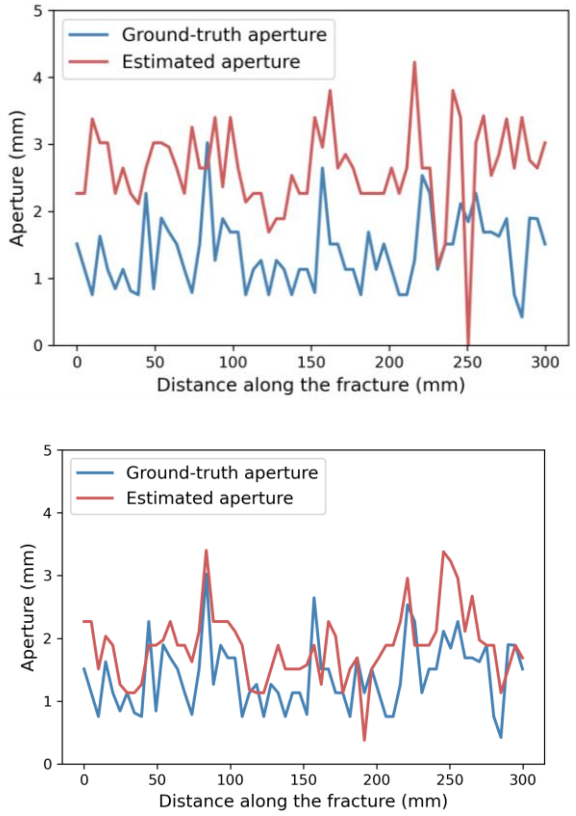


Fig. 7. Improvement in aperture accuracy for Sample 3 in Table 4 from calculation based on initial mask (top) to enhanced mask after a second segmentation (bottom), using standard Mask R-CNN.

3.3 Visual inspection results

Visual inspection of the predicted fracture masks proved that the two-stage method provided more accurate fracture segmentation than the one-stage method using both Mask R-CNN architectures. The proposed method showed visual improvement in the segmentation accuracy of fractures in the test images. The fracture masks can be observed in Figure 8. For standard Mask R-CNN (Figure 8a), enhancement in the fracture segmentation can be noticed in the mask obtained from the second stage of segmentation with a 0.38 IoU compared to the initial mask with a 0.15 IoU. Similarly, for Mask R-CNN + PointRend (Figure 8b), the two-stage method yielded a more detailed segmentation for the fracture than that provided by the one-stage method, as indicated by the IoU increasing from 0.40 to 0.68

From the visual comparison of the results from both architectures, we observed that Mask R-CNN + PointRend segmented the fractures more accurately than the standard Mask R-CNN, in particular for fine fractures, such as the fracture shown in Figure 9. This observation agreed with the IoU-based evaluation, which indicated that Mask R-CNN + PointRend offered more accurate fracture segmentation, i.e., higher IoUs than the standard Mask R-CNN. We attribute this to the mask refinement process by the PointRend module as introduced in Section 2.2.2 and Figure A1 of Appendix A.

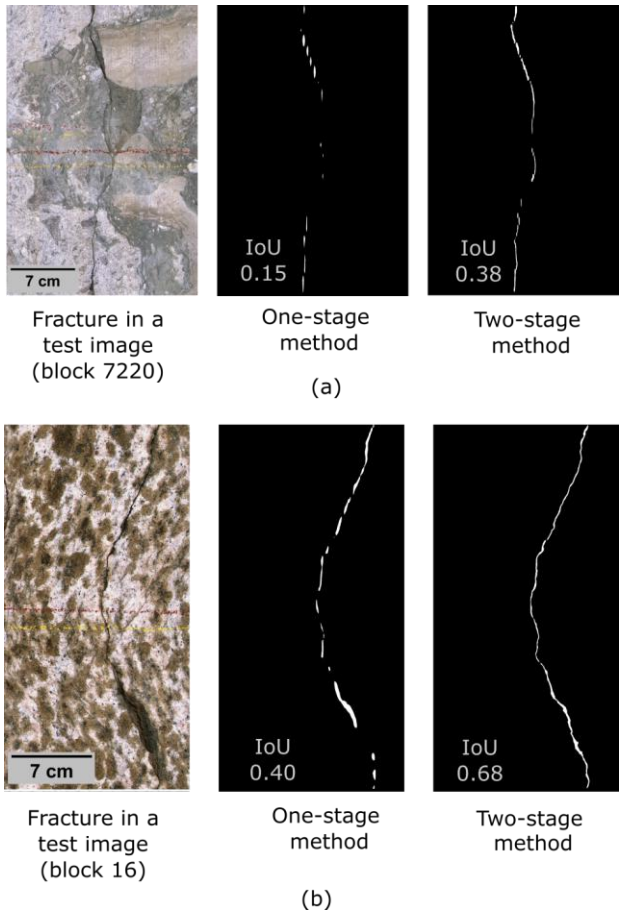


Fig. 8. Visualization of fracture masks enhancement by the two-stage segmentation using (a) standard Mask R-CNN and (b) Mask R-CNN + PointRend.

Although Mask R-CNN + PointRend showed noticeably better segmentation when evaluated both quantitively and qualitatively than the standard architecture, it introduced an issue when segmenting adjacent or overlapped fractures. In some cases, Mask R-CNN + PointRend failed to distinguish between the main fracture and other fractures present within the detected region. A prime example is shown in Figure 10. We found that this problem, on the other hand, appeared much less often in the results from standard Mask R-CNN.

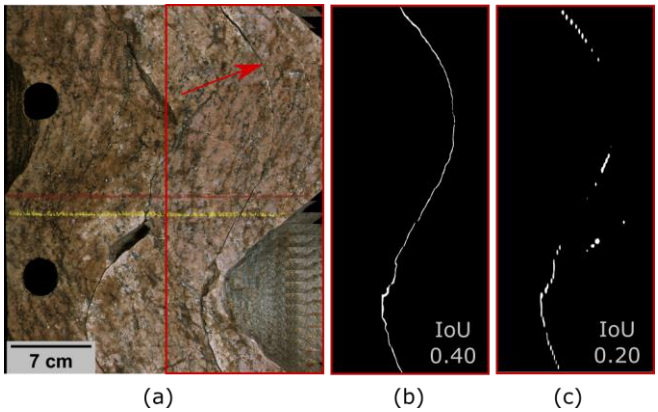


Fig. 9. Fine fracture in test image from block 16 (a) segmented more accurately by Mask R-CNN + PointRend (b) than standard Mask R-CNN (c).

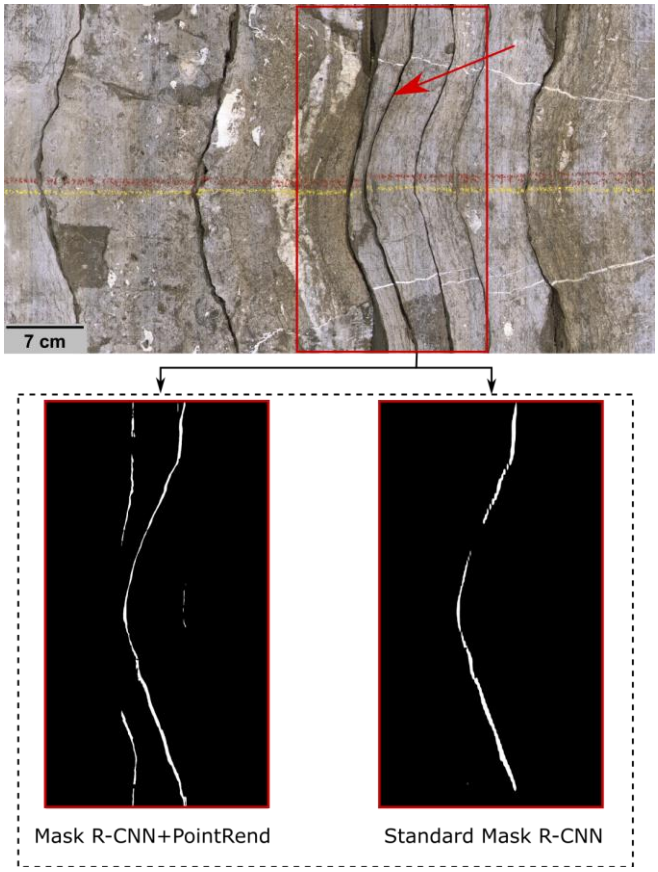


Fig. 10. Visualization of Mask R-CNN + PointRend problem in segmenting adjacent fractures compared to correct segmentation by standard Mask R-CNN.

Overall, the visual evaluation of the results confirmed that the second stage of segmentation improved the results significantly for both architectures. It also demonstrated that Mask R-CNN + PointRend did not always provide better fracture segmentation than standard Mask R-CNN as it was sensitive to the complexity level of the fractures and the presence of adjacent or overlapped fractures.

4 Conclusion

We proposed a two-stage segmentation method to improve fracture segmentation from unwrapped core images and enable detailed fracture characterization. The method relied on using two instance segmentation models that segmented the image at different scales. We tested two architectures of Mask R-CNN with and without the PointRend module.

Fracture segmentation results from the test images were assessed quantitively and qualitatively. Quantitative analysis showed that the two-stage segmentation can increase the IoU by up to 0.34 and 0.27 using standard Mask R-CNN and Mask R-CNN + PointRend, respectively. Enhancing fracture segmentation can potentially obtain a 46% increase in the accuracy of aperture calculation. The quantitative analysis was confirmed by visual evaluation of the results that demonstrated an improvement in the fracture segmentation, which was noticeable for both tested architectures.

The comparison between standard Mask R-CNN and Mask R-CNN + PointRend demonstrated that the latter provided more accurate and detailed segmentation than the former, but it was affected by overlapping and neighboring fractures.

Instance segmentation is a powerful tool for the identification and characterization of image features where its utility has been readily realized for various applications. For core analysis, instance segmentation provides a platform for the identification of core features. While the current study is limited to fracture identification, future works can build on this platform to provide a spate of tools for the automated analysis of core images. Overall, we envisage a "Digital Geologist" machine learning platform based on deep learning and convolutional neural networks.

Acknowledgment

The authors thank Lundin Energy for providing the unwrapped core images.

List of abbreviations

Mask R-CNN	Mask region-based convolutional neural network
PointRend	Point-based rendering
JRC	Joint roughness coefficient
ML	Machine Learning
Mask R-CNN + PointRend	Mask R-CNN with the PointRend module

IoU	Intersection-over-union
RPN	Region proposal network
FPN	Feature pyramid network
RoI	Region of interest

References

1. T. S. Paulsen, R. D. Jarrard, and T. J. Wilson, "A simple method for orienting drill core by correlating features in whole-core scans and oriented borehole-wall imagery" J. Struct. Geol., **24**, 8, 1233–1238 (2002), doi: 10.1016/S0191-8141(01)00133-X.
2. S. Tiwari *et al.*, "High resolution core scan facility at BGRL-MoES, Karad, India*"* J. Geol. Soc. India, **90**, 6, 795–797 (2017), doi: 10.1007/s12594-017-0793-6.
3. R. Schepers, G. Rafat, C. Gelbke, and B. Lehmann, "Application of borehole logging, core imaging and tomography to geotechnical exploration" Int. J. Rock Mech. Min. Sci., **38**, 6, 867–876 (2001), doi: 10.1016/S1365-1609(01)00052-1.
4. M. M. Al-Fahmi, S. I. Ozkaya, and J. A. Cartwright, "FracRough—Computer program to calculate fracture roughness from reservoir rock core" Appl. Comput. Geosci., **9**, p. 100045 (2021), doi: 10.1016/j.acags.2020.100045.
5. M. Chakraborty and S. Mukherjee, "Structural geological interpretations from unrolled images of drill cores" Mar. Pet. Geol., **115**, p. 104241 (2020), doi: 10.1016/j.marpetgeo.2020.104241.
6. M. M. Al-Fahmi, S. I. Ozkaya, and J. A. Cartwright, "FracRough—Computer program to calculate fracture roughness from reservoir rock core" Appl. Comput. Geosci., **9**, p. 100045 (2021), doi: 10.1016/j.acags.2020.100045.
7. N. Barton and V. Choubey, "The Shear Strength of Rock Joints in Theory and Practice" Rock Mech., **10**, 1–54 (1977).
8. F. Lemy, J. Hadjigeorgiou, P. Côté, and X. Maldague, "Image analysis of drill core" Inst. Min. Metall. Trans. Sect. A Min. Technol., **110** (2001), doi: 10.1179/mnt.2001.110.3.172.
9. H. Ozturk and I. T. Saricam, "Core Segmentation and Fracture Path Detection Using Shadows" J. Image Graph., **6**, 1, 69–73 (2018), doi: 10.18178/joig.6.1.69-73.
10. A. Xavier, C. E. Guerra, and A. Andrade, "Fracture analysis in borehole acoustic images using mathematical morphology" J. Geophys. Eng., **12**, 3, 492–501 (2015), doi: 10.1088/1742-2132/12/3/492.
11. D. Wedge, E. J. Holden, M. Dentith, and N. Spadaccini, "Fast and objective detection and analysis of structures in downhole images" J. Appl. Geophys., **144**, 157–172 (2017), doi: 10.1016/j.jappgeo.2017.07.004.
12. R. A. Q. Cruz, D. C. Cacau, R. M. dos Santos, E. J. Ribeiro Pereira, F. R. Leta, and E. Gonzalez Clua, "Improving Accuracy of Automatic Fracture Detection in Borehole Images with Deep Learning and GPUs" in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 345–350 (2017), doi: 10.1109/SIBGRAPI.2017.52.
13. L. O. Dias *et al.*, "Automatic detection of fractures and breakouts patterns in acoustic borehole image logs using fast-region convolutional neural networks" J. Pet. Sci. Eng., **191**, p. 107099 (2020), doi: 10.1016/j.petrol.2020.107099.
14. K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN" in *2017 IEEE International Conference on Computer Vision*, 2961–2969 (2017), doi: 10.1109/ICCV.2017.322.
15. "COCO - Common Objects in Context." [Online]. Available: https://cocodataset.org/. [Accessed: 1-May-2021].
16. Y. Wang, L. Zhao, M. Wang, and Z. Song, "Organ at Risk Segmentation in Head and Neck CT Images Using a Two-Stage Segmentation Framework Based on 3D U-Net" *IEEE Access*, **7**, 144591–144602 (2019), doi: 10.1109/ACCESS.2019.2944958.
17. M. Amiri, R. Brooks, B. Behboodi, and H. Rivaz, "Two-stage ultrasound image segmentation using U-Net and test time augmentation" Int. J. Comput. Assist. Radiol. Surg., **15**, 6, 981–988 (2020), doi: 10.1007/s11548-020-02158-3.
18. N. J. Tustison and G. C. James, "Introducing Dice, Jaccard, and Other Label Overlap Measures To ITK" (2009).
19. A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering" in *Computer Vision and Pattern Recognition (CVPR)*, 9799–9808 (2020).
20. "Lundin Energy." [Online]. Available: https://www.lundin-energy.com/. [Accessed: 14-Oct-2020].
21. NPD, "FactPages from the Norwegian Petroleum Directorate." [Online]. Available: https://factpages.npd.no/. [Accessed: 26-Apr-2021].
22. "Supervisely." [Online]. Available: https://supervise.ly/. [Accessed: 20-Oct-2020].
23. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2117–2125 (2017).
24. Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2" (2019).
25. T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context" in *European Conference on Computer Vision*, **8693 LNCS**, PART 5, 740–755 (2014), doi: 10.1007/978-3-319-10602-1_48.

Appendix A

1. PointRend module

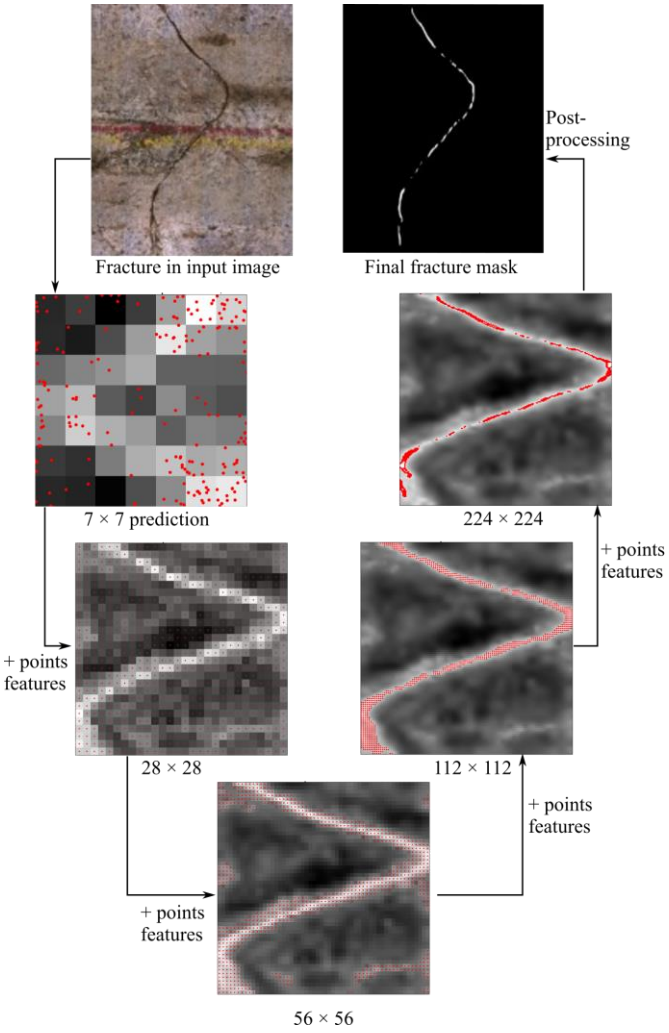


Fig. A1. Mask refinement in PointRend module–incorporated in Mask R-CNN's segmentation branch– through multiple iterations. Red dots represent the selected uncertain points. At each iteration, a low-resolution mask and points features from the CNN were used to predict a higher resolution mask. The visualization of the intermediate masks was based on [24].

2. Fracture aperture calculation

Fracture aperture was calculated from the binary masks produced by the models or by manual segmentation.

For each fracture, the aperture was measured at over 60 locations by moving a small window along the fracture with a vertical step of 5 mm as shown in Figure A2, then we implemented the following steps at each location:

a) Extracting fracture points from fracture skeleton at that location.
b) Finding the line of best fit to the extracted points.
c) Finding a line perpendicular to the best-fit line.
d) Calculate fracture aperture in pixels, represented by the white pixels along the perpendicular line.
e) Converting the measurement from pixels to mm using the scale of the unwrapped core image.

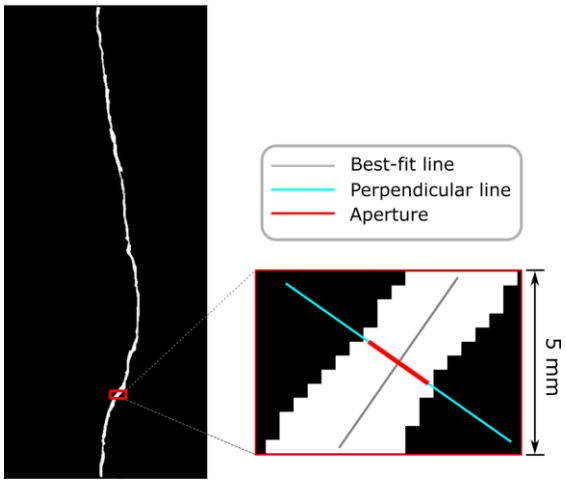


Fig. A2. Demonstration of fracture aperture calculation from a binary mask, showing the process at the highlighted location.