

## Automatic fracture detection and characterization from unwrapped drill-core images using mask R-CNN



Fatimah Alzubaidi<sup>a</sup>, Patrick Makuluni<sup>a</sup>, Stuart R. Clark<sup>a</sup>, Jan Erik Lie<sup>b</sup>, Peyman Mostaghimi<sup>a</sup>, Ryan T. Armstrong<sup>a,\*</sup>

<sup>a</sup> School of Minerals and Energy Resources Engineering, The University of New South Wales, Sydney, Australia

<sup>b</sup> Lundin Energy Norway AS, Lysaker, Norway

### ARTICLE INFO

**Keywords:**  
Unwrapped core images  
Mask R-CNN  
Fracture detection  
Fracture analysis

### ABSTRACT

Drill cores provide reliable information about fractures in subsurface formations as they present a clear and direct view of fractures. Core observation and image log interpretation are usually integrated for fracture analysis of underground layers. There has been a strong move towards developing automated fracture detection methods, however, the focus has been on extracting fracture information from log images, such as acoustic or resistivity image logs. Such efforts using core images are significantly less. This paper presents a machine learning-based approach for automatic fracture recognition from unwrapped drill-core images. The proposed method applies a state-of-the-art convolutional neural network for object identification and segmentation. The study also investigates the feasibility of using synthetic fracture images for the training of a learning model. Synthetic data can provide an alternative to real data, and thus address data availability issues common for supervised machine learning applications. We first create two types of synthetic data by using masks of real fractures and creating sinusoidal shaped fractures. The trained model is evaluated on real core images from two boreholes and provides an average precision of approximately 95%. The identified fractures are further analyzed and compared to manually segmented fractures in terms of fracture dip angle and dip direction, which achieved average absolute errors of around 2° and 11°, respectively. Overall, the study presents a novel application of an advanced machine learning algorithm for fracture detection and analysis from unwrapped core images.

### 1. Introduction

Fracture analysis for subsurface formations is crucial in many geological, geotechnical, and petroleum related applications. It is important for geological modeling, oil and gas reservoir characterization, borehole stability, and assessing rock quality for subsurface engineering. Subsurface fractures are analyzed using well log data and drill cores; fractures are detected and evaluated to obtain fracture parameters, such as width, orientation, and nature (Fernández-Ibáñez et al., 2018; Lai et al., 2021; Nian et al., 2016; Van Golf-Racht, 1982). Fracture attributes (e.g., orientation) can also provide insights into the generation and evolution of fractures, through fracture clustering and the determination of fracture sets (Massiot et al., 2017; Shanley and Mahtab, 1976). These analyses assist in understanding the geology of oil and gas reservoirs, which is essential for reservoir modeling and reservoir characterization (Van Golf-Racht, 1982). Fracture properties and origin are used to evaluate the effect of fractures on the flow and storage

properties of reservoirs; fractures can therefore have a major contribution to the permeability and porosity of reservoirs (Agosta et al., 2007; Nelson, 2001; Zeng and Li, 2009). In the conventional fracture analysis procedure, experts manually identify fractures present in core and image logs by visual inspection. This process is usually laborious and time-consuming; therefore, there is a need for fast and reliable approaches to automate fracture detection and analysis.

Most previous works focus on automating fracture analysis using image logs as they cover a significant length of the wellbore; however, drill cores provide a more reliable, clear, and direct view of underground fractures. The importance of core observation for the detection of natural fractures was demonstrated by Fernández-Ibáñez et al. (2018). In the studied boreholes, they concluded that integrating core and image logs eliminated more than half of the fractures identified only from image logs and enabled observing thin fractures below the resolution of the logging tool. Another advantage of fracture analysis from drill cores, according to Laubach and Monson (1988), is the identification of

\* Corresponding author.

E-mail address: [ryan.armstrong@unsw.edu.au](mailto:ryan.armstrong@unsw.edu.au) (R.T. Armstrong).

coring-induced fractures, which provide information regarding reservoir behavior during hydraulic fracturing by comparing induced fractures and natural fractures. While fracture analysis from drill core is essential, attempts towards automating it are relatively limited compared to image log-based studies.

Recently, core storage has moved towards digital archiving of core tray images or unwrapped core images, owing to the availability of advanced core scanning tools (Betlem et al., 2020; Schodlok et al., 2016; Tiwari et al., 2017). To extract relevant data from such images, rapid and reliable means are needed to process and analyze the produced images. For fracture analysis, the unwrapped core images are superior to core tray images because they show detailed fracture features of the entire core. These images are taken by rotating the core sample for 360° and capturing the core at each angle, with up to 40 pixels/mm (i.e. 25 µm) resolution (Tiwari et al., 2017). Studies based on analyzing unwrapped core images are limited although they provide valuable fracture information.

An early attempt for automating fracture recognition from drill-core images was conducted by Hall et al. (1996). The study applied an edge detection algorithm and the Hough transform to locate geological features from both unwrapped core images and borehole binary images. Almost a similar approach was used by Lemy et al. (2001) to detect edges in core tray images. Instead of binary images, they used grayscale images after enhancing the contrast between bright regions (core) and dark regions (fractures) and applied Steger's line detection algorithm for fracture detection. Ozturk and Saricam (2018) detected fractures from core tray images by enhancing shadows between cores during image capturing then extracting the edges using a Canny edge detector. These studies were based on image processing algorithms that have limited generalization and robustness when applied to new data because of being sensitive to image quality, image noises, and the presence of veins and bedding.

Image processing techniques have also been applied to identify fractures in image logs (Elkington and Assous, 2017; Wedge et al., 2017; Xavier et al., 2015; Ye, 2016). Xavier et al. (2015) used mathematical morphology to detect edges from borehole acoustic images and then applied fourth-degree polynomials to fit the detected edges and represent the fractures. Their method was evaluated on a very limited number of real acoustic images. A more recent study that used image processing was presented by Wedge et al. (2017) to detect fractures from acoustic, resistivity, and optical downhole images. They introduced a pre-processing step prior to the automatic detection to exclude zones with complex structures by evaluating the input image complexity. The main steps of their workflow, similar to other image processing-based methods, were edge detection followed by sinusoidal detection and fitting. The sinusoidal detection was applied on pairs of symmetrical detected segments. The method required a user-defined threshold of fracture confidence to optimize the detections.

Recently, machine learning (ML) algorithms have been adopted to improve the automatic detection of fractures from image logs. Cruz et al. (2017) used a convolutional neural network (CNN), AlexNet (Krizhevsky et al., 2012), to detect fractures from electrical and acoustic image logs. However, they first used the Hough transform to mark fracture candidates, and then used the CNN to confirm or eliminate the detections (Cruz et al., 2017). In a later study, Dias et al. (2020) applied a fast region-based CNN (FRCNN) (Girshick, 2015) to detect fractures and breakouts from acoustic images directly. The FRCNN was trained with synthetic acoustic images by fine-tuning a pre-trained model. Their results were evaluated by measuring the Area Under the Receiver Operating Characteristic Curve (AUC); a value near 100% indicates an excellent classification (Fan et al., 2006). The method achieved 98% AUC for fractures and 90% AUC for breakouts based on synthetic images (Dias et al., 2020). Their approach was also tested on a small section of real images with nine target fractures and one breakout; all fractures were detected with one false detection and the breakout was detected through eight separated boxes with a false detection. The main

advantages of this CNN-based method were that it eliminated a common issue of training data limitation and provided better detection than other previous approaches. However, the objects were detected by bounding boxes that cannot be used directly for further characterization and analysis.

Using ML for fracture analysis from unwrapped core images has not yet been attempted, according to our knowledge. Therefore, advanced approaches based on ML algorithms, such as CNNs, need to be developed and evaluated to analyze fractures in core images. The main advantage of using ML algorithms in image analysis is their ability to learn representative features that makes them less sensitive to image noises than traditional image processing algorithms; these features are learned usually from many training images to create generalized models. Unlike traditional image processing methods, such as edge detection algorithms that rely mainly on identifying sudden changes in pixel intensity of grayscale images (Maini and Aggarwal, 2009; Shrivakshan and Chandrasekar, 2012), ML models can be trained to recognize fractures among other structural features, such as veins, layering, and bedding using color and other learned representative features. A particular type of ML algorithms called Mask R-CNN has been used in a wide array of industries for object detection and image segmentation. The applications include medical-related segmentation (Chiao et al., 2019; Liu et al., 2018) and segmentation from remote sensing images (Maxwell et al., 2020; Nie et al., 2020). Mask R-CNN, which is an object instance segmentation model, performs object localization, classification, and segmentation. It is an extension to previous object detection networks, i.e., Faster R-CNN (Ren et al., 2015) and Fast R-CNN (Girshick, 2015).

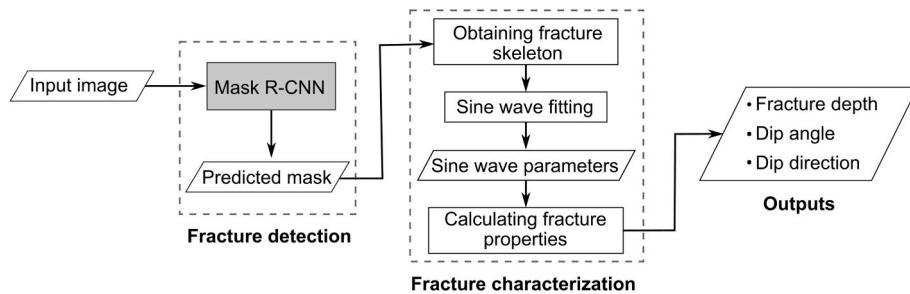
In this work, we demonstrate an innovative approach for fracture characterization whereby a Mask R-CNN model (He et al., 2017) was developed for fracture detection from unwrapped core images. The model was trained to detect macro fractures, i.e., fractures with width >0.1 mm (Van Golf-Racht, 1982). We used the model to detect and segment fractures in core images; we then fit a sine wave to the detected fracture points to obtain: depth, dip angle, and dip direction, as demonstrated in Fig. 1. In addition, the model was trained with synthetic images and evaluated using real images. We tested the model on 88 m of core from two boreholes in a total processing time of 5 minutes. Our calculations assumed the case of a vertical core where the objective was to compare the model results to those from manual detection whereas providing a real geological interpretation of the region was beyond our scope.

## 2. Data and methods

A description of the core images used and development of the synthetic images are provided in Section 2.1, Mask R-CNN application, training and evaluation are explained in Section 2.2, and the fracture characteristic calculations are described in Section 2.3.

### 2.1. Data acquisition and preparation

The study used unwrapped 360° drill-core images obtained from Lundin Energy; a Norwegian-based company with a long history of exploration and production of oil and gas. The core images were extracted from 1149 to 1203 m of well no. 7220-6-1 with a resolution of 2.6 pixel/mm, and from 1852 to 1945 m depth of well no. 7220-11-3 with a resolution of 9 pixel/mm. These wells are located on the Loppa High, which is a structural high, shallow-water carbonate shelf of the Norwegian Barents Sea (Sayago et al., 2018). The unwrapped images were collected from the Gipsdalen Group, which mainly comprise mid-Carboniferous to Early Permian carbonate units (Sayago, 2014). The interval between 1149 and 1203 m in well no. 7220-6-1 is specifically composed of fine and coarse dolomites, dolomitic packbreccias, green shales, and siltstones. Similarly, the interval between 1852 and 1945 m in well no. 7220-11-3 is comprised of dolomites and limestones with minor siliciclastic lithologies (Hansen, 2019). The Loppa High was



**Fig. 1.** The workflow of the proposed method. Inputs are unwrapped core images and outputs are depth, dip angle, and dip direction of detected fractures.

created by the Paleozoic Extension, and the region has been continuously rotated and uplifted, thereby producing various structural features, such as fractures, as observed in the unwrapped images (Sayago et al., 2018). In the remainder of the study, boreholes 7220-6-1 and 7220-11-3 will be referred to as BH1 and BH2, respectively.

The availability of labeled data is a common challenge in supervised machine learning. A large majority of the data (usually 60–80%) are used for training, leaving less than 40% available for testing. In this study, however, the objective was to train the model with synthetic data and evaluate it with real data. We created two types of synthetic fractures using fracture shapes from real fractured-core images and a sine function. The type and number of images selected for the training, validation, and test datasets are summarized in Table 1. The synthetic data were only used for training and validation, not for testing. The next sub-sections describe the three types of images listed in Table 1.

### 2.1.1. Images with real fractures

Most of the images with real fractures were used for testing the proposed approach. The test images included approximately 50 m of core (with 108 fractures) from BH1 and 38 m (with 135 fractures) from BH2. The training dataset included only 15 images with real fractures. We applied image augmentation (cropping, horizontal flipping, and vertical flipping) to increase the number of images to 62 with a total of 112 fractures. The validation dataset included 35 real images including 61 fractures. The validation dataset was used during the development of the Mask R-CNN model and the optimization of the training and architecture related parameters while the test dataset was used for the final evaluation of fracture detection and characterization. Both training and validation images were taken from BH2, reserving all images from BH1 for testing.

The manual labelling of fractures was performed using the Supervisely tool (Supervisely); an online platform for image labelling and data management.

### 2.1.2. Images with synthetic fractures A

We added the masks of real fractures from the training dataset to background images to replicate fractured-core images. Starting with 26 background images from BH2, different images were created by applying random-based processing steps on both the background image

and the fracture mask before generating a fractured-core image. The process is shown in Fig. 2. The background images were randomly flipped (vertically or horizontally) and cropped to a random ratio of 60–90% of their original size. The fracture mask pre-processing included: (1) decreasing or increasing fracture apertures using image morphology, (2) vertical or horizontal flipping, (3) stretching or compressing the masks in the dimension perpendicular to the fracture direction to create fractures with different dip angles from a single mask. Before merging the fracture mask and the background image, a gradient was created between the fracture (white pixels) and the black background to smooth the transition in intensity. The last processing step included blurring and reducing the intensity of the entire mask to 30–50% of the original value to mimic a real fracture that is not purely black. Finally, the merging process was performed by subtracting the fracture area in the mask from the background image at a randomly selected location.

A total of 353 images were created. Also, an additional 236 images were generated by adding a second fracture to create more complex structures with nearby or overlapped fractures. The resulting images had different sizes of ~1530–2860 pixels width and ~940–2820 pixels height. We used 90% of the generated images for training and 10% for validation, as explained in Table 1.

### 2.1.3. Images with synthetic fractures B

Another type of synthetic image was created by generating the fracture shape from a sine function. A similar approach was applied by Dias et al. (2020) on log images. The same background images from Section 2.1.2 were used with applying additional augmentation including randomly altering the contrast and brightness, flipping, and cropping. The images were downsampled to have 600 pixels on the large edge. The purpose of the augmentation and downsampling was to create different images from the previous synthetic images and, in turn, diverse training images with a range of size, contrast, and brightness.

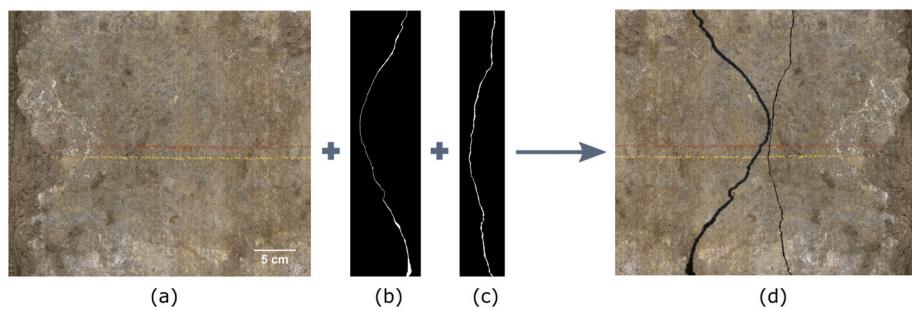
Fracture shapes were created using the following equation from (Elkington and Assous, 2017)

$$f(x) = loc + a \sin\left(\frac{2\pi}{w}(x - \alpha)\right), \quad (1)$$

**Table 1**

Shows the number and type of fractures in the training, validation, and test datasets. Synthetic fractures A and B were created according to 2.1.2 and 2.1.3, respectively.

|                       | Training dataset |                     | Validation dataset |                     | Test dataset |                     |
|-----------------------|------------------|---------------------|--------------------|---------------------|--------------|---------------------|
|                       | Number of images | Number of fractures | Number of images   | Number of fractures | Core length  | Number of fractures |
| Real fractures        | 62               | 112                 | 35                 | 61                  | 88 m         | 243                 |
| Synthetic fractures A | 531              | 744                 | 58                 | 81                  |              |                     |
| Synthetic fractures B | 525              | 793                 | 52                 | 76                  |              |                     |
| Total                 | 1118             | 1649                | 145                | 218                 |              |                     |



**Fig. 2.** An illustration of the process of creating core images with synthetic fractures A, showing a background image in (a) combined with two fracture masks (b and c) to create a fractured core image (d).

where  $loc$ ,  $a$  and  $\alpha$  are randomly defined location, amplitude, and phase angle of the sine wave, respectively;  $w$  is the width of the background image. After creating the shape as a 1-pixel line, fracture aperture was added by dividing the fracture into 1–4 segments and adding a different aperture to each segment, instead of having a uniform aperture along the fracture. The same steps in Section 2.1.2 were followed to merge the sinusoidal fractures and background images.

A total of 285 images were created with a single fracture, and 292 with two fractures. Fig. 3 presents sample images of different contrast and brightness with sinusoidal fractures of different dip angles, dip directions, and apertures.

## 2.2. Mask R-CNN

Mask R-CNN is an instance segmentation model developed by He et al. (2017) as a state-of-art framework that outperformed other existing models in terms of accuracy and simplicity. As an instance segmentation model, Mask R-CNN localizes, classifies, and segments objects, combining both object detection (Papageorgiou et al., 1998; Redmon et al., 2016; Ren et al., 2015) and semantic segmentation (Garcia-Garcia et al., 2017; Long et al., 2015). The default Mask R-CNN architecture was built based on the Faster R-CNN architecture for object detection (Ren et al., 2015) by adding a new branch for the mask prediction to the existing branch of classification and bounding box regression (Fig. 4). The model uses a deep CNN for extracting features from the input images. The extracted features are the main input for the subsequent operations. Mask R-CNN is a region-based algorithm, i.e., it uses a region proposal network (RPN) to predict regions that potentially include objects. RPN relies on the extracted features, instead of the input image, to propose candidate object regions of different sizes and aspect ratios according to predefined anchors. Then, a region of interest aligning layer (RoIAlign) uses both the proposed boxes by RPN and the feature maps from the backbone to crop small feature maps and resize them to a uniform shape as inputs to the final prediction branches. RoIAlign is a replacement to the ROI pooling layer in the original Faster

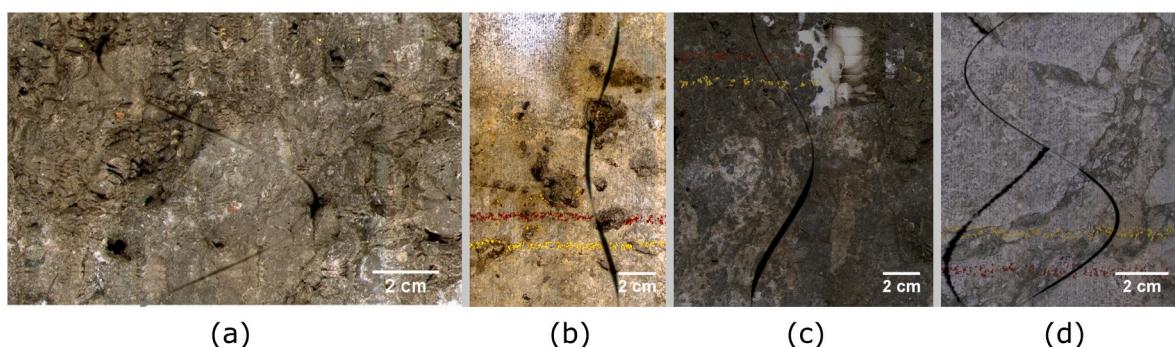
R-CNN that allows a better pixel-wise segmentation. Each output feature map from RoIAlign has a shape of  $7 \times 7$  pixels and  $14 \times 14$  pixels for the detection and segmentation branches, respectively; these shapes are hyper-parameters, as discussed in (Girshick, 2015). Finally, the cropped RoIs pass through each branch in parallel to receive a classification score, bounding box coordinates, and a binary segmentation mask.

The architecture adopted in our study is shown in Fig. 4. The feature extraction backbone was a 50-layer deep ResNet network (He et al., 2016) with the adaptation of a feature pyramid network (FPN) proposed by Lin et al. (2017). FPN builds a feature pyramid inside the network via a top-down architecture with lateral connections. In ResNet-FPN, a feature level is created from each stage/block; it has four levels obtained from the last four blocks excluding the first convolutional layer (Lin et al., 2017). With FPN, RoIs features are extracted from different levels of the pyramid rather than using features from the last layer as in ordinary networks. According to He et al. (2017), Mask R-CNN with ResNet-FPN resulted in a better and faster prediction.

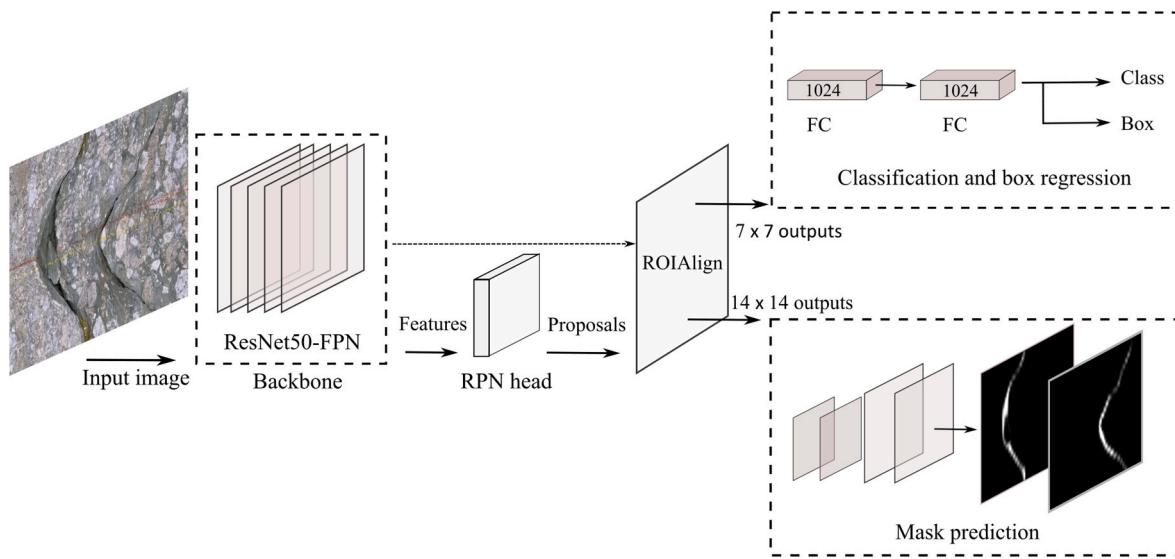
For RPN anchors, we used larger sizes than the default anchors in the original Mask R-CNN and different height-to-width ratios to match the sizes and aspect ratios of the fractures in our dataset (images of horizontal cores). We defined anchors with sizes of  $256^2$ ,  $400^2$ ,  $512^2$ ,  $700^2$ , and  $1024^2$  pixels and height-to-width ratios of 3:4, 1:1, and 2:1. We had two classes: fracture and background. The remainder of the architecture was according to the default Mask R-CNN with FPN backbone.

### 2.2.1. Training implementation

Input images were resized to have a minimum of 600 pixels on the short side and a maximum of 2000 pixels on the large side. Using a larger input size was more computationally expensive and not feasible in terms of time and memory requirement. A batch of two images was passed through the model during each iteration using a GPU (see section 2.2.2). We started with a pre-trained model on the MSCOCO dataset of natural images (Lin et al., 2014). Because the early layers of the network extract generic features, we froze the weights of the first two blocks of the



**Fig. 3.** Examples of the images with synthetic fractures B. (a) Fine high dip fracture. (b and c) Fractures with variable aperture sizes. (c) Overlapped fractures. The resulting images had different scales, sizes, brightness, and contrast.



**Fig. 4.** The Mask R-CNN architecture used in this study; the model outputs a classification label, a bounding box, and a binary mask for each fracture in the input image.

ResNet-50-FPN and re-trained the last three blocks and the Mask R-CNN head. To avoid overfitting, we trained the model gradually through four stages; the specified layers were trained for 15 epochs during each stage. In the first stage, only the head of Mask R-CNN was trained, and the entire backbone was frozen, then during each later stage, a backbone block was fine-tuned (unfrozen) starting from the last block. In the first two stages, a learning rate of 0.005 was initialized and decreased by 0.1 every five epochs using a learning rate scheduler. For the next two stages, a smaller learning rate of 0.001 was used. The model was optimized using a stochastic gradient descent optimizer with 0.9 momentum and 0.0005 weight decay. Mask R-CNN uses a multi-task loss calculated as a summation of classification, box regression, and segmentation losses. To improve the mask detection, we multiplied the segmentation loss by a factor of 1.5, following Xu et al. (2020). The total loss was, thus, defined as

$$\text{loss} = l_{\text{cls}} + l_{\text{box}} + 1.5 l_{\text{mask}}, \quad (2)$$

where  $l_{\text{cls}}$  is the classification loss calculated using the cross-entropy loss,  $l_{\text{box}}$  is the box regression loss calculated using the smooth L<sub>1</sub> loss (details in Girshick (2015)), and  $l_{\text{mask}}$  is the segmentation loss calculated as an average of the binary cross-entropy loss. Other training configurations remained as default.

### 2.2.2. Software and hardware details

The code was written in Python programming language (version. 3.7.7) (van Rossum and de Boer, 1991). For Mask R-CNN implementation, we used PyTorch deep learning library (version 1.5.0) (Paszke et al., 2019) and Torchvision (version 0.6.0). Other packages included OpenCV (version 3.4.2) (Bradski, 2000), Pillow (version 7.1.2) (Clark, 2021), Numpy (version 1.18.1) (Oliphant, 2006), Scikit-image (version 0.16.2) (Van Der Walt et al., 2014), and SciPy (version 1.5.2) (Virtanen et al., 2020).

The training and testing of the Mask R-CNN model were accelerated using an NVIDIA GeForce RTX 2080 Ti GPU with 11 GB of memory.

The code for creating synthetic fracture images, Mask R-CNN training, and the application of the proposed model is available on GitHub at <https://github.com/fatimahgit/Mask-RCNN-for-fracture-detection>.

### 2.3. Fracture characterization

The main steps of fracture characterization were: (1) pre-processing

the predicted fracture masks to extract fracture pixels, (2) fitting the extracted pixels to a sine wave, and (3) calculating fracture statistics from the parameters of the fitted sine wave. As discussed in the introduction, we assumed the case of a vertical core. For inclined and oriented cores more complex fitting equations are required along with the core inclination angle and orientation, which are explained in Chakraborty and Mukherjee (2020). For a real interpretation, the fitting function needs to represent the specific borehole orientation, which requires additional information besides the core images.

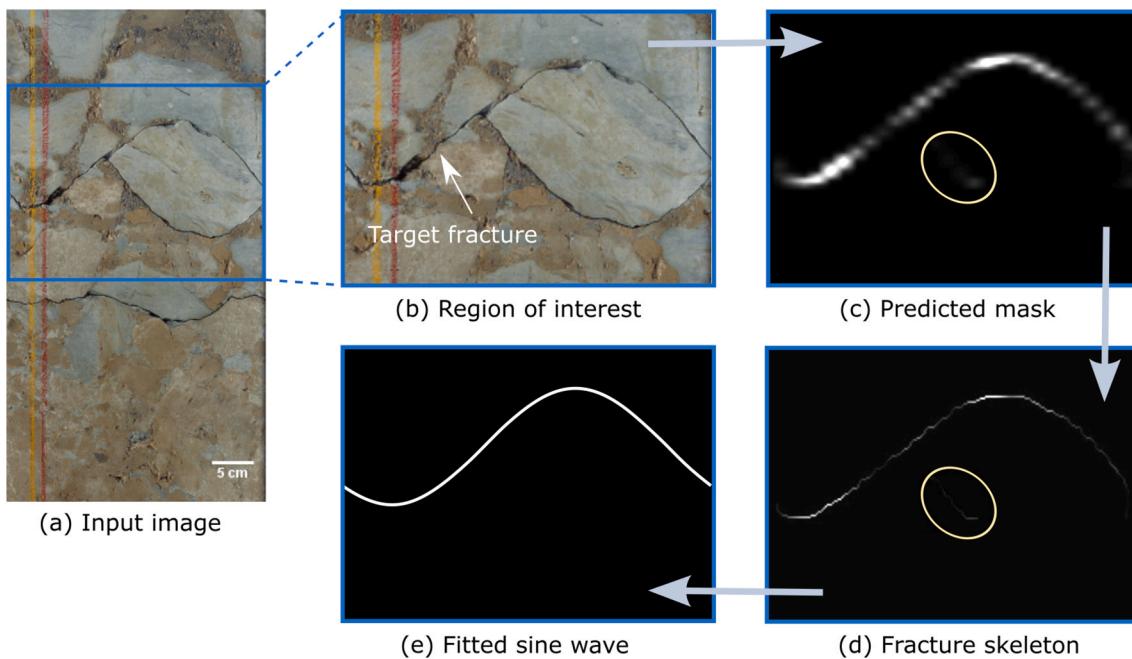
To accelerate the above operations, the images—whose sizes were up to 8000 × 2000 pixels—were first downsampled by a factor of 0.25. Each pixel in the predicted masks had a value between 0 and 1 indicating a background or fracture region, respectively; all non-zero pixels were used to define the fractures. Then a 1-pixel skeleton was created from each mask, using morphological image processing from Scikit-image (Van Der Walt et al., 2014), to represent the fracture shape (Fig. 5).

After extracting fracture points, sine wave fitting was performed through two steps: (1) primary fitting and (2) final fitting. Both fittings were achieved using non-linear least squares from the SciPy optimization library (Virtanen et al., 2020). The purpose of the primary fitting was to exclude peculiar points that result from imperfect segmentation or skeletonization, as shown by the yellow circle in Fig. 5c&d. A sine wave was fitted to the fracture points with a range of wavelengths of 0.5–2 w, where w is the image width. The primary fitting function was similar to Eq. (1) with the addition of variable l to represent the wavelength, producing

$$f(x) = \text{loc} + a \sin\left(l \frac{2\pi}{w}(x - \alpha)\right). \quad (3)$$

In addition, the original pixel scores were used during the primary fitting to determine the uncertainty in the fracture points. During the minimization of the cost function  $\sum \left(\frac{r}{\sigma}\right)^2$ , each least squares residual, r, was divided by its error  $\sigma$ , obtained as  $\frac{1}{\text{pixel score}}$ . This significantly reduced the influence of low scored points, and thus gave more weight to those with higher confidence of belonging to the fracture. In the final fitting, a sine wave of a wavelength equals to w was fit to the remaining points using Eq. (1). In Fig. 5e, the final sine fitting followed the correct path of the detected fracture without being disturbed by the inaccurate segment (circled in yellow) as it was eliminated by the primary fitting.

The parameters from the fitted sine wave were used to obtain fracture depth, dip angle, and dip direction. The relationship between these



**Fig. 5.** Post-processing and sine fitting of a detected fracture. Fracture skeleton was obtained from the predicted mask, with different scores (pixel intensities), then a sine wave was fitted to the skeleton points. The circled region shows an imperfect segmentation with low fracture scores removed by the primary fitting and had no effect on the final fit in (e).

parameters and the fracture on the unwrapped image is demonstrated in Fig. 6. The location of a fracture in the image was obtained from the centerline of the sine wave,  $loc$  in Eq. (1), and then converted from pixel to meter based on the image scale. Next, the depth of the fracture was calculated by adding its location on the image to the starting depth of the core in the image. This step was performed automatically by obtaining the starting depths from the names of the image files; thus, it depends on the way that the core images were labeled.

The dip angle was calculated from the amplitude of the sine function and core diameter, using the following equation

$$\tan(\theta) = 2A \frac{\pi}{w}, \quad (4)$$

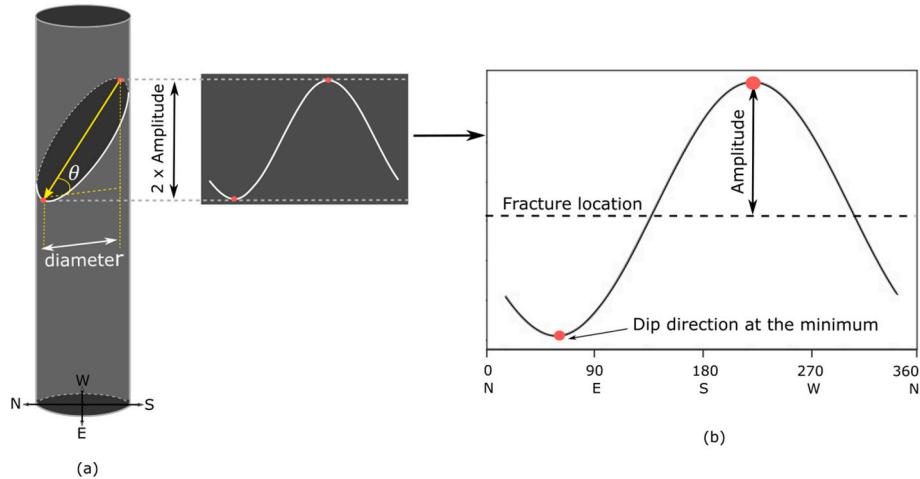
where  $\theta$  is the dip angle,  $A$  the amplitude of the sine wave, and  $w$  is the image width.

Finally, the dip direction was obtained from the minimum point in the sine wave (Rider, 1996) and converted from pixels to degrees. For

reasonable evaluation of dip direction, it was calculated only for fractures with more than  $5^\circ$  dip angle. This was because for low dip fractures ( $<5^\circ$ ) the location of the minimum was uncertain especially when the fracture path did not represent a perfect sine wave.

#### 2.4. Evaluation metrics

The performance of the model on the validation dataset was evaluated using common assessment metrics for instance segmentation models. Precision and recall are usually calculated as an average for a range of intersection over union (IoU) thresholds of 0.5–0.95 between the detected and the ground truth boxes or masks. Precision is defined as  $\frac{TP}{TP+FP}$ , representing the ratio of true positives  $TP$  to all detections ( $TP$  and false positives  $FP$ ). Recall is defined as  $\frac{TP}{TP+FN}$ , reporting the ratio of  $TP$  to all ground truth objects. The IoU threshold determines whether the detection is true positive depending on its IoU with the corresponding ground truth box or mask.



**Fig. 6.** Cylindrical vertical core with a fracture in (a) and the representation of the unwrapped fracture as a sine wave in (b).

For the test images, IoUs were not calculated as the images were used to assess the prediction of fracture depth, dip angle, and dip direction. The results were instead reported using precision and recall. *TP*, *FP*, *FN* were determined by comparing the depths for the detected fractures to the ground truth depths and by visual inspection. For assessing the accuracy of dip angle and dip direction results, absolute errors were computed between the predicted and ground truth values. Ground truth dip angles and dip directions were obtained using sine wave parameters derived from manually annotated fracture skeletons.

We defined a threshold for including fractures based on the classification score, which indicates the certainty of the fracture detection; only fractures with a score greater than the threshold were included. We tested the model with two threshold scores of 0.3 and 0.1.

In addition, the Mask R-CNN model allows defining an IoU threshold for the Non-maximum Suppression (NMS) to exclude detected boxes with large overlapping. This threshold was left as the default value of 0.5. Instead, we used the distance between the centerlines of the detected fractures to remove multi-detections of a single fracture. We defined a minimum distance of 15 mm, which was found suitable for the fractures in our dataset.

### 3. Results and discussion

#### 3.1. Fracture detection results

This section presents the results of the Mask R-CNN model for the validation and test datasets described in Section 2.1, using the evaluation metrics explained in Section 2.4.

For the validation dataset, the model achieved bounding box detection of 76.9% average precision (0.5–0.95 IoU thresholds). Based on an IoU threshold of 0.5, the precision of detection was 89.6%. The pixel-wise segmentation of detected fractures achieved less accurate results. The mask segmentation of the detected fractures had a 3.3% average precision for 0.5–0.95 IoU thresholds and a 10.1% precision for the 0.5 IoU threshold. This was possibly due to the limited capacity of the segmentation branch of Mask R-CNN that was insufficient to accurately segment the high-resolution core images; fracture regions in the core images had roughly up to 2000 pixels on the large edge. The resulting segmentation quality, however, mainly affected segmentation of fracture boundaries whereas fracture skeletons were still correctly recognized.

The model achieved promising detection results from the test images from BH1 and BH2 despite being mostly trained on synthetic images. Based on a classification threshold score of 0.3, the model resulted in a detection precision of 92.3% and a recall of 88.9% on the test images from BH1, which was an entirely new borehole. For BH2, the model achieved a higher precision of 98.3% and a slightly lower recall of 84.4%. After reducing the classification threshold score to 0.1, the model was expected to output more detections with less confidence, producing more false positives and fewer false negatives. Accordingly, the average recall of both boreholes increased by 1.6% while the average precision decreased by 4.3%, as shown in Table 2. Lowering the threshold score resulted in a slight increase in the recall compared to a large reduction in the precision. Thus, the results for the remainder of

the study used a classification threshold score of 0.3. Based on this threshold score, the differences between the depths of the correctly detected fractures and ground truth fractures were on average 3 mm for BH1 and 2 mm for BH2.

As shown in Table 3, BH1 had lower precision and higher recall than BH2 because the detection produced more false positives and fewer false negatives than BH2. The false positives in BH1 occurred mostly at the core edges as the images had a black background, which were not included during the ground truth labeling of fractures. The false negative in BH2 occurred due to: (1) the high downsampling ratio where the images from this borehole had originally higher resolution, (2) failure to detect some fine fractures, and (3) the presence of adjacent fractures within a less than 15 mm distance. Such errors can be addressed by including more complicated structures in the training data. In addition, the detection of fine cracks with higher resolution images than the current inputs would be an approach to reduce the false negatives. However, increasing the input image size is computationally expensive and requires more memory on the GPU. An alternative method would be using a smaller scanning window, which can potentially provide better local detection and segmentation but requires additional post-processing to filter out the detections and connect the segments of a single fracture.

#### 3.2. Fracture characterization results

After obtaining the correctly detected fracture masks from Section 3.1, we followed the procedure in Section 2.3 to calculate fracture depths, dip angles, and dip directions. Dip angle calculations achieved promising results from both boreholes. The calculated dips were in agreement with true dips obtained based on manual labeling of fractures (Fig. 7a&b). Dip angle measurements resulted in absolute errors of 2.6° and 1.8° for BH1 and BH2, respectively.

According to the overall error analysis of 210 detected fractures from both boreholes, 80% of the fractures had absolute error of 3° or less and 7% had error greater than 5° (Fig. 7c). The average error for all fractures was 2.2°. The error mainly resulted from inaccurate fracture segmentation and when a fracture did not represent a perfect sine wave.

Dip direction results were less accurate than dip angle results. Because dip direction depends on the lowest point of the sine wave, it is highly sensitive to the sine fitting and, in turn, to the fracture segmentation. The estimated dip direction could result in greater than 5° absolute error even when the detected and true sine waves were almost identical with coefficient of determination ( $R^2$ ) greater than 0.9. For example, a correctly detected fracture with an  $R^2$  of 0.99 between its detected and true sine waves (Fig. 8a) had a dip direction error of approximately 6°. Another fracture, shown in Fig. 8b, with a lower  $R^2$  value of 0.94, had approximately 7° absolute error.

Consequently, the dip direction calculation obtained average absolute error of 11.5° and 9.8° for BH1 and BH2, respectively. However, the calculated dip directions were still in good agreement with true dip directions, see Fig. 9a&b. Furthermore, the error analysis of a total of 186 detected fractures from both boreholes showed that greater than 75% of the points occurred with an error of less than 10.6°, while only 5% had errors greater than 40° (Fig. 9c).

High error occurred due to improper alignment of the fitted sine waves with the true path of the fractures, which resulted mostly from the

**Table 2**

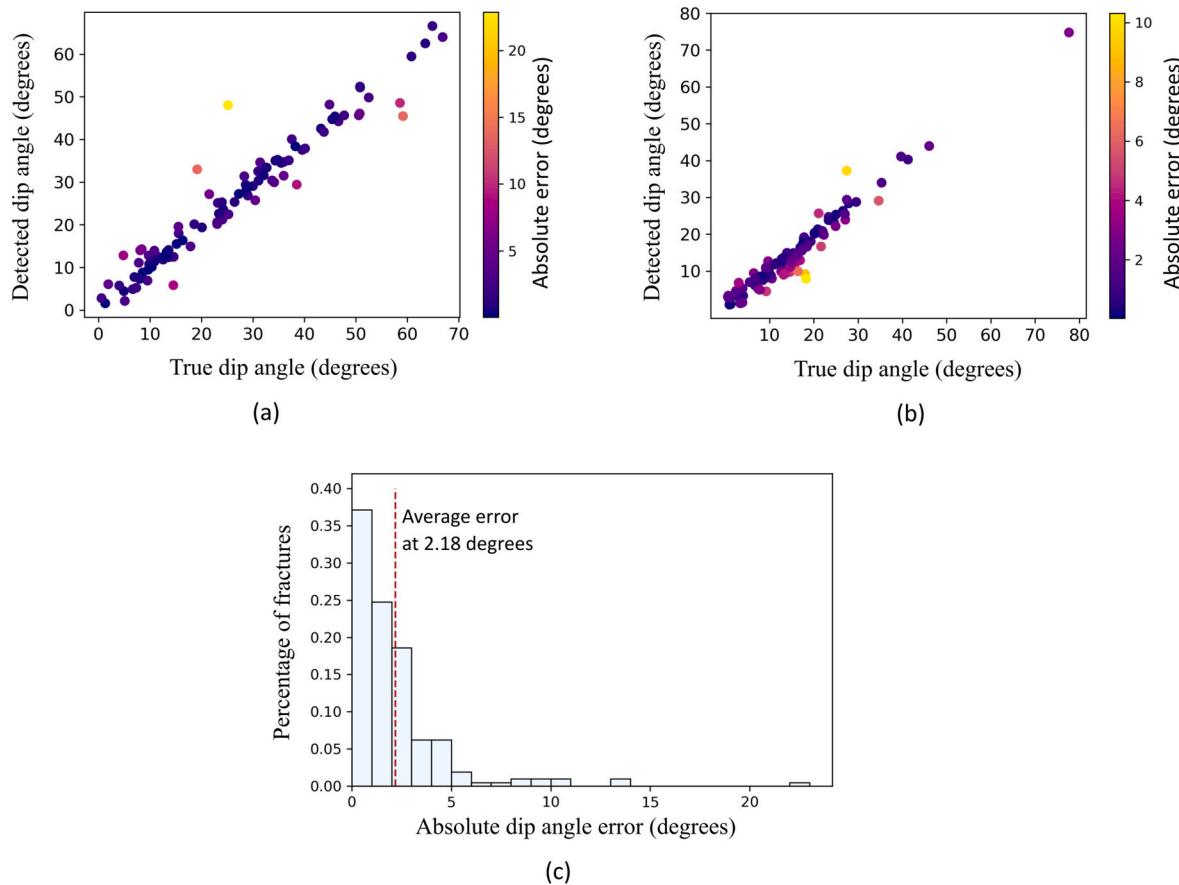
The results of fracture detection from the test images using two classification threshold scores. A threshold score of 0.3 resulted in an overall better detection than 0.1.

|                       | BH1  | BH2  | Average |
|-----------------------|------|------|---------|
| Threshold score = 0.3 |      |      |         |
| Precision (%)         | 92.3 | 98.3 | 94.8    |
| Recall (%)            | 88.9 | 84.4 | 86.7    |
| Threshold score = 0.1 |      |      |         |
| Precision (%)         | 85.1 | 95.9 | 90.5    |
| Recall (%)            | 89.8 | 86.8 | 88.3    |

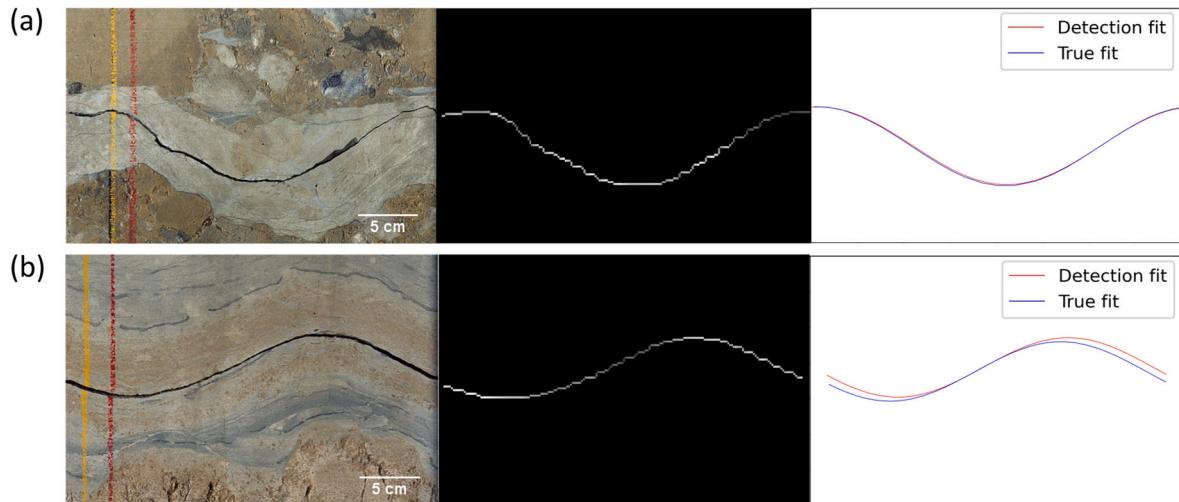
**Table 3**

The number of true positives, false positives and false negatives for the test images based on 0.3 threshold score.

|                     | BH1 | BH2 |
|---------------------|-----|-----|
| True positives      | 96  | 114 |
| False positives     | 8   | 2   |
| False negatives     | 12  | 21  |
| Total ground truths | 108 | 135 |



**Fig. 7.** The results of dip angle calculations for BH1 (a) and BH2 (b); the color indicates absolute error for each fracture. (c) Shows the error histogram calculated for all true positives. The results had a small overall error; most points had good agreement with the true dip angles. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

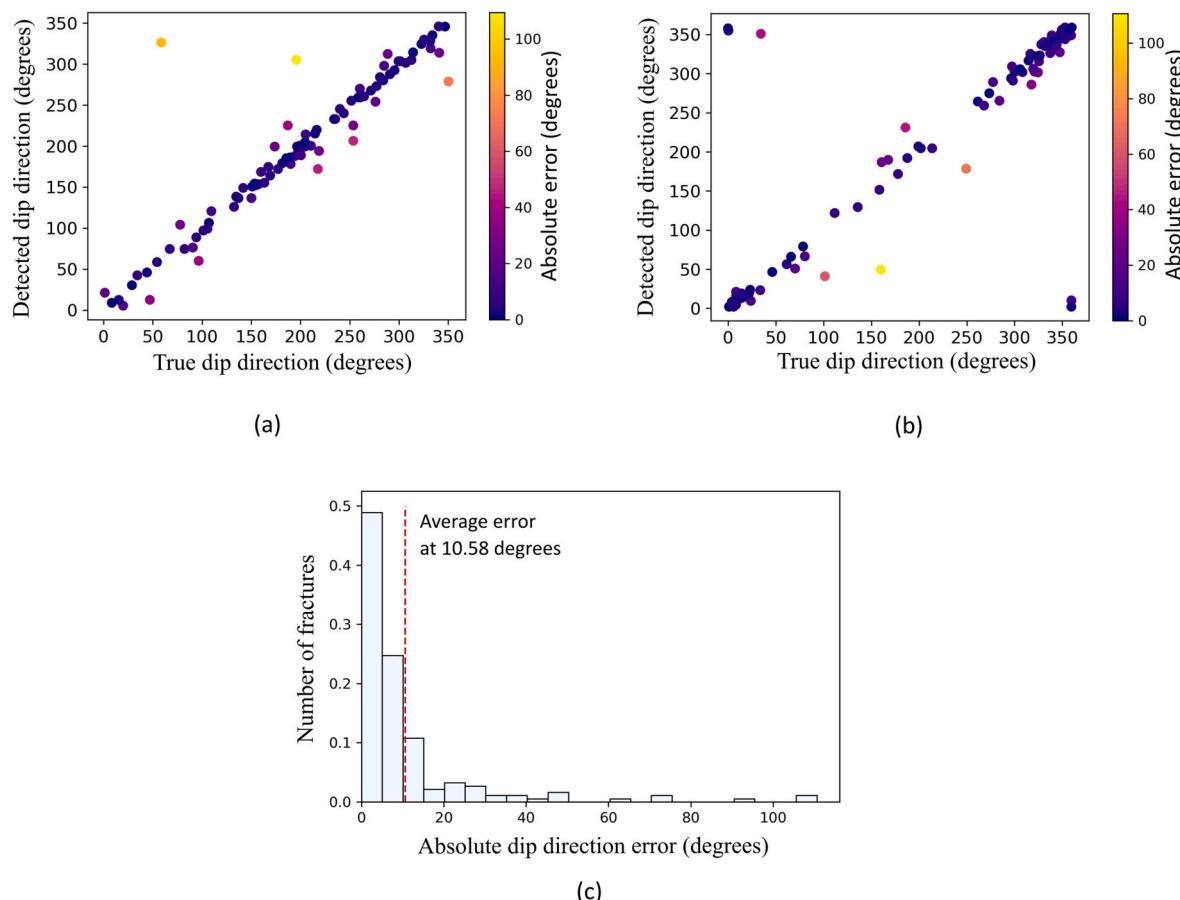


**Fig. 8.** The sensitivity of dip direction calculations to the sine fitting results. Almost perfect agreement between the detected and true sine fits with  $R^2$  of 0.99 in (a) and 0.94 in (b) resulted in approximately  $6^\circ$  error and  $7^\circ$  error, respectively.

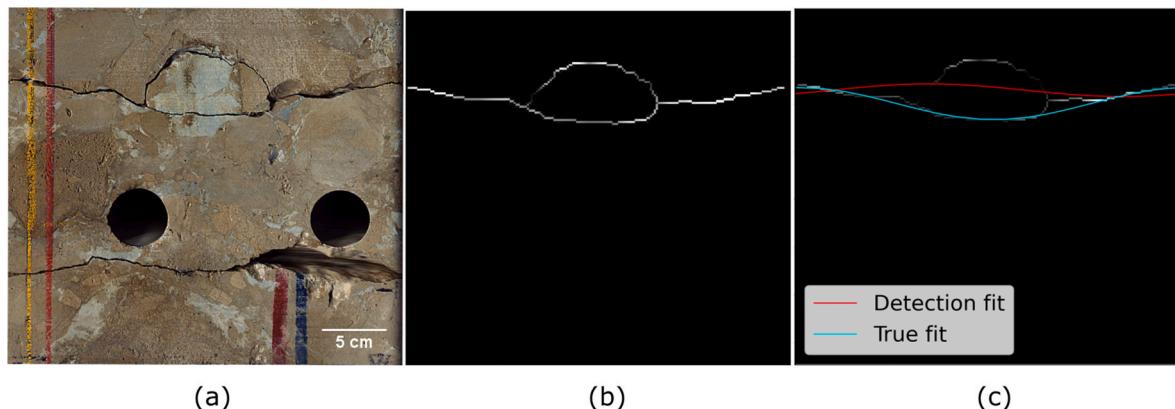
misdetection of other segments especially those with high segmentation scores. A prime example of this is provided in Fig. 10, where inaccurate segmentation in (b) disturbed the sine wave fitting represented by the red line in (c), and thus affected the calculated dip direction.

### 3.3. Overall evaluation

The model achieved results for dip angle and dip direction calculations with less than  $3^\circ$  and  $11^\circ$  average absolute errors, respectively, and for detection with almost 95% precision. The total time required to process 1 m of core was less than 1 s for BH1 and 7 s for BH2, using the GPU described in Section 2.2.2. An example of the test images and their



**Fig. 9.** The results of dip direction calculations for BH1 (a) and BH2 (b); the color indicates absolute error for each fracture. (c) Shows the error histogram calculated for all true positives. The results had a reasonable overall error; most points had good agreement with the true dip directions. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

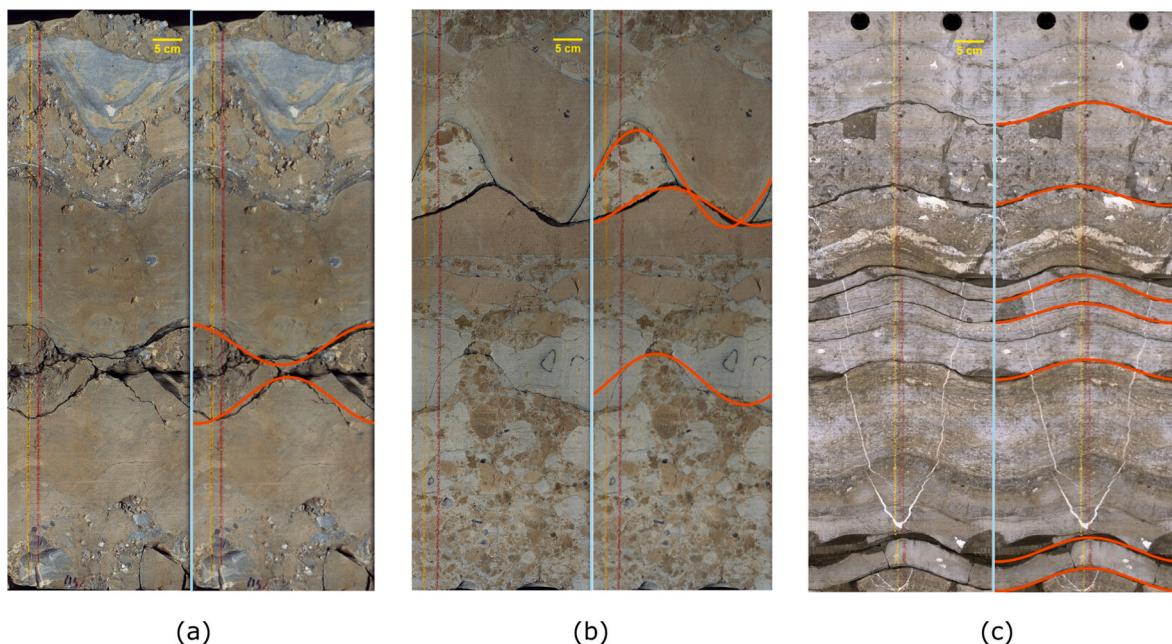


**Fig. 10.** An example of high dip direction error due to misdetection and incorrect sine fitting. (a) Input image, (b) detected fracture skeleton, (c) fitted sine waves from detection and manual segmentation.

final fracture detections are presented in Fig. 11. For these images, the model was able to recognize most of the fractures with a wide range of dips, directions, apertures, and levels of complexity. Moreover, the detections were not affected by bedding lines or core texture, as evident in the provided images.

The approach could be improved by addressing the current main limitations. Due to data constraints, the model was trained mainly on core images with synthetic fractures. The synthetic images allowed for creating desirable structures (e.g., complex and overlapped fractures) in

the training data but in future, including real fractures can potentially improve the model performance and robustness. Moreover, the test images from both boreholes included primarily carbonate rocks; the performance of the model on other rock types, such as shale was not evaluated. In terms of the pixel-wise segmentation, the fracture masks had limited accuracy at the fracture boundaries, which restricted the derived fracture characteristics. This limitation will be addressed in our future work that will focus on enhancing the boundary segmentation and investigating the feasibility of obtaining additional fracture



**Fig. 11.** Overall fracture detection of the model, showing test images from BH1 in (a and b) and from BH2 in (c), and their detected sine waves in red. Most fractures were well captured and characterized. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

attributes. Overall, the model can be improved by expanding the training and testing data as well as exploring different architectures of the instance segmentation model.

#### 4. Conclusion

The study offers a fully automated workflow to provide detailed fracture characterization from unwrapped core images. The workflow was developed based on an advanced ML model for instance segmentation, Mask R-CNN, that has been successfully applied for detection and segmentation tasks in other domains, such as medicine and remote sensing, yet remained unused for geosciences until now.

The Mask R-CNN model was trained mostly on synthetic data and tested on real core images from two boreholes. The tested images included a total of 88 m of core with 243 fractures. The model achieved average detection precision and recall of almost 95% and 87%, respectively. For the characterization of the detected fractures, we relied on the segmented fracture masks by fitting a sine wave to the detected fracture points. The parameters of the resulting sine wave were used to calculate fracture depth, dip angle, and dip direction. Fracture characterization achieved excellent results from dip angle calculations with an average absolute error of approximately 2°. Although dip direction had a higher average error of approximately 11°. The total processing time depended on the image resolution, and on average processing time was 4 s per meter of core. Hence, the proposed approach can provide rapid, accurate, and inexpensive fracture detection and characterization from core images, which can be integrated with results from image log interpretation. The method can be easily adopted in industry and further developed for different rock types and to enhance its robustness and accuracy by expanding the used datasets.

With the rapid development in ML and the continued improvement of instance segmentation models, we intend to explore the performance of other frameworks for fracture detection, and particularly for fracture segmentation. In addition, future work should focus on improving the segmentation accuracy of the current method, extending the method to classify different types of fractures, and provide a more in-depth geological interpretation for the region studied by integrating the unwrapped core images with log data.

#### Author contributions

**Fatimah Alzubaidi:** Methodology, Investigation, Writing - Original Draft. **Patrick Makuluni:** Validation. **Stuart R. Clark:** Data acquisition and analysis, Validation, Writing – Review & Editing. **Jan Erik Lie:** Data acquisition, Writing – Review & Editing. **Peyman Mostaghimi:** Conceptualization, Writing – Review & Editing. **Ryan T. Armstrong:** Conceptualization, Writing – Review & Editing, Supervision.

#### 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 authors acknowledge Lundin Energy for providing the unwrapped core images used in the study.

#### References

- Agosta, F., Prasad, M., Aydin, A., 2007. Physical properties of carbonate fault rocks, fucino basin (Central Italy): implications for fault seal in platform carbonates. *Geofluids* 7, 19–32. <https://doi.org/10.1111/j.1468-8123.2006.00158.x>.
- Betlem, P., Birchall, T., Ogata, K., Park, J., Skurtveit, E., Senger, K., 2020. Digital drill core models: structure-from-motion as a tool for the characterisation, orientation, and digital archiving of drill core samples. *Rem. Sens.* 12 <https://doi.org/10.3390/rs12020330>.
- Bradski, G., 2000. The OpenCV library. *Dr Dobb's J. Softw. Tools* 25, 120–125.
- Chakraborty, M., Mukherjee, S., 2020. Structural geological interpretations from unrolled images of drill cores. *Mar. Petrol. Geol.* 115, 104241. <https://doi.org/10.1016/j.marpgeo.2020.104241>.
- Chiao, J.-Y., Chen, K.-Y., Liao, K.Y.-K., Hsieh, P.-H., Zhang, G., Huang, T.-C., 2019. Detection and classification of the breast tumors using mask R-CNN on sonograms. *Medicine (Baltimore)* 98, e15200. <https://doi.org/10.1097/MD.00000000000015200>.
- Clark, A., 2021. Pillow (PIL Fork) Documentation.
- Cruz, R.A.Q., Cacau, D.C., dos Santos, R.M., Ribeiro Pereira, E.J., Leta, F.R., Gonzalez Clua, E., 2017. 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). IEEE, pp. 345–350. <https://doi.org/10.1109/SIBGRAPI.2017.52>.
- Dias, L.O., Bom, C.R., Faria, E.L., Valentín, M.B., Correia, M.D., de Albuquerque, Márcio, P., de Albuquerque, Marcelo, P., Coelho, J.M., 2020. Automatic detection of

- fractures and breakouts patterns in acoustic borehole image logs using fast-region convolutional neural networks. *J. Petrol. Sci. Eng.* 191, 107099. <https://doi.org/10.1016/j.petrol.2020.107099>.
- Elkington, P.A.S., Assous, S., 2017. Methods of and apparatuses for identifying geological characteristics in boreholes. U. S. Jpn. Outlook, 9,704,263 B2.
- Fan, J., Upadhye, S., Worster, A., 2006. Understanding receiver operating characteristic (ROC) curves. *Can. J. Emerg. Med.* 8, 19–20. <https://doi.org/10.1017/S1481803500013336>.
- Fernández-Ibáñez, F., DeGraff, J.M., Ibrayev, F., 2018. Integrating borehole image logs with core: a method to enhance subsurface fracture characterization. *Am. Assoc. Petrol. Geol. Bull.* 102, 1067–1090. <https://doi.org/10.1306/0726171609317002>.
- Garcia-Garcia, A., Orts-Escalano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J., 2017. A Review on Deep Learning Techniques Applied to Semantic Segmentation (arXiv e-prints).
- Girshick, R., 2015. Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, pp. 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>.
- Hall, J., Ponzi, M., Gonfalone, M., Maletti, G., 1996. Automatic extraction and characterisation of geological features and textures from borehole images and core photographs. In: SPWLA 37th Annual Logging Symposium. Louisiana, New Orleans.
- Hansen, B.E., 2019. Imaging Reservoir Properties of the Loppa High, Norwegian Barents Sea Examples from Alta Discovery. University of Oslo.
- He, K., Gkioxari, G., Dollar, P., Girshick, R., 2017. Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision. IEEE, pp. 2961–2969. <https://doi.org/10.1109/ICCV.2017.322>.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Las Vegas, NV, USA, pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25, 1097–1105.
- Lai, J., Chen, K., Xin, Y., Wu, X., Chen, X., Yang, K., Song, Q., Wang, G., Ding, X., 2021. Fracture characterization and detection in the deep Cambrian dolostones in the Tarim Basin, China: insights from borehole image and sonic logs. *J. Petrol. Sci. Eng.* 196, 107659. <https://doi.org/10.1016/j.petrol.2020.107659>.
- Laubach, S.E., Monson, E.R., 1988. Coring-induced fractures: indicators of hydraulic fracture propagation in a naturally fractured reservoir. In: SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers. <https://doi.org/10.2118/18164-MS>.
- Lemy, F., Hadjigeorgiou, J., Côté, P., Maldague, X., 2001. Image analysis of drill core. *Min. Technol.* 110, 172–177. <https://doi.org/10.1179/mnt.2001.110.3.172>.
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125.
- Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft COCO: common objects in context. In: European Conference on Computer Vision. Springer Verlag, pp. 740–755. [https://doi.org/10.1007/978-3-319-10602-3\\_48](https://doi.org/10.1007/978-3-319-10602-3_48).
- Liu, M., Dong, J., Dong, X., Yu, H., Qi, L., 2018. Segmentation of Lung Nodule in CT images based on mask R-CNN. In: 2018 9th International Conference on Awareness Science and Technology, ICAST 2018. Institute of Electrical and Electronics Engineers Inc., pp. 95–100. <https://doi.org/10.1109/ICAwST.2018.8517248>.
- Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.
- Lundin Energy. URL. <https://www.lundin-energy.com/>. accessed 10.14.20.
- Maini, R., Aggarwal, H., 2009. Study and comparison of various image edge detection techniques. *Int. J. Image Process.*
- Massiot, C., Townsend, J., Nicol, A., McNamara, D.D., 2017. Statistical methods of fracture characterization using acoustic borehole televIEWER log interpretation. *J. Geophys. Res. Solid Earth* 122, 6836–6852. <https://doi.org/10.1002/2017JB014115>.
- Maxwell, A.E., Pourmohammadi, P., Poyner, J.D., 2020. Mapping the topographic features of mining-related valley fills using mask R-CNN deep learning and digital elevation data. *Rem. Sens.* 12, 547. <https://doi.org/10.3390/rs12030547>.
- Nelson, R.A., 2001. Reservoir management. In: Geologic Analysis of Naturally Fractured Reservoirs.
- Nian, T., Wang, G., Xiao, C., Zhou, L., Sun, Y., Song, H., 2016. Determination of in-situ stress orientation and subsurface fracture analysis from image-core integration: an example from ultra-deep tight sandstone (BSJQ Formation) in the Kelas Belt, Tarim Basin. *J. Petrol. Sci. Eng.* 147, 495–503. <https://doi.org/10.1016/j.petrol.2016.09.020>.
- Nie, X., Duan, M., Ding, H., Hu, B., Wong, E.K., 2020. Attention mask R-CNN for ship detection and segmentation from remote sensing images. *IEEE Access* 8, 9325–9334. <https://doi.org/10.1109/ACCESS.2020.2964540>.
- Oliphant, T.E., 2006. Guide to NumPy.
- Ozturk, H., Saricam, I.T., 2018. Core segmentation and fracture path detection using shadows. *J. Image Graph.* 6, 69–73. <https://doi.org/10.18178/joig.6.1.69-73>.
- Papageorgiou, C.P., Oren, M., Poggio, T., 1998. General framework for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE, pp. 555–562. <https://doi.org/10.1109/iccv.1998.710772>.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* 32, 8026–8037.
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788.
- Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: towards Real-Time Object Detection with Region Proposal Networks (arXiv e-prints).
- Rider, M.H., 1996. The Geological Interpretation of Well Logs, second ed. Rider-French Consulting Ltd.
- Sayago, J., 2014. Late Paleozoic Basin Analysis of the Loppa High and Finnmark Platform in the Norwegian Barents Sea: Integration of Seismic Attributes and Seismic Sequence Stratigraphy. The University of Potsdam.
- Sayago, J., Di Lucia, M., Mutti, M., Sitta, A., Cotti, A., Frijia, G., 2018. Late Paleozoic seismic sequence stratigraphy and paleogeography of the paleo-Loppa High in the Norwegian Barents Sea. *Mar. Petrol. Geol.* 97, 192–208. <https://doi.org/10.1016/j.marpgeo.2018.05.038>.
- Schodlok, M.C., Whitbourne, L., Huntington, J., Mason, P., Green, A., Berman, M., Coward, D., Connor, P., Wright, W., Jolivet, M., Martinez, R., 2016. HyLogger-3, a visible to shortwave and thermal infrared reflectance spectrometer system for drill core logging: functional description. *Aust. J. Earth Sci.* 63, 929–940. <https://doi.org/10.1080/08120099.2016.1231133>.
- Shanahan, R.J., Mahtab, M.A., 1976. Delineation and analysis of clusters in orientation data. *Math. Geol.* 8.
- Shrivakshan, G.T., Chandrasekar, C., 2012. A comparison of various edge detection techniques used in image processing. *Int. J. Comput. Sci. Issues* 9, 269–276.
- Supervisely. URL. <https://supervisely.ly/>. accessed 10.20.20.
- Tiwari, S., Mishra, S., Srihariprasad, G., Vyas, D., Warhade, A., Nikalje, D., Bartakke, V., Mahesh, B., Tembhurnikar, P., Roy, S., 2017. High resolution core scan facility at BGRL-MoES, Karad, India. *J. Geol. Soc. India* 90, 795–797. <https://doi.org/10.1007/s12594-017-0793-6>.
- Van Der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., 2014. Scikit-image: image processing in python. *PeerJ* 2014 e453. <https://doi.org/10.7717/peerj.453>.
- Van Golf-Racht, T.D., 1982. Fundamentals of Fractured Reservoir Engineering. Elsevier Science & Technology, Oxford.
- van Rossum, G., de Boer, J., 1991. Interactively testing remote servers using the Python programming language. *CWI Q* 4, 283–304.
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, R., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., Vijaykumar, A., Bardelli, A., Pietro Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C.N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D.A., Hagen, D.R., Pasechnik, D.V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G.A., Ingold, G.L., Allen, G.E., Lee, G.R., Audren, H., Probst, I., Dietrich, J.P., Silterra, J., Webber, J.T., Slavić, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J.L., de Miranda Cardoso, J.V., Reimer, J., Harrington, J., Rodríguez, J.L.C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmeler, M., Bolingbroke, M., Tartrre, M., Pak, M., Smith, N.J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P.A., Lee, P., McGibbon, R.T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T.J., Robitaille, T.P., Spura, T., Jones, T.R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y.O., Vázquez-Baeza, Y., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>.
- Wedge, D., Holden, E.J., Dentith, M., Spadaccini, N., 2017. Fast and objective detection and analysis of structures in downhole images. *J. Appl. Geophys.* 144, 157–172. <https://doi.org/10.1016/j.jappgeo.2017.07.004>.
- Xavier, A., Guerra, C.E., Andrade, A., 2015. Fracture analysis in borehole acoustic images using mathematical morphology. *J. Geophys. Eng.* 12, 492–501. <https://doi.org/10.1088/1742-2132/12/3/492>.
- Xu, S., Lan, S., Zhu, Q., 2020. MaskPlus: improving mask generation for instance segmentation. In: Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, pp. 2019–2027. <https://doi.org/10.1109/WACV45572.2020.9093379>.
- Ye, S.J.C., 2016. Automatic dip picking from wellbore azimuthal image logs. *U. S. Jpn. Outlook*, 9,366,135 B2.
- Zeng, L., Li, X.Y., 2009. Fractures in sandstone reservoirs with ultra-low permeability: a case study of the upper Triassic Yanchang formation in the Ordos basin, China. *Am. Assoc. Pet. Geol.* 93, 461–477. <https://doi.org/10.1306/09240808047>.