Crack Identification and Characterization in Deformed Nb3Sn Rutherford Cable Stacks Using Machine Learning

2MPo2A-01

Crack Identification and Characterization in Deformed Nb_3Sn Rutherford Cable Stacks using Machine Learning

Jean-Francois Croteau, Giorgio Vallone, Nandana Menon, Marika D'Addazio, Fabrizio Niccoli, Ian Pong, *Senior Member IEEE* Paolo Ferracin, *Senior Member, IEEE*, Soren Prestemon, *Senior Member, IEEE*

***Abstract*—An investigation of instance segmentation of cracks in Nb_3Sn 4-stack 40-strand Rutherford cables using machine learning is presented. Three samples were uniaxially and biaxially loaded before metallographic inspections were performed. The Mask R-CNN model was used in the Detectron2 framework with pre-trained weights but fine-tuned to detect and segment cracks. The model detected cracks with bounding box and mask average precisions (AP) of 42.8 and 27.9, respectively, and was used for instance segmentation of all cracks in the three samples. More cracks were found in the sample pre-loaded along the *z*-axis (i.e. along the cable length). Pre-loading along the *x*-axis (i.e. on the cables' edges) reduced the number of cracks and changed the crack orientation distribution, away from being highly aligned with the *y*-axis (i.e. normal to the cables' broad faces), i.e. the direction with the highest applied load. Fine-tuning of the Segment Anything Model (SAM) was also studied but performed poorly without human-provided prompts. However, the zero-shot capability of SAM showed high promises to accelerate the image annotation process for applications beyond this study.**

***Index Terms*—Machine learning, image analysis, Nb_3Sn, cracks, superconducting magnets**

I. Introduction

THE brittle nature of Nb_3Sn can lead to irreversible performance degradation in superconducting accelerator magnets [1]. Therefore, the development of Nb_3Sn magnets at conductor peak field above 11 T for the High-Luminosity upgrade of the Large Hadron Collider [2], [3] and R&D efforts for fields of up to 16 T for the Future Circular Collider [4] are increasing the complexity to assemble structures where stresses on the conductor at every location are not limiting or degrading performances. The complex stress states in the magnet during assembly, cooldown, and energization have motivated the development of numerical models [5], [6], [7], [8], [9] and experiments to study and predict the onset of crack formation for uniaxial [10], [11], [12] and biaxial deformation of Rutherford cables and stacks [9].

Optical and electron microscopy has been a key tool to investigate the location and nature of superconducting magnet and cable performance limitations for fusion and accelerator applications [13], [14], [15], [16]. Recently, cross-sections of the MQXF coils were analyzed [13] and helped identify the source of performance limitations in magnets by locating cracks in coil cross-section images.

To the best of the authors' knowledge, all previous studies have relied on humans to locate and count cracks in Nb_3Sn wires, cables, and coils. This process is slow, expensive, not scalable, and limits the analysis to crack count and approximate crack location. Automating the detection of cracks in these complex materials is challenging since the use of traditional image analysis techniques, such as thresholding that rely on pixel intensity, cannot be used alone due to the large number of features with similar colors, e.g. voids. Image analysis techniques that segment objects based on morphological descriptors are required.

Supervised machine learning (ML) models, such as convolutional neural networks, are well suited for such tasks as models are trained to segment components based on features. Moreover, recent progress in the field of computer vision for applications not related to material sciences, e.g. self-driving cars, has resulted in the development of faster and more accurate open-source models for object detection and instance and semantic segmentation. The models are often trained on very large datasets composed of everyday objects of varying complexity and shape and can be fine-tuned to detect or segment previously unseen objects. This process, often referred to as transfer learning, can simplify and accelerate the training process of existing models for scientific applications [17].

This paper presents an attempt to use two such models with very distinct architectures, namely Mask R-CNN [18] and Segment Anything Model [19], and designed with different

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, through the Magnet Development Program, under Contract DE-AC02-05CH11231, which is performed at Lawrence Berkeley National Laboratory. (*Corresponding author: Jean-Francois Croteau.)*

J.-F. Croteau, G. Vallone, N. Menon, I. Pong, P. Ferracin, S. Prestemon are with the Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA (e-mail: jcroteau@lbl.gov).

M. D'Addazio is with the Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA and with the Politecnico di Torino, Turin 10129, Italy.

F. Niccoli is with the Università della Calabria, Rende 97036, Italy.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org

This article has been accepted for publication in IEEE Transactions on Applied Superconductivity. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASC.2024.3513940

objectives in mind for instance segmentation of cracks in three Nb_3Sn 4-stack samples used in a previous study [9].

II. Experimental Procedure

A. Stack Preparation and Deformation

Three 4-stack MQXF Nb_3Sn Rutherford cables (40 strands, 0.85 mm diameter, RRP® 108/127 [2]) fabricated at Lawrence Berkeley National Laboratory and characterized in previous work [9] are used in this study. The samples were uniaxially or biaxially loaded at room temperature for three stress states listed in Table 1. More details on the experimental setup and the loading procedure are available in [9].

Table 1. Sample loading conditions. Sample coordinate axes are shown in Figure 2a.

Sample number	σ_x (MPa)	σ_y (MPa)	σ_z (MPa)
1	0	150	0
2	0	150	20
3	20	150	0

B. Metallography and Microscopy

The samples were polished using the procedure in [9]. A final step of overnight vibratory polishing with diamond suspension particles of 0.04 µm (Struers OP-S NonDry) was used to reveal the finest cracks. After polishing, samples were cleaned with soap and water, followed by an ultrasonic bath in water and drying by centrifugal force in a Struers Lavamin. The samples were further dried using hot air. Sample cleaning and drying was challenging due to the large number of small voids in each subelement that trap water and polishing solutions during overnight vibratory polishing. The trapped liquid has the tendency to oxidize the surface by creeping up to the surface of the pristine sample due to the high vacuum in a scanning electron microscope or to the local heat generated by the concentrated light of an optical microscope. Unsuccessful attempts were made to completely remove cleaning artefacts, which motivated the use of a robust machine learning model to detect cracks in the presence of artefacts, like those shown in Figure 1. If successful, this model could be used for future coil inspections.

After sample polishing, cleaning, and drying, images of all wires were acquired using a digital microscope (Olympus DSX-1000) with a 40x lens and an optical zoom of 1.4x. Images with a spatial resolution of about 0.28 µm/pixel, sufficient to see the smallest cracks in Nb_3Sn, were saved and used for analysis. Single wires were imaged by automatic stitching of multiple images in *x*-*y* and stacking in *z* was used to ensure that all regions of the wires were in-focus.

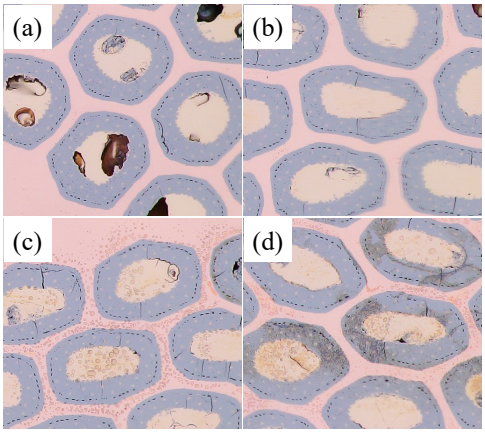

Figure 1. Examples of images with (a) no artefacts, (b) blurred features, a stitching artefact, (c) droplets, and (d) oxidized copper and subelements.

C. Dataset Preparation for Model Training

Figure 2 shows a schematic of the workflow used to prepare the dataset for training different machine learning models. First (a), an image of each 4-stack was acquired to study the distribution of cracks as a function of wire position in the stack. Second (b), high-magnification and high-resolution color images of individual wires were acquired. Third (c), the images were padded with black pixels and cropped in smaller images of 512 px x 512 px (144.17 µm x 144.17 µm), hereafter *patches*, to ease image annotation and to limit the memory requirements when analyzing the images. Fourth (d and e), 54 patches were selected for manual annotation of the ground truth. The patch selection process was done in a way to ensure that the images were representative of the entire dataset. Therefore, multiple images with the artefacts presented in Figure 1 were included. These images were annotated by hand by drawing polygons with near-pixel precision around cracks using MakeSense.ai [20]. The annotations were exported to a human-readable JavaScript Object Notation (JSON) file following the Common Object in Context (COCO) data format [21]. The annotated images were separated into two datasets with a ratio of approximately 80:20 for training (43 images with 332 cracks) and validation (11 images with 83 cracks), respectively. Finally, the annotated images were used to train the Mask R-CNN model and fine tune the Segment Anything Model (SAM).

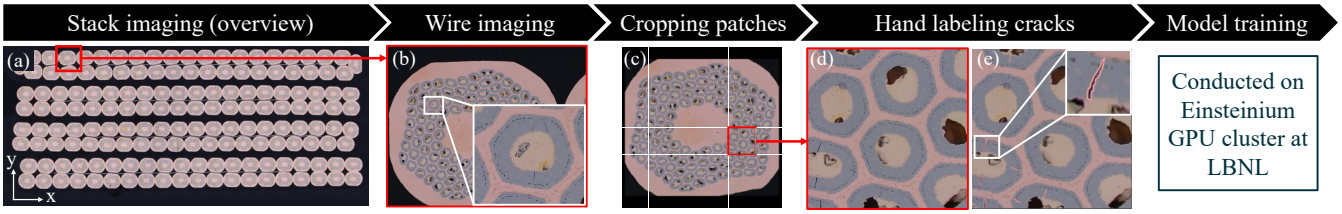

Figure 2. Schematic of the workflow used in this study, from image acquisition to machine learning model training. (a) Imaging of 4-stack, (b) high-resolution and high-magnification imaging of individual wires by automatic stitching and stacking, (c) padding and cropping the wires in 512 px x 512 px images, (d) single image from (c) before annotation, and (e) hand labeling of cracks for the training and validation datasets.

D. Image Analysis

Threshold-based segmentation or clustering of pixels with unsupervised machine learning methods like K-means or Gaussian mixture, are not suitable on their own for this problem as morphological aspects of the cracks must be considered to distinguish them from voids and other dark pixels. Two machine learning models were considered for this study: Mask R-CNN using the Detectron2 framework and Segment Anything Model. Both models were implemented in interactive Python notebooks and trained or fine-tuned on the Einsteinium GPU cluster hosted at Lawrence Berkeley National Laboratory.

Detectron2, developed by Meta's Fundamental AI Research (FAIR) [22], is an open-source framework that eases the use of different computer vision models for object and keypoint detection and instance and panoptic segmentation. Different models pre-trained on very large image datasets can directly be used or fine-tuned to detect specific features.

The Mask R-CNN architecture, i.e. a Region-based Convolutional Neural Network (R-CNN) architecture, with different model configurations was previously used in Detectron2 to detect pores and cracks in additively manufactured parts [23], for the detection of defects in synthetic fiber ropes [17], and it is used in this study for instance segmentation of cracks in Nb_3Sn. The Mask R-CNN model works in two steps. First, object detection is performed for all different classes. Note that only one class was used for this study: *crack*. Second, semantic segmentation of the foreground and the background pixels is performed in the box bounding the detected object.

The Segment Anything Model (SAM) is a segmentation foundation model that was also developed by Meta's Fundamental AI Research. The model was pre-trained on ~11 million images and 1.1 billion masks [19] with the objective of being able to segment any object in the foreground based on a prompt, i.e. a point, text, a bounding box, or a mask. The model's strength lies in its *zero-shot learning* i.e. the ability to segment unseen objects and features following simple input prompts. It also has the potential to be tuned to segment specific features, like for medical and satellite imagery [24], [25]. For this study, an attempt was made to fine-tune the model to segment cracks.

III. Machine Learning Models Comparison

A. Mask R-CNN Training and Performances

Image analysis using Mask R-CNN was conducted in Python using the Detectron2 framework. In this study, the model selected from Detectron2 uses ResNet-50 with a Feature Pyramid Network (FPN) and was pre-trained using a COCO dataset (*train2017* and *val2017*) with a 3x schedule, equivalent to about 37 COCO epochs [26]. In Detectron2, images go through a feature extraction step that differs based on the pre-trained model selected. The extracted features are then fed to Mask R-CNN which works in two steps. First, a Region Proposal Network (RPN) model uses a sliding window over the features to predict the detection of objects and outputs bounding box locations and corresponding classes for the second part of the model. All objects with a confidence score above a set threshold are kept. Second, semantic segmentation is conducted in each bounding box using a CNN that outputs a binary mask of the object. An overlay of these outputs on a segmented image is shown in Figure 3: (1) a bounding box around the crack, (2) a binary mask of the crack, (3) the corresponding class, and (4) a confidence score. Only objects with a confidence score above 60% were used.

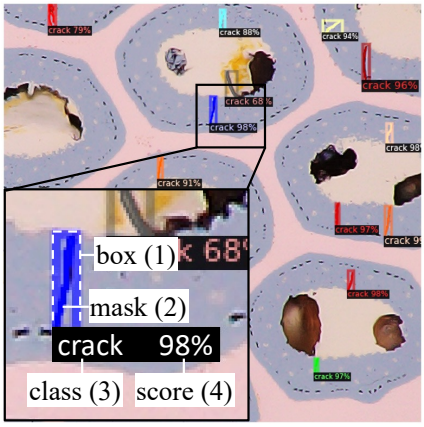

Figure 3. Segmented wire patch using Mask R-CNN. (Inset) List of attributes generated by Detectron2 for each crack.

The model was fine-tuned on a single GPU node using pre-trained weights from the Detectron2 repository and the 43 labeled patches with 332 cracks set aside, i.e. the training dataset. At each iteration, the total loss, i.e. the sum of the classification, bounding box, and mask losses defined in [18], was computed for the training and validation datasets to observe potential model overfitting. Figure 4 shows the training plot for the Mask R-CNN model. A deviation between the training and validation losses was observed after ~1,000 iterations, which justified stopping model training at that point to avoid overtraining and overfitting. The model and weights were exported from LBNL's Einsteinium GPU cluster as *.yaml* and *.pth* files after 1,000 iterations. The use of these file formats in combination with an open-source model should ease the ability to share the trained model with other users and institutes for instance segmentation of cracks in the future.

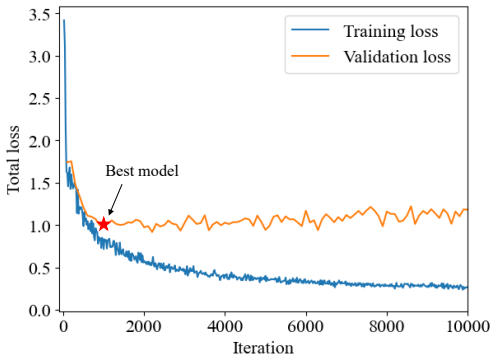

Figure 4. Total loss curves for 10,000 iterations during fine-tuning of the Mask R-CNN model.

Standard COCO evaluation metrics, such as the average precision (AP) for intersection over union (IoU) thresholds of 0.5 to 0.95, for 0.05 increments, were evaluated [27]. Precision is defined for bounding boxes and masks as the predicted true

positive over the sum of the predicted true and false positives. IoU is the intersection of the area of the predicted and the ground truth bounding boxes (or masks) divided by the union of the area of these bounding boxes (or masks). Bounding box and mask AP, averaged over all IoU thresholds, of 42.8 and 27.9, respectively, were calculated at the last training iteration. Mask mean IoU and Dice coefficient of 0.417 and 0.584 were computed for the validation dataset. In addition to these metrics to assess the model accuracy, visual inspection of the model's performance to segment cracks in the 11 patches in the validation dataset was conducted and confirmed the model's ability to detect and segment cracks with a high accuracy. Compared to SAM, discussed in Section III-B, the mean IoU and Dice coefficient are slightly lower, but the number of cracks detected was more accurate, which justified the use of Mask R-CNN for instance segmentation of the entire dataset.

To keep traceability of the location of cracks in a wire, a cable, and a stack, instance segmentation of patches with the fined-tuned Mask R-CNN model was performed on a wire-by-wire basis and the output mask of each patch was reassembled to obtain a binary mask of cracks for an entire wire. Some mislabeled cracks, i.e. false positives, were found for cracks and pores in the epoxy surrounding the wire and the cable glass-fiber insulation. As the training and validation datasets mainly included patches with subelements closer to the core of the wires, these mislabeled regions were not detected during training. Since these regions were all outside of the wires, the copper matrix was segmented by traditional image analysis using automatic thresholding with the multi-Otsu method [28]. Other image analysis functions from the Python scikit-image library [29] were used to clean and remove additional features in the wires, e.g. subelements and pores, and to obtain a binary mask of the wire. The intersection of the wire and the crack masks was used to remove all mislabeled cracks surrounding the wires. Since both machine learning instance segmentation and traditional image analysis were done in Python, this additional step was seamlessly integrated to the segmentation of cracks for the entire dataset.

B. Segment Anything Model Tuning

The image analysis using SAM was performed in two phases. First, a pre-trained model was used to leverage the model's zero-shot learning capability. This pre-trained model using the base architecture for the vision transformer was tested on the images to segment them without prior training. In this "segment everything" mode, the pre-trained SAM identifies masks to segment the entire image using an in-built automatic sampler. This sampler generates point-based prompts and filters them for quality and duplicates. Figure 5 shows the resultant masks that segment the image.

The pre-trained SAM, in its inference mode, was able to segment the background from the foreground and hence identify the subelements, cores, large voids, and artefacts such as droplets, as evident in Figure 5. However, it could not segment finer features such as the cracks using automatic sampling. The pretrained SAM segments everything, making it task-agnostic, and hence fails to delineate finer features of interest making it unsuitable for crack detection without providing appropriate prompts.

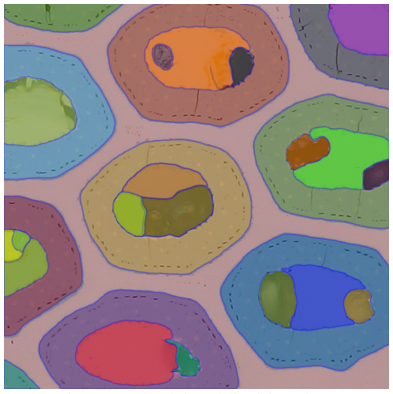

Figure 5. Segmentation masks identified by the pre-trained SAM in its segment everything mode on the same image as Figure 3.

Therefore, the second part of the analysis involved fine-tuning SAM to a portion of the images to allow the model to learn the features of interest i.e. cracks. The images used for training and validation were the same as those employed for the Mask R-CNN model in Section III-A. The 512 x 512 sized images as well as their corresponding masks were further divided into 256 x 256 sized patches to comply with the requirements of the SAM architecture. The prompts for the images were provided as bounding boxes defined by the minimum and maximum coordinates of a box that encapsulated a crack. This was identified from binary masks generated using the JSON file with the annotations for the cracks. Since SAM can only take a single prompt at a time, the training dataset was prepared as a set of triplets where each one constituted an image, its true mask, and a bounding box prompt for a single crack. So, if an image contained multiple (n) cracks, there were as many triplets with a corresponding bounding box to account for each crack ([image, mask, $bbox_1$], [image, mask, $bbox_2$], … [image, mask, $bbox_n$]).

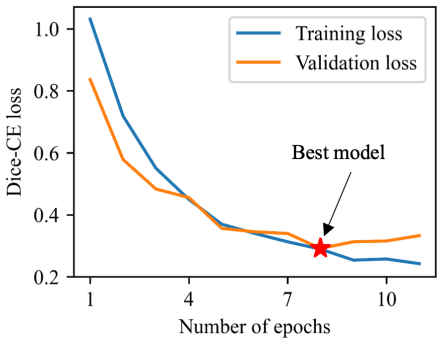

Figure 6. Loss curves observed during fine-tuning of SAM showing the evolution of training and validation losses. The model was fine-tuned for 11 epochs before it terminated due to the early stopping criteria. The best model was selected based on the lowest losses (indicated by the red star).

To limit the resources for training and to minimize overfitting the parameters on the relatively small dataset, the weights of the prompt encoder were frozen. The vision encoder and mask decoder were fine-tuned to the training dataset. Adam optimizer with a learning rate of $1x10^{-3}$ and a weight decay of $1x10^{-4}$ was selected. The loss function was the sum of cross entropy and Dice loss. An early stopping criterion was also applied to prevent overtraining and overfitting. This criterion ensured the

model stopped training if the validation loss did not improve after 3 consecutive epochs. Figure 6 shows the loss curves where the model was fine-tuned for 11 epochs on a single GPU node until it hit an early stop. The fine-tuned SAM with the best weights from training was evaluated without specific positional inputs i.e. no prompts were provided for predicting the validation data. This made it equivalent to the pre-trained SAM in its segment everything mode. The fine-tuned SAM achieved a mean IoU and Dice coefficient of 0.498 and 0.663 on the validation dataset. Figure 7 shows the cracks predicted by this fine-tuned SAM as a probability map for a representative image from the validation dataset. Here, cracks correspond to a probability of 1, and 0 refers to the background. While the fine-tuned model was an improvement over its pre-trained version, the model suffered from issues such as over-segmentation, i.e. segmenting too many pixels around a crack making it wider or longer, of cracks (boxes 1 and 2), missing instances (box 3), and incorrect segmentation of pores in Nb_3Sn (box 4).

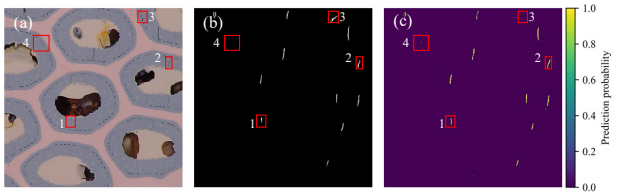

Figure 7. Segmentation by fine-tuned SAM with the validation image in (a), true binary mask identifying cracks in (b), and the predicted mask as a probability map in (c).

Although the model accuracy could improve with more training data or data augmentation, a major deterrent to model performance was the quality of prompts. Since the model can only take a single prompt at a time, the selected box would either include a single crack or encompass all cracks, capturing a significant portion of the image instead of narrowing into the regions of interest. In this case, the former method that takes a single crack per bounding box was used for fine-tuning as it was more stable and yielded better performance. However, this approach created a virtual increase in the dataset by including multiple instances of the same image (and mask) to include individual cracks using bounding boxes. This means for a given input; the model was provided different prompts to learn the same true mask containing all the cracks. While this provided the instance-level isolation required to make SAM delineate the cracks instead of segmenting everything, the actual diversity of the dataset was limited and hence, the model failed to generalize, making it prone to overfitting. Moreover, both the pre-trained and fine-tuned models were made to segment everything without manual prompts to localize the region of interest. SAM's performance, thus, evidently hinges on the precision of prompts, making user interaction necessary for accurate crack detection. Consequently, SAM shows promise for image annotation as an interactive segmentation tool. By using a pre-trained SAM backend, users can easily draw bounding boxes to obtain a predicted mask around the crack [30], [31]. This significantly accelerates the labeling process compared to hand-drawing the outlines to generate datasets for training more robust instance segmentation models.

IV. Crack Distribution in Wires and in Stacks

To perform quantitative analyses on the segmented cracks in individual wires and in cable stacks, some properties were computed from the binary masks generated by the Mask R-CNN model and discussed in Section III-A. Using the Python-based library scikit-image, the number of distinct cracks in each wire and the centroid, the length, and the orientation of each crack were measured and stored in *.csv* files.

Figure 8a shows the distribution of cracks in wires as 2D histograms of the centroid of each crack grouped by cable number in a stack and for different loading conditions listed in Table 1. Darker regions indicate a higher crack occurrence. As previously reported for uniaxial compression of stacks [10], [11], [12] and from finite element modeling [10] cracks are mostly located along ~45° shear bands, with respect to the orientation with the largest applied load. These observations remain valid for samples also loaded along the *x* and *z* axes by 20 MPa. The more annular distribution of cracks for Cable 1 of Sample 3 in Figure 8a is likely a consequence of loading and polishing artefacts at the edges of the stack, for the wires highlighted in red in Figure 9 and not a consequence of the additional load of 20 MPa along the *x*-axis.

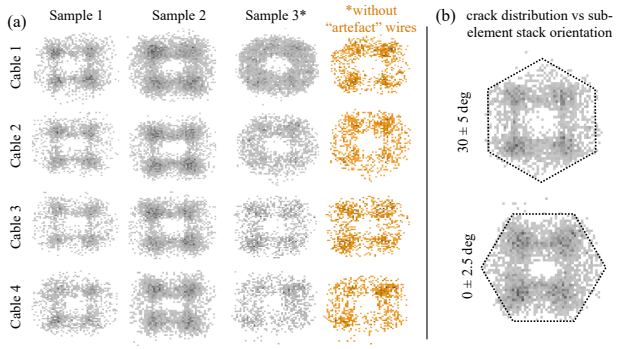

Figure 8. Crack distribution in wires grouped by (a) cable position in a 4-stack and sample loading condition and (b) subelement stack hexagonal orientation.

To study the effect of crack distribution in wires as a function of the subelement hexagonal stack orientation, this orientation was estimated using traditional image analysis. First, the subelements were segmented using automatic thresholding with the multi-Otsu method on grayscale images. Second, the convex Hull of the subelements was computed to fill empty spaces between subelements. Third, the distance between the centroid of the convex Hull and all points at its edge was calculated. The edge location with the largest distance was defined as an apex of the hexagon. Finally, the angle between a horizontal line and that apex was measured and converted to the 0° to 30° range. This angle was used to define the orientation of the subelement stack in each wire as shown with dashed lines in Figure 8b. Note that an arguably more robust method was defined by A. Baskys et al. [32] in a study of sheared subelements at the triplet of Rutherford cable edges. However, the method described above was deemed sufficiently accurate as most of the wires in the current study are along the broad faces and therefore less deformed than in [32]. Figure 8b shows 2D histograms of the centroid of cracks in all wires with subelement hexagonal stacks at two different orientations: 30°

± 5° and 0° ± 2.5°. As shown in Figure 16 of G. Lenoir et al. [11] for uniaxial loading and observed in the current study for different loading conditions, the hexagonal stack orientation is not significantly affecting the distribution of cracks in wires. Cracks are still mostly located along ~45° shear bands, with respect to the orientation with the largest applied load.

The centroid of each wire in the 4-stack was computed using traditional image analysis on the lower resolution images of the stacks to perform stack-based analyses of cracks. Figure 9 shows the distribution of the number of cracks in each wire for the three samples. Larger and darker circles indicate that more cracks were counted in a given wire. Qualitatively, Sample 2 has more cracks than the other two samples and they are concentrated around the center of the stack, where no edge loading artefacts are expected. Sample 3 has a high concentration of cracks in the top corners of the stack (highlighted in red) that is likely caused by a combination of loading and polishing artefacts. The latter could be explained by the sample being slightly off-centered in the 1.25" epoxy puck used for metallographic preparation and to doming of the polished surface from compliance of the polishing pad media.

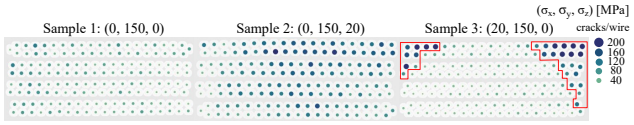

Figure 9. Crack distribution in the sample stacks with markers of different size and color based on the number of cracks found in each wire. Regions with artefacts are highlighted in red for Sample 3 and discussed in the text.

Figure 10a, b, and c show crack statistics grouped by loading conditions listed in Table 1 and Table 2 summarizes the median and distribution of each metric. The data from wires highlighted in red in Sample 3, where a large amount of cracks was found, is not used in these plots. As previously observed [9] and shown with boxplots in Figure 10a, the addition of a compressive stress along the *x*-axis (Sample 3) or along the *z*-axis (Sample 2) reduces and increases the number of cracks per wire, respectively. Values of the median, 1st, and 3rd quartiles (Q1 and Q3) of the number of cracks per wire are reported in Table 2.

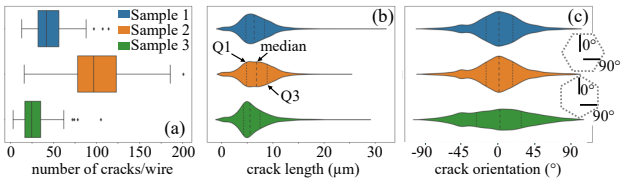

Figure 10. Crack statistics for the three samples for loading conditions listed in Table 1. (a) box plots of the number of cracks per wire and probability distribution of (b) crack length and (c) crack orientation.

Compared to previous studies where cracks were manually counted [9], [10], [11], [12], the binary mask of each crack allows for the computation of morphological descriptors, such as crack length, orientation, eccentricity, solidity, and more. Figure 10b shows probability distribution as violin plots of the crack length for all wires in each sample. Inner dashed and solid lines represent the median and Q1 and Q3 of the crack length distribution, respectively, and are listed in Table 2. Slightly shorter cracks were measured in Sample 3, supporting the observation that additional pre-loading along the *x*-axis limits crack propagation in the brittle A15 phase.

Table 2. Summary of computed crack statistics with median and interquartile ranges for skewed distributions (Q1–Q3) or standard deviation for near-symmetric distributions (± σ).

Sample number	cracks/wire (-)	crack length (μm)	crack orientation (°)
1	42 (32–56)	6.3 (4.8–8.2)	0.9 (± 27.7)
2	97 (78–123)	6.8 (5.0–8.8)	0.8 (± 28.6)
3	25 (17–35)	5.8 (4.3–8.0)	1.9 (± 37.3)

Figure 10c shows violin plots of the crack orientation for all wires in each sample. As visually observed and expected for samples where the highest load was applied along the *y*-axis, most of the cracks propagated along the *y*-axis, corresponding to a crack angle of 0°. A wider distribution of crack orientation was observed for the sample that was loaded along the *x*-axis by 20 MPa, see the standard deviation of Sample 3 in Table 2. This suggests that in addition to preventing cracks to open along the *y*-axis and reducing the total number of cracks found in stacks, the resultant stress-state due to pre-loading along the *x*-axis is also affecting the direction of crack propagation. More observations will be conducted in a future study to further investigate the results reported based on Figure 10b and c.

V. Conclusion

Instance segmentation of cracks in three Nb_3Sn 4-stack samples deformed under uniaxial and biaxial loading conditions was done using the Mask R-CNN model in the Detectron2 framework. The model was trained on a single GPU node using 43 images containing 332 cracks and pre-initialized weights. Bounding box and mask average precisions on the validation dataset of 42.8 and 27.9, respectively, were obtained after 1,000 iterations. Crack statistics were computed after instance segmentation and compared for the different loading conditions and as a function of position in stacks and in wires. As previously reported, cracks were mostly concentrated along ~45° shear bands, with respect to the direction with the highest applied load, for all loading conditions and different subelement stack hexagonal orientations. The statistics on entire stacks, except for the artefact wires (Fig. 9), confirm observations reported on a handful of wires for the same samples [9] that pre-loading by 20 MPa reduces (pre-loading along the *x*-axis) or increases (pre-loading along the *z*-axis) the number of cracks found.

A similar study was also attempted using SAM. Pre-trained SAM in its "segment-everything" mode could not delineate the cracks. After fine-tuning, it was able to segment and identify most cracks with modest accuracy. However, the model still showed limited performance without manual prompts. Moreover, SAM in itself cannot classify or label objects. While this was not a problem for the binary problem of identifying cracks, more complex analysis requiring the detection of multiple defects warrants the use of Mask R-CNN. SAM serves better as an annotation tool allowing for interactive segmentation, thereby simplifying the data generation process.

References

[1] N. Cheggour, T. C. Stauffer, W. Starch, L. F. Goodrich, and J. D. Splett, "Implications of the strain irreversibility cliff on the fabrication of particle-accelerator magnets made of restacked-rod-process Nb_3Sn wires," *Sci. Rep.*, vol. 9, no. 1, p. 5466, Apr. 2019, doi: 10.1038/s41598-019-41817-7.

[2] P. Ferracin *et al.*, "The HL-LHC Low-β Quadrupole Magnet MQXF: From Short Models to Long Prototypes," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, pp. 1–9, Aug. 2019, doi: 10.1109/TASC.2019.2895908.

[3] G. Apollinari, O. Brüning, T. Nakamoto, and L. Rossi, "High Luminosity Large Hadron Collider HL-LHC," 2015, *CERN*. doi: 10.5170/CERN-2015-005.1.

[4] A. Ballarino and L. Bottura, "Targets for R&D on Nb3Sn Conductor for High Energy Physics," *IEEE Trans. Appl. Supercond.*, vol. 25, no. 3, pp. 1–6, Jun. 2015, doi: 10.1109/TASC.2015.2390149.

[5] B. Bordini, P. Alknes, L. Bottura, L. Rossi, and D. Valentinis, "An exponential scaling law for the strain dependence of the Nb3Sn critical current density," *Supercond. Sci. Technol.*, vol. 26, no. 7, p. 075014, May 2013, doi: 10.1088/0953-2048/26/7/075014.

[6] G. Vallone, B. Bordini, and P. Ferracin, "Computation of the Reversible Critical Current Degradation in Nb_3Sn Rutherford Cables for Particle Accelerator Magnets," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 4, pp. 1–6, Jun. 2018, doi: 10.1109/TASC.2018.2810222.

[7] G. Vallone, E. Anderssen, B. Bordini, P. Ferracin, J. F. Troitino, and S. Prestemon, "A methodology to compute the critical current limit in Nb_3Sn magnets," *Supercond. Sci. Technol.*, vol. 34, no. 2, p. 025002, Dec. 2020, doi: 10.1088/1361-6668/abc56b.

[8] G. Vallone *et al.*, "Computation of the Strain Induced Critical Current Reduction in the 16 T Nb_3Sn Test Facility Dipole," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, pp. 1–5, Aug. 2023, doi: 10.1109/TASC.2023.3247690.

[9] G. Vallone *et al.*, "Measurement and Computation of Nb_3Sn Rutherford Cables Strength Under Multi-Axial Loading Conditions," *IEEE Trans. Appl. Supercond.*, vol. 34, no. 5, pp. 1–5, Aug. 2024, doi: 10.1109/TASC.2023.3340126.

[10] P. Ebermann *et al.*, "Influence of transverse stress exerted at room temperature on the superconducting properties of Nb_3Sn wires," *Supercond. Sci. Technol.*, vol. 32, no. 9, p. 095010, Sep. 2019, doi: 10.1088/1361-6668/ab2e51.

[11] G. Lenoir, K. Puthran, C. Barth, J. Fleiter, and A. Ballarino, "Effect of transverse compressive stress applied at room temperature on Nb_3Sn Rutherford cables," *Supercond. Sci. Technol.*, vol. 37, no. 2, p. 025013, Feb. 2024, doi: 10.1088/1361-6668/ad1341.

[12] K. Puthran, C. Barth, A. Ballarino, A. Devred, and T. Arndt, "Onset of Mechanical Degradation due to Transverse Compressive Stress in Nb_3Sn Rutherford-Type Cables," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, pp. 1–6, Aug. 2023, doi: 10.1109/TASC.2023.3241568.

[13] A. Moros *et al.*, "A Metallurgical Inspection Method to Assess the Damage in Performance-Limiting Nb_3Sn Accelerator Magnet Coils," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, pp. 1–8, Aug. 2023, doi: 10.1109/TASC.2023.3237662.

[14] C. Sanabria *et al.*, "Metallographic autopsies of full-scale ITER prototype cable-in-conduit conductors after full testing in SULTAN: 1. The mechanical role of copper strands in a CICC," *Supercond. Sci. Technol.*, vol. 28, no. 8, p. 085005, Aug. 2015, doi: 10.1088/0953-2048/28/8/085005.

[15] S. Balachandran *et al.*, "Metallographic analysis of 11 T dipole coils for High Luminosity-Large Hadron Collider (HL-LHC)," *Supercond. Sci. Technol.*, vol. 34, no. 2, p. 025001, Feb. 2021, doi: 10.1088/1361-6668/abc56a.

[16] P. Ebermann *et al.*, "Irreversible degradation of Nb3Sn Rutherford cables due to transverse compressive stress at room temperature," *Supercond. Sci. Technol.*, vol. 31, no. 6, p. 065009, May 2018, doi: 10.1088/1361-6668/aab5fa.

[17] A. Rani, D. Ortiz-Arroyo, and P. Durdevic, "Defect Detection in Synthetic Fibre Ropes using Detectron2 Framework," *Appl. Ocean Res.*, vol. 150, p. 104109, Sep. 2024, doi: 10.1016/j.apor.2024.104109.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," Jan. 24, 2018, *arXiv*: arXiv:1703.06870. Accessed: Aug. 25, 2024. [Online]. Available: http://arxiv.org/abs/1703.06870

[19] A. Kirillov *et al.*, "Segment Anything," Apr. 05, 2023, *arXiv*: arXiv:2304.02643. Accessed: Aug. 25, 2024. [Online]. Available: http://arxiv.org/abs/2304.02643

[20] P. Skalski, "Make Sense." 2019. [Online]. Available: https://github.com/SkalskiP/make-sense/

[21] "COCO - Common Objects in Context." Accessed: Sep. 10, 2024. [Online]. Available: https://cocodataset.org/#format-data

[22] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[23] H. Wen, C. Huang, and S. Guo, "The Application of Convolutional Neural Networks (CNNs) to Recognize Defects in 3D-Printed Parts," *Materials*, vol. 14, no. 10, p. 2575, May 2021, doi: 10.3390/ma14102575.

[24] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nat. Commun.*, vol. 15, no. 1, p. 654, Jan. 2024, doi: 10.1038/s41467-024-44824-z.

[25] L. P. Osco *et al.*, "The Segment Anything Model (SAM) for remote sensing applications: From zero to one shot," *Int. J. Appl. Earth Obs. Geoinformation*, vol. 124, p. 103540, Nov. 2023, doi: 10.1016/j.jag.2023.103540.

[26] "detectron2/MODEL_ZOO.md at main · facebookresearch/detectron2 · GitHub." Accessed: Sep. 16, 2024. [Online]. Available: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md

[27] "COCO - Common Objects in Context." Accessed: Sep. 23, 2024. [Online]. Available: https://cocodataset.org/#detection-eval

[28] P.-S. Liao, T.-S. Chen, and P.-C. Chung, "A Fast Algorithm for Multilevel Thresholding," *J. Inf. Sci. Eng.*,

vol. 17, no. 5, pp. 713–727, Sep. 2001, doi: 10.6688/JISE.2001.17.5.1.

[29] S. van der Walt *et al.*, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, Jun. 2014, doi: 10.7717/peerj.453.

[30] CVAT. ai Corporation, *Computer Vision Annotation Tool (CVAT)*. (Jul. 18, 2024). Zenodo. doi: 10.5281/ZENODO.12771595.

[31] S. Bhattiprolu, *DigitalSreeni Image Annotator*. (2024). Python. Accessed: Sep. 23, 2024. [Online]. Available: https://github.com/bnsreenu/digitalsreeni-image-annotator

[32] A. Baskys *et al.*, "RRP Nb_3Sn Subelement Shear Dependence on Hexagonal Subelement Stack Orientation and the Strand's Position Within a Rutherford Cable," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, pp. 1–5, Aug. 2023, doi: 10.1109/TASC.2023.3265910.