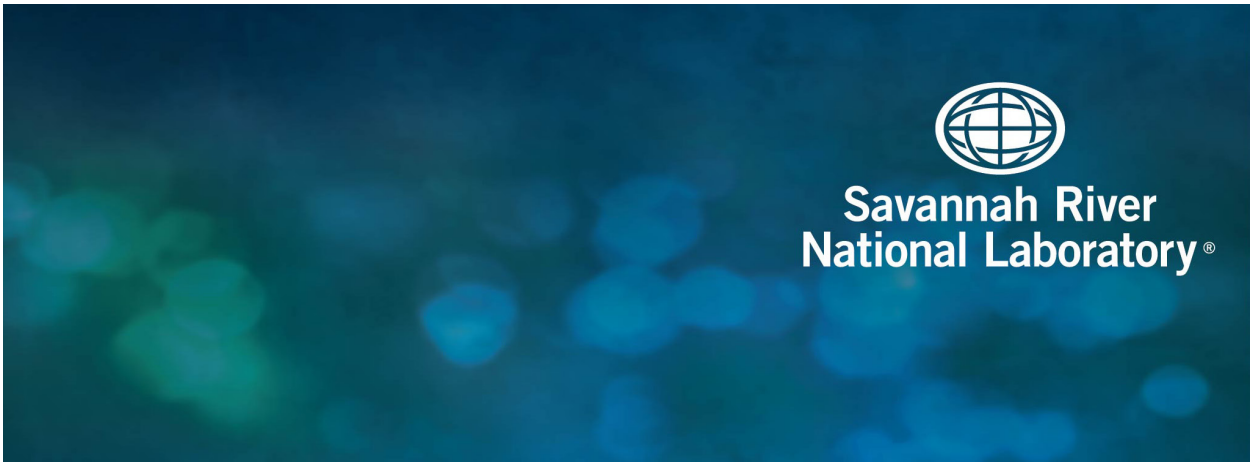
Contract No:

This document was prepared in conjunction with work accomplished under Contract No. 89303321CEM000080 with the U.S. Department of Energy (DOE) Office of Environmental Management (EM).

Disclaimer:



A U.S. DEPARTMENT OF ENERGY NATIONAL LAB • SAVANNAH RIVER SITE • AIKEN, SC • USA

FY21 Progress Report: SRNL Analysis of ICCWR LCM and WAMS data for Corrosion and Cracking

Bruce Hardy (SRNL)
Anna d'Entremont (SRNL)
Jason Bakos (University of South Carolina)
Taylor Clingenpeel (University of South Carolina)
Michael Martínez-Rodríguez (SRNL)
Roderick Fuentes (SRNL)

September 30, 2021
SRNL-STI-2021-00680, Revision 0

DISCLAIMER

Printed in the United States of America

Prepared for
U.S. Department of Energy

Keywords: *3013 Corrosion, 3013 Cracking, 3013 Surveillance Program, Machine Learning 3013 Corrosion*

Retention: *Varies*

FY21 Progress Report: SRNL Analysis of ICCWR LCM and WAMS data for Corrosion and Cracking

Bruce Hardy (SRNL)

Anna d'Entremont (SRNL)

Jason Bakos (University of South Carolina)

Taylor Clingenpeel (University of South Carolina)

Michael Martínez-Rodríguez (SRNL)

Roderick Fuentes (SRNL)

September 30, 2021

Savannah River National Laboratory is operated by Battelle Savannah River Alliance for the U.S. Department of Energy under Contract No. 89303321CEM000080.

REVIEWS AND APPROVALS

AUTHORS:

Bruce Hardy, SRNL	December 16, 2021
Anna d'Entremont, SRNL	January 3, 2022
Jason Bakos, University of South Carolina	January 7, 2022
Taylor Clingenpeel, University of South Carolina	January 7, 2022
Michael Martínez-Rodríguez, SRNL	January 10, 2021
Roderick Fuentes, SRNL	January 7, 2022

TECHNICAL REVIEW:

Carina Grady, SRNL	January 10, 2022

APPROVAL:

Patricia Lee, Manager SRNL/Advanced Modeling, Simulation and Analytics	January 11, 2021
Marissa Reigel, Manager SRNL/Actinide Materials Science and Technology	January 11, 2021

EXECUTIVE SUMMARY

The development of algorithms for machine learning and data analysis for the 3013 Surveillance Program is a collaborative effort by the Savannah River National Laboratory (SRNL) and the University of South Carolina (USC). For corrosion detection, Laser Confocal Microscope (LCM) or Wide Area 3D Measurement System (WAMS) data is extracted from large binary files, with software written to convert the data to physical attributes (e.g., height, color and grayscale values; all as functions of a location in a plane projection). A user-friendly Matlab Graphical User Interface (GUI) that reads data from either LCM or WAMS files was developed to integrate input data with software developed for processing and evaluation. The GUI can selectively download binary data, interrogate data attributes, label data, flag significant features, execute Machine Learning (ML) algorithms, output parameters for trained ML algorithms, report ML model accuracy with respect to labeled data, and generate graphical representations for various analyses. Features can be called out by user-specified thresholds, manual labeling or machine learning algorithms when they have been completed. The ability to rapidly label data is important because of the volume of data required for training machine learning algorithms.

The GUI has the flexibility to allow addition of improved ML algorithms, methods for data visualization, and statistical computations. Statistical analyses via the GUI include areas of pits within a defined range of pit depths, correlations between Red-Green-Blue (RGB) or grayscale intensity and relative surface height, covariances between values associated with features, and feature histograms.

The development of supervised machine learning algorithms, however, has been hindered by a lack of training data. The machine learning algorithms for crack identification are being refined but require improvements to the true positive rate for crack detection. This shortcoming is an artifact of the limited training data currently available, perhaps more so than the structure of the neural networks. At present, the best results are had from a consensus over an ensemble of randomly generated Deep Neural Network (DNN) or Convolutional Neural Network (CNN) algorithms. Although the consensus accuracy method has yielded optimum true positive and true negative rates in excess of 80%, additional validation testing is necessary.

In addition to the suite of LCM data that was initially used, and which represents the majority of the work presented in this report, WAMS image data was also reviewed at a preliminary level. The review included a comparison between image resolution and dynamic range for each method. WAMS (ZON file) image data was found to have a pixel pitch of 3.69μm compared to 1 μm for the LCM (vk4 file) data, which implies a lower resolution for the WAMS images. Conversely, the ratio of dynamic range of the WAMS data to the LCM data was approximately 41:20 for height data, suggesting that information from WAMS should more accurately determine the depth of pits. At present, the significance of the greater dynamic range of the WAMS data relative to the LCM data has not yet been evaluated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CAE	Convolutional Autoencoder
CNN	Convolutional Neural Network
CPR	Crack Positive Rate
CS	Computer Science
DNN	Deep Neural Network
DOE	Department of Energy
GUI	Graphical User Interface
ICCWR	Inner Container Closure Weld Region
LANL	Los Alamos National Laboratory
LCM	Laser Confocal Microscope
ML	Machine Learning
NN	Neural Network
RELU	Rectified Linear Unit
RGB	Red, Green, Blue Colors
SCC	Stress Corrosion Cracking
SRNL	Savannah River National Laboratory
TNR	True Negative Rate
TPR	True Positive Rate
USC	University of South Carolina, Columbia, SC
WAMS	Wide Area 3D Measurement System

1.0 Introduction

Through-wall penetration from stress corrosion cracking (SCC) of the 3013 inner container has been identified as the most credible condition for failure within the 50-years lifetime[1]. Chlorides contained in Pu-bearing material, together with intra-canister humidity levels, metallurgical conditions, and internal stresses have been found to produce corrosion in the Inner Container Closure Weld Region (ICCWR) of the 3013 canister system, which is used throughout the DOE complex. A Laser Confocal Microscope (LCM) is used as part of the 3013 Surveillance Program protocol to identify the prevalence of corrosion and corrosion-related cracking in the ICCWR[2]. With the LCM, a close visual examination is made of the ICCWR surface along with measurements of corrosion-related features[2]. LCM inspections produce immense amounts of image data: approximately 6000 LCM images per can, having 786,432 pixels per image, with 8 layers of data for each pixel. There is currently an 8-year backlog of images, with approximately 45 canisters that must be evaluated. Simplistic computer-aided image analysis can flag some basic surface characteristics, such as pit depth, to guide manual examinations for corrosion. However, while this approach greatly improves the efficiency of the examination process compared to unaided manual screening, it is still excessively time consuming. A more efficient and sophisticated approach is to assess the data using Machine Learning (ML) algorithms to identify corrosion without manual intervention.

In FY21 the development of ML algorithms and the user interface was continued from the work done in FY20[3]. As before, the objective of this project is the development of computer-facilitated methods to facilitate rapid identification of corrosion and corrosion-related damage in 3013 canisters from very large sets of LCM (vk4 files) and Wide Area 3D Measurement System (WAMS) data (ZON files). The LCM data includes: RGB, RGB + laser intensity, laser intensity and height data. Data from WAMS files includes height and RGB data.

Computational methods for identifying surface corrosion and cracking include user-specified thresholds for flagging, covariance, morphological filtering, and machine learning. While all of these methods were implemented at some stage of development for this project, the latter, machine learning, was a major focus area due to its potential for rapid interpretation of surface data and direct incorporation into statistical methods. In this study, the development of algorithms for machine learning and data analysis was a collaborative effort by the Savannah River National Laboratory (SRNL) and the University of South Carolina, Columbia (USC).

The process for detection of corrosion and cracking in the 3013 ICCWR consists of first extracting surface data from large binary files. This data is processed using software that was written to convert it into physical attributes. Data taken by LCM and WAMS measurements consisted of height, color and grayscale values, all as functions of a location in a plane projection. To facilitate this operation, a user-friendly Graphical User Interface (GUI) was developed to selectively download binary data and interrogate its attributes. The GUI is a complete package that:

- Reads binary data files to enable viewing and processing of both height and optical data.
- Stitches individual LCM images into a mosaic with matched edges to give a panoramic view of the surface.
    - The image “view” can include height as well as optical data.
- Zooms and rotates images and 3D height data for local and regional examination.
- Performs tilt/curvature correction to flatten the image so that local height variations are not obscured by the canister wall curvature or sample tilt.
- Allows height thresholds, input by the user, to automatically flag surface features of interest, especially pits and other surface irregularities.

- Permits an expert to label features for training ML algorithms.
- Computes histograms that summarize the distribution of features.
- Incorporates and executes ML algorithms.
    - The ML algorithms are implemented as a module so that they can readily be replaced as they are improved or exchanged with algorithms better suited to a particular data set or application.
    - The GUI facilitates training and testing of ML algorithms.
- Can apply and graphically display statistical operations to surface features.

Currently, the ML algorithms are being refined to more reliably identify corrosion and cracking, see Section 5. Figure 1 shows a flowchart for the overall development of the methodology, as applied to LCM data.

2.0 FY21 Objectives

The FY21 objectives for this project were:

- Development of data analysis, data modeling, and machine learning algorithms for large data evaluation by USC/SRNL.
    - Improvement or upgrades of the image processing software used to identify and quantify surface features from WAMS or LCM data
    - Use of labeled training and test data for 3013 corrosion features to explore the selection and implementation of appropriate machine learning algorithms.
- Collaboration with LANL as needed to enable data collection and compare methods for corrosion and crack identification.

3.0 Approach

Corrosion is strongly, but not exclusively, associated with surface pitting and cracking, coloration, along with shapes and patterns of surface features, see Figure 2. Conversely, not all pits and surface lesions are the result of corrosion: some are artifacts of fabrication, impact, scoring or other non-corrosion events. Corrosion is identified via the combined properties of pit depth, area, edge contour, color and clustering. Software was developed to extract these features from large binary files generated by the LCM, and analogously will apply to images from WAMS data. The individual LCM images, which collectively span the ICCWR, were stitched together and corrected to eliminate the effect of the canister surface curvature on measurement of the local height. Various image processing methods were tested for identifying the presence of cracks and corrosion. The methods included Deep Neural Networks (DNNs), Convolutional Neural Networks (CNN's), gradient methods, statistical characterization, correlations, and filters[4,5,6]. The process is shown schematically in Figure 1.

Samples of LCM image data taken for the 3013 Surveillance Program containing identified cracks, pits and other features characteristic of corrosion were used as training data for ML algorithms. To provide an efficient means for handling large amounts of binary image data, a GUI was developed to serve as an interface with the data files, manipulate and group images, label features for training the ML algorithm, group features with user-defined thresholds, correct for sample tilt and curvature, stitch images, train ML algorithms, and apply the algorithms for crack and corrosion identification. Further, methods were developed to read binary WAMS data, which has recently been adopted for 3013 image interrogation. After preprocessing, images obtained from LCM and WAMS data were partitioned into tiles (rectangular blocks of pixels). Image data used for training and testing ML algorithms was labeled and classified on a per-tile basis. Studies conducted during FY20 emphasized that larger views, represented by image tiles containing a larger number of pixels, improve the accuracy of crack detection by the ML algorithms. The low

incidence of corrosion and cracking in the actual ICCWR samples resulted in an imbalanced data set with a much larger fraction of tiles without cracks than tiles containing cracks, making it necessary to incorporate data augmentation schemes for effective training of the ML algorithms developed for this application. Data augmentation consisted of generating vertical and horizontal translations of the original tiles labeled as containing cracks (Figure 3), thus multiplying the number of crack tiles in the training set. Capability for labeled image rotation was also developed (Figure 4). Augmenting via translation (or rotation) ensures that the cracks in the new training tiles remain realistic examples and encourages robustness of the algorithm relative to the position (or orientation) of the crack in the tile.

4.0 Results/Discussion

Cracks, pits and color patterns are all associated in various forms with corrosion. Pits can readily be detected using height data thresholds or have their edges defined from optical data through a band pass filter that incorporates a discrete Fourier transform. However, cracks, particularly "hairline" cracks, do not always have a definitive height signature. Rather, crack identification is a combination of grayscale image intensity (pixel value) and height data. Initially, it was hoped that standard edge detection methods could be used with pixel values to extract crack edges. Methods considered included: erosion and dilation, blurring, Fourier and Gaussian filters, and gradient methods. Unfortunately, other surface features combined to create background noise that was similar in frequency to that associated with crack edges.

To overcome the problems with using feature morphology directly, crack identification was attempted through the use of ML methods. Generally, the training of the ML algorithms suffered due to the small amount of crack data available and the severe imbalance between the amount of crack data and the much larger amount of non-crack surface (this imbalance tends to result in algorithms that identify every tile as non-cracks, resulting in high overall accuracy but no useful predictive capability). This was particularly true in the early stages of algorithm development. The low fraction of tiles containing cracks was offset somewhat by augmenting the crack tiles via vertical and horizontal translation to synthesize additional labeled crack data.

Studies conducted during FY20 indicated that larger views, represented by image tiles containing a larger number of pixels, improved the accuracy of crack detection by Deep Neural Networks. In FY20, it was found that increasing the amount of labeled crack training data through augmentation, adjusting the DNN algorithms, and increasing the image tile size from 64x64 to 112x112 pixels improved the precision and recall for crack identification. Examples of labeled training data for cracks, taken from LCM images, are shown in Figure 5. In FY21, the maximum tile size was further increased to 128x128 pixels to provide additional context for feature elucidation when using the improved ML methods that were developed. It was found that the greatest accuracy was obtained by using a consensus drawn from an ensemble of CNN or DNN algorithms, with randomly generated hyperparameters.

The consensus is a vote by all models in the ensemble on whether or not an image contained a crack. The accuracy of the ensemble is measured in terms of the Crack Positive Rate (CPR) and the True Negative Rate (TNR). The maximum consensus threshold value that achieves a CPR of 100% is selected for use in application of the ensemble of neural networks. This allows the ensemble to maximize its TNR while still detecting at least one tile of each of the distinct cracks. (see Figures 6 and 7). Another parameter, the True Positive Rate (TPR) is used in the selection of individual neural networks within the ensemble, as described below. These parameters are defined as

$$CPR \equiv Crack\ Positive\ Rate = \frac{\sum_{i=1}^{N_{Cracks}} \begin{cases} 1\ if\ TP_i > 0 \\ 0\ otherwise \end{cases}}{N_{Cracks}}$$

$TNR \equiv$ *True negative rate*

$=$ *(number of tiles correctly classified as non-crack containing)/(actual number of non-crack containing tiles)*

$\equiv \frac{TN}{TN+FP}$

and

$TPR \equiv$ *True positive rate*

$=$ *(number of tiles correctly classified as containing a crack)/(actual number of tiles containing a crack)*

$\equiv \frac{TP}{TP+FN}$

where: TP_i = Number of tiles correctly classified as containing the i[th] distinct crack

N_{Cracks} = Total number of labeled distinct cracks

TN = Number of tiles correctly classified as *not* containing a crack

FP = Number of tiles incorrectly classified as containing a crack

TP = Number of tiles correctly classified as containing a crack

FN = Number of tiles incorrectly classified as *not* containing a crack

The method is summarized as:

- Randomly generate an ensemble of DNNs, CNNs or a combination of both.
  - Each DNN/CNN in the ensemble has randomly generated hyperparameters:
    - Number of layers, number of maps (e.g., filters) per layer, receptive field (kernel) size, kernel size in the various layers, stride length for pooling layers, architecture of convolution, fully connected and pooling layers, pooling kernel size, etc.
  - Each individual added network, after training, must meet a threshold for its TPR and TNR for its training data or it is discarded, and a new network is randomly generated
- Crack containing tiles are classified by consensus over the ensemble of DNN's
  - The consensus uses a vote threshold, i.e., a minimum percentage of ensemble classifiers that must vote that a tile contains a crack in order for the whole ensemble to classify the tile as containing a crack. Figures 6-7 show the accuracies of the ensemble predictions as a function of the vote threshold ranging from 10-100% (horizontal axis). Higher vote threshold requires greater consensus among models to classify a tile as a crack, so the crack positive rate decreases (more false negatives) and the true negative rate increases (fewer false positives) with increasing vote threshold.
  - This approach reduces the error to be less than or equal to that for a single DNN in the ensemble, and much more if the individual DNN errors are uncorrelated (Ref. Goodfellow, Bengio, Courville *Deep Learning*, Sect 7.11)
- Metrics for consensus classification are the *crack positive rate* and the *true negative rate*
  - Crack positive rate
    - Since human screeners identify entire features that may span multiple tiles rather than classifying on a per-tile basis, sets of contiguous crack tiles were identified as distinct labeled cracks, on the grounds that an algorithm that identifies *at least one* tile in a given crack will successfully flag the feature for human review. Assessing how algorithms perform on this

basis helps to compensate for labeling uncertainties where an individual tile along a crack may be ambiguous when viewed out-of-context.

  - The maximum consensus threshold value that achieves a CPR of 100% is selected. This allows the ensemble to maximize its TNR while still detecting at least one tile of each of the distinct cracks (see Figures 6 and 7).
- Tile size, which is based on the number of pixels in an image, affects the training and validation accuracy (see figures below)
  - Too large a tile can result in overfitting (attributable to the limited training set; dividing the available images into larger tiles reduces the number of unique tiles seen during training)
  - Too small a tile can result in lack of context and significant confusion between crack and non-crack images
  - Figures 6 and 7 below show the crack positive and true negative rates for different tile sizes, neural network types and consensus thresholds
  - The accuracy obtained by the consensus method is high and comparable to the better results for image identification presented in the literature.

In comparison to the ML models, a sample of distributions of features based on morphology and color for crack versus non-crack tiles are shown in Figures 8 - 11. These features are the inputs to the DNNs, which were able to achieve reasonably high accuracies by analyzing them using ML (Figures 6-7). As seen in the figures, there is no *obvious* distinction between crack and non-crack tiles based on the geometric features that would allow for discrimination between them based on any individual feature. The use of the ML techniques was necessary to identify the more complex patterns among numerous features to distinguish between crack and non-crack tiles.

5.0 FY21 Accomplishments

The FY21 accomplishments in chronological order are:

Oct. 2020:

- Expanded training dataset from 4 to 7 labeled images (where each image consists of 2 or more stitched LCM images overlapping a crack), through the gradual creation of a ledger detailing the datasets received thus far.
- Developed new approach for building ensembles, in which randomly trained models are only added if min(TPR,TNR)>threshold.
- Built an index for all of the LCM data, mapping each dataset to a description of its annotated features.
- Improved processing WAMS files accounting for ‘out of bounds’ pixels.
- Provided LANL with the software required to read the WAMS data.

Nov. 2020:

- Implemented WAMS file reading into the vk4 analysis application.
- Reformulated the vk4 analysis into a two-application system for Labeling and Training (named Analyze&Label and Train&Validate).
- Analyze&Label was written to accept arbitrary file types.

Dec. 2020:

- Parallelized code using Matlab Parallel Computing Toolbox.
- Designed MATLAB save file format for transferring labeled datasets and opening and closing applications.

Jan. 2021:

- Expanded training dataset from 7 to 10 labeled images (each ≥2 stitched LCM images containing at least one crack).
- Analyze&Label and vk4 analysis were given user-friendly additions like click and drag support, bug fixes, and improved file processing.
- Redesigned Analyze&Label's tab system to work more seamlessly and provided added tabs.

Feb 2021:

- Developed a new accuracy metric: the crack positive rate, which is built from a method by which contiguous crack (positive) tiles are grouped into interconnected regions.
- Developed new ensemble algorithm: consensus threshold, which varies the number of ensemble votes needed to classify a tile as positive.
- Solidified the Train&Validate application for generating neural networks.
- Created a save file format for Train&Validate to save previously generated networks and load them.

Mar. 2021:

- Performed statistical analysis on the feature set used in the deep neural network.
- Analyze&Label was modified to support arbitrary classifications and labeling.
- Analyze&Label was written to function with arbitrary block sizes.

Apr. 2021:

- Added new computer vision features:
    - Elongation score to pits (binarized height data).
    - Histogram for grayscale optical data.
- Expanded the training dataset from 10 to 15 labeled images (each ≥2 stitched LCM images containing at least one crack).
- Train&Validate was modified to accept arbitrary block sizes as input.

May 2021:

- Started working on manuscript for publication in the Journal of Pattern Recognition.
- Added new computer vision features:
    - Multi-threshold optical angle and eccentricity.
- Evaluated a 16x16 tile size for training data.

June 2021:

- Train&Validate was rewritten to better reflect the current AI code.
- Designed an assemble tab to later be implemented into Train&Validate.
- Unit testing of Analyze&Label.

July 2021:

- Added the ability to randomly delete negative tiles from training set to balance the number of negative and positive examples while constraining the total database size (limiting the amount of augmentation needed).
- Changed the ensemble acceptance criteria to be based on training accuracy instead of testing accuracy.
- Analyze&Label was augmented to support "Crack Positivity" statistics when the file is saved.
- Train&Validate was augmented to support validating networks that were previously generated.
- Redesigned neural networks to save their structure while incorporating new statistics and parameters for the learning algorithm.

Aug. 2021:

- Increased the maximum tile size from 112x112 to 128x128.

- Train&Validate was modified to accept arbitrary classifications.
- The master label system was designed to better reflect users' intention when using arbitrary labels and classifications.
- Designed a simple MATLAB assemble save format.

Sept. 2021:

- Added dropout layer capability to the CNN algorithm.

6.0 Conclusions

A user-friendly Matlab GUI that reads data from either LCM or WAMS files was developed. The GUI can label features for reference, further examination, archival storage, or for development of a training set for machine learning. Features can be called out by user-specified thresholds, manual labeling or machine learning algorithms when they have been completed. The ability to rapidly label data is important because of the volume of data required for training machine learning algorithms.

The GUI has the flexibility to allow addition of improved ML algorithms, methods for data visualization, and statistical computations. Available data statistics include areas of pits within a defined range of pit depths, correlations between RGB or grayscale intensity and relative surface height, covariances between values associated with features, and feature histograms.

Overall, the development of supervised learning algorithms is hindered by a lack of training data. The machine learning algorithms for crack identification are in a state of partial development and require improvements to the true positive rate. This shortcoming is an artifact of the limited training data currently used, perhaps more so than the structure of the neural networks. The best results are had from an ensemble consensus over an ensemble of randomly generated DNN or CNN algorithms. The consensus accuracy yielded optimum true positive and true negative rates in excess of approximately 80%. However, additional validation testing is necessary.

WAMS (ZON file) image data was found to have a pixel pitch of 3.69μm compared to 1 μm for the LCM (vk4 file) data. The significance of the lower resolution for the WAMS data with regards to corrosion and crack identification is not clear at this point and will require further evaluation. Further, the ratio of dynamic range of the height data from WAMS to that from the LCM was approximately 41:20. This implies that the WAMS data should more accurately determine height than the LCM data. Again, the significance of the greater dynamic range of the WAMS data relative to the LCM data has not yet been evaluated.

7.0 Recommendations for Future Work

- Perform research to determine what image features mask the presence of cracks to the ML methods developed in this work.
    - Synthetically modify crack images to yield nearly 100% predictive accuracy, then add actual optical image features until the prediction accuracy is adversely affected.
    - Iterate on optical feature addition to isolate those features that "confuse" the ensemble of ML algorithms.
    - When the problem features are identified, develop methods to either remove them or to accommodate their presence into the ML ensemble so that predictive accuracy is acceptable.
- Acquisition of additional training data.
    - Augmentation via geometric manipulation.
    - Using data from corrosion coupons.
    - Make use of baseline canister images as a large collection of relatively easily labeled data, i.e., images demonstrating canister features in the absence of corrosion. In conjunction with multiclass labeling, this data could help to refine classification of non-

corroded surfaces. (Note: Training on baseline canister images solely as "non-crack"/"non-corrosion" data in binary-class training would further imbalance the training set and thus be detrimental to crack identification.)

- Investigate potential for using AI methods to relate surface features to diagnostic data for subsurface corrosion, voiding and deterioration.
    - Investigate whether there are surface features associated with subsurface voids and cracks observed in tomography images.
- Improvement to ML for crack detection.
    - Include hyperparameter optimization and other forms of classification algorithms, e.g. decision trees, support vector machines, etc.
    - Further development of AI for corrosion detection based on surface image and height data.
- Testing of ML training based on multiclass labeling.
    - Separately labeling non-crack features that are visually similar to cracks, such as machining marks, may help to drive the learning process to learn distinguishing features between them and thus improve crack identification over binary labeling.
    - Multiclass training/classification allows for detection of separate classes of corrosion features, e.g., cracks versus pitting.
- Processing of WAMS image data using the GUI interfaces. ML training using WAMS data (which requires sufficient WAMS files containing identified cracks/corrosion).
- Provide a more complete assessment of the impact of reduced image resolution for WAMS images on ML capability for identification of cracks and corrosion.
- Evaluate parity between LCM and WAMS measurements for areas associated with depressions, and for pit areal density (individual pits per area).

8.0 References

1 Berg, J.M., D.K. Veirs, E.J. Kelly, Juan G. Duque, S.A. Joyce, J.E. Narlesky, J.M. Duffey, J.I. Mickalonis, and K.A. Dunn, Test Plan for Assessing Potential for Stress Corrosion Cracking in the 3013 Inner Container Closure Weld Region (FY 2014). 2014, LA-UR-14-20785.

2 Martínez-Rodríguez, M.J., Laser Confocal Microscope for Analysis of 3013 Inner Container Closure Weld Region. 2017, SRNL-STI-2017-00589.

3 Hardy B., d'Entremont A., Bakos J., Clingenpeel T., Martinez-Rodriguez M., Garcia-Diaz B. "FY20 Progress Report: SRNL Analysis of ICCWR LCM and WAMS data for Corrosion and Cracking." SRNL-STI-2020-00355, Revision 0 (September 30, 2020).

4 Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, Cambridge MA (2016)

5 Aurélien Géron. Hands-On Machine Learning with Scikit-Learn &Tensorflow. O'Reilly, Sebastapol, CA (10/12/2018).

6 Shapiro L.G., Stockman G.C. "Computer Vision." Prentice Hall, Upper Saddle River, NJ.

9.0 Figures

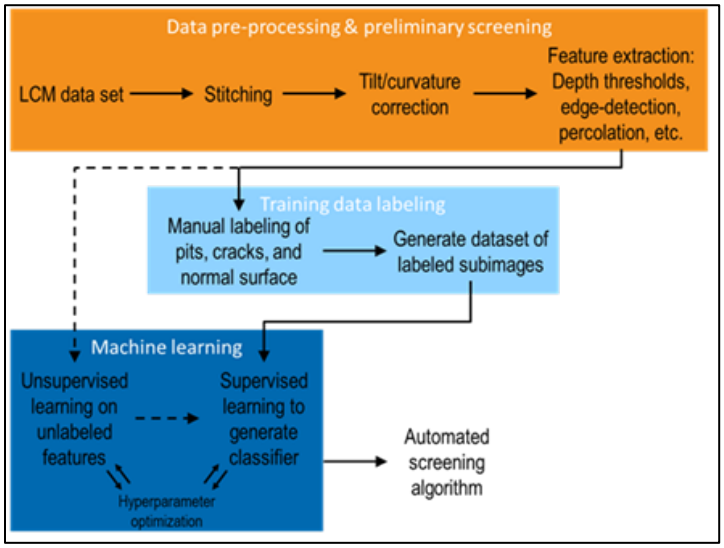


Figure 1. Flowchart for data processing and application of machine learning and other analysis methods for identification of corrosion and cracking in LCM and WAMS data.

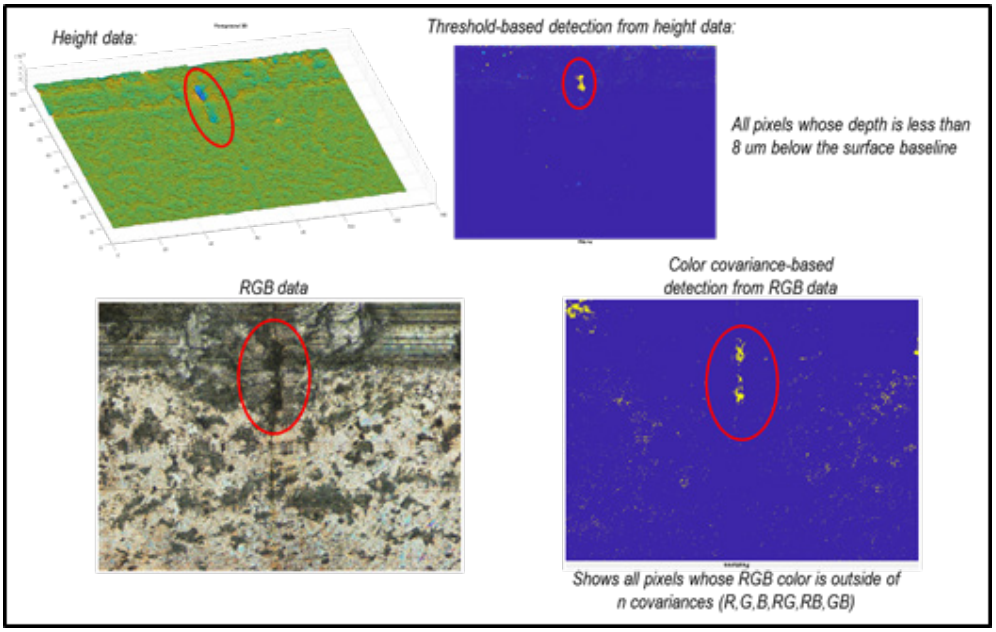


Figure 2. Crack and corrosion data from LCM images. Data channels include RGB, grayscale and height.

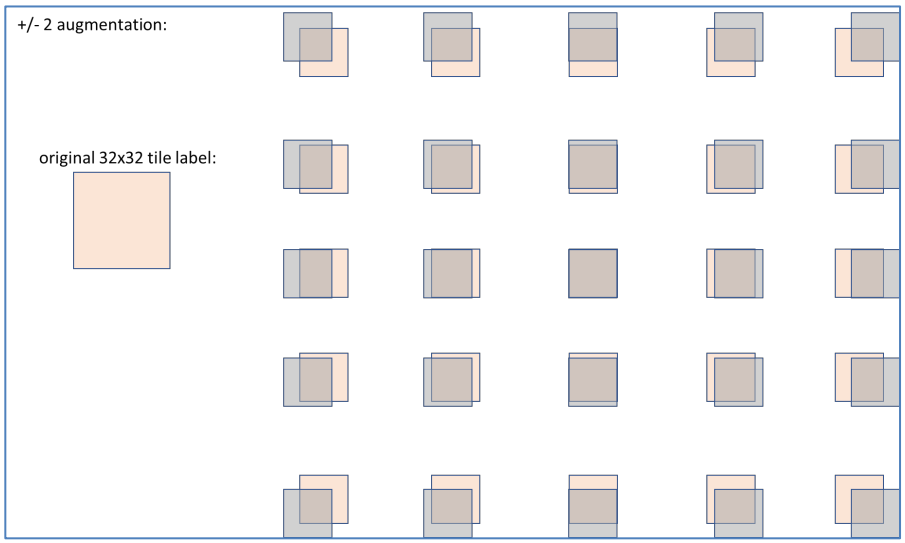


Figure 3. Data augmentation by translation of image. Translated offsets are shown in gray.

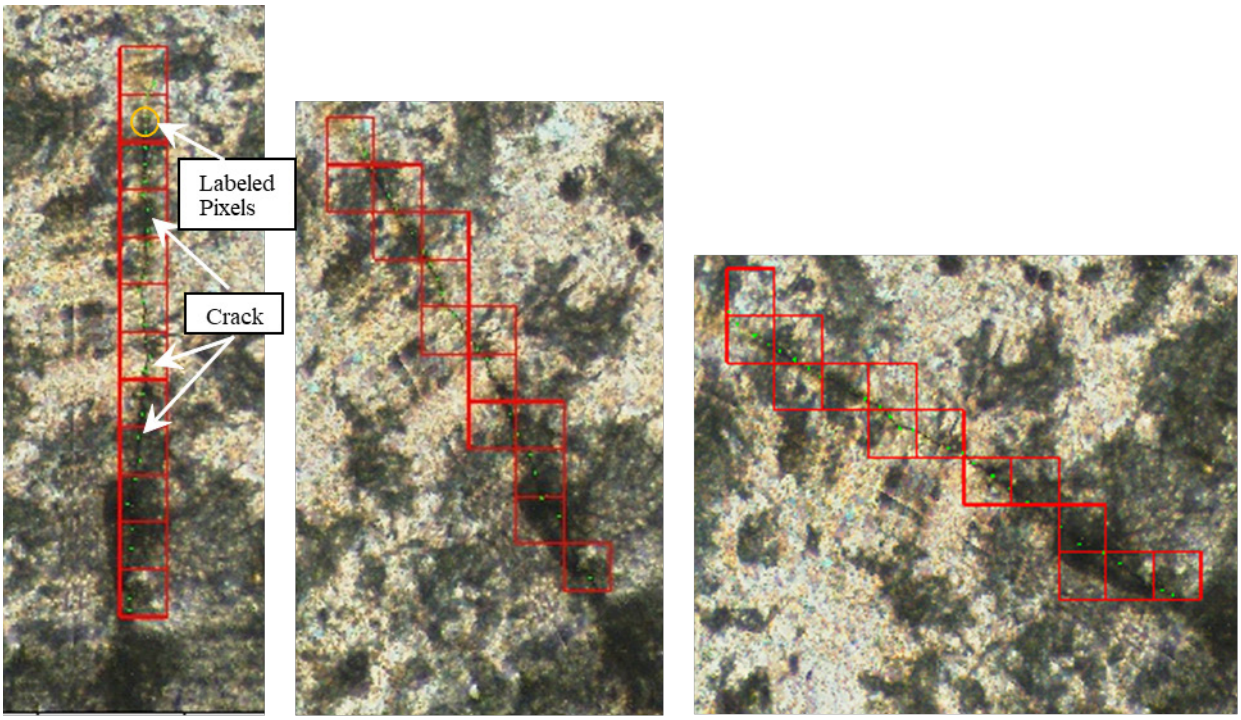


Figure 4. Data augmentation by rotation of crack image. The images are cropped show the highlighted region at a larger scale. Green dots (barely visible) along crack are pixels manually selected by the analyst via mouse click. The baseline orientation is shown in (A). When the image is divided into tiles, those containing a crack pixel are highlighted with a red boundary and labeled as a crack containing tile. Tiles that do not contain a crack pixel are defined and labeled as not containing a crack and are not highlighted. In (B), the image and crack pixels are rotated 30 degrees in the counterclockwise direction, and the image is divided into tiles. Again, those tiles containing a crack pixel are highlighted with a red boundary and are labeled as a crack containing tile. Tiles that do not contain crack pixels are labeled as such, and not highlighted. Similarly, (C) shows a rotation 60 degrees in the counterclockwise direction. The tile labeling

based on the marked pixels updates automatically in each new orientation, resulting in a multiple distinct sets of crack tiles from the same baseline image.

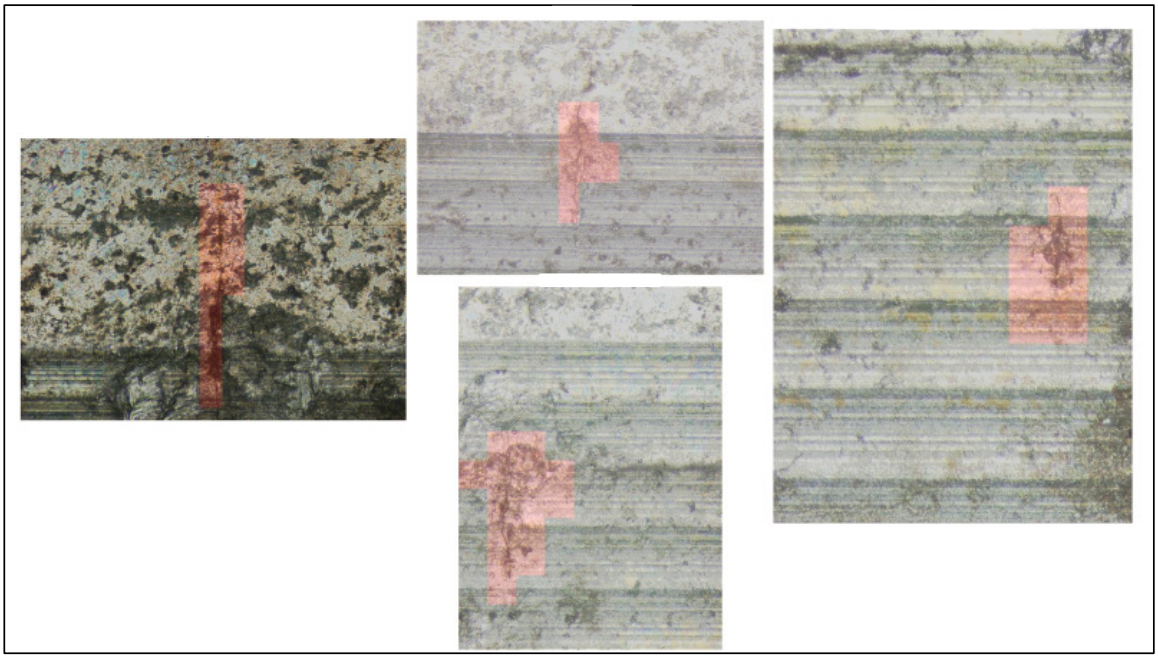


Figure 5. Examples of four labeled images, in this case divided into 112x112 pixel tiles (blocks) with labels on a per-tile basis. The training process applied augmentation to the crack data tiles using horizontal and vertical translation. Highlighted regions are tiles that contain cracks.

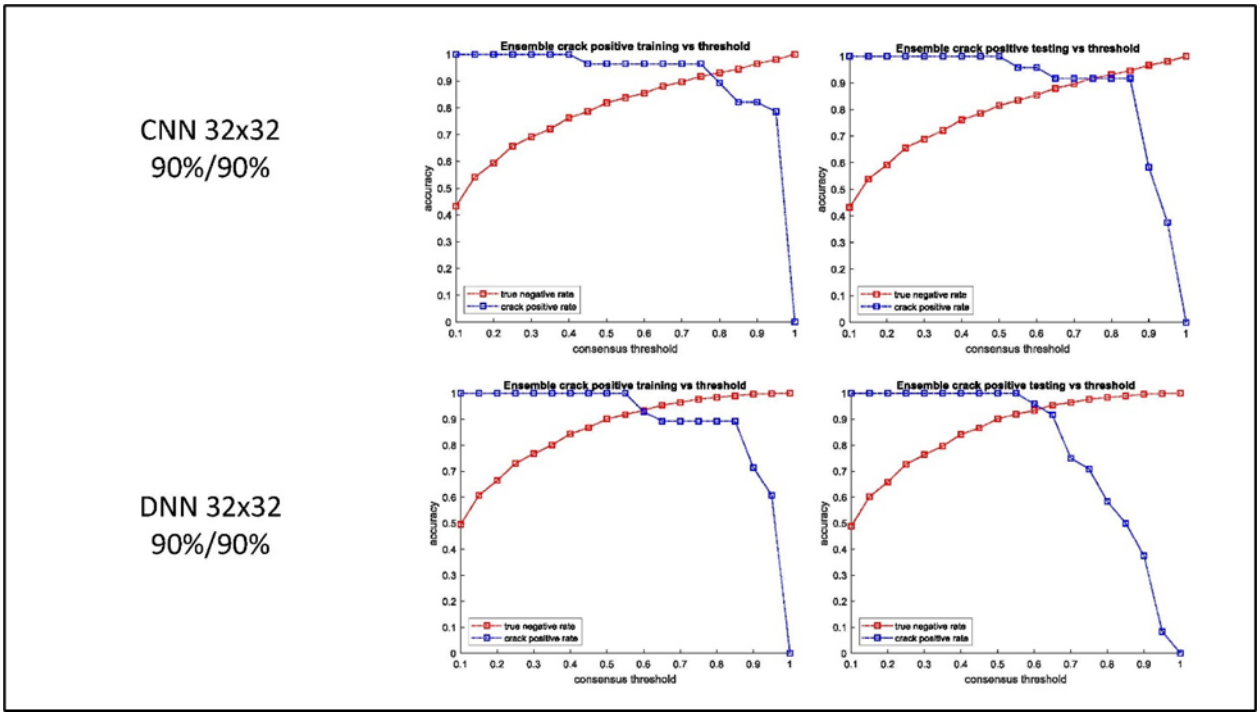


Figure 6. Comparison of the accuracy of crack identification from an ensemble consensus of CNN and DNN algorithms for a tile size of 32X32 pixels as the consensus vote threshold (horizontal axis)

is varied from 10% to 100%. The percentages at the left of the plots are the accuracy values corresponding to the intersection of the True Negative Rate (TNR) and the Crack Positive Rate (CPR).

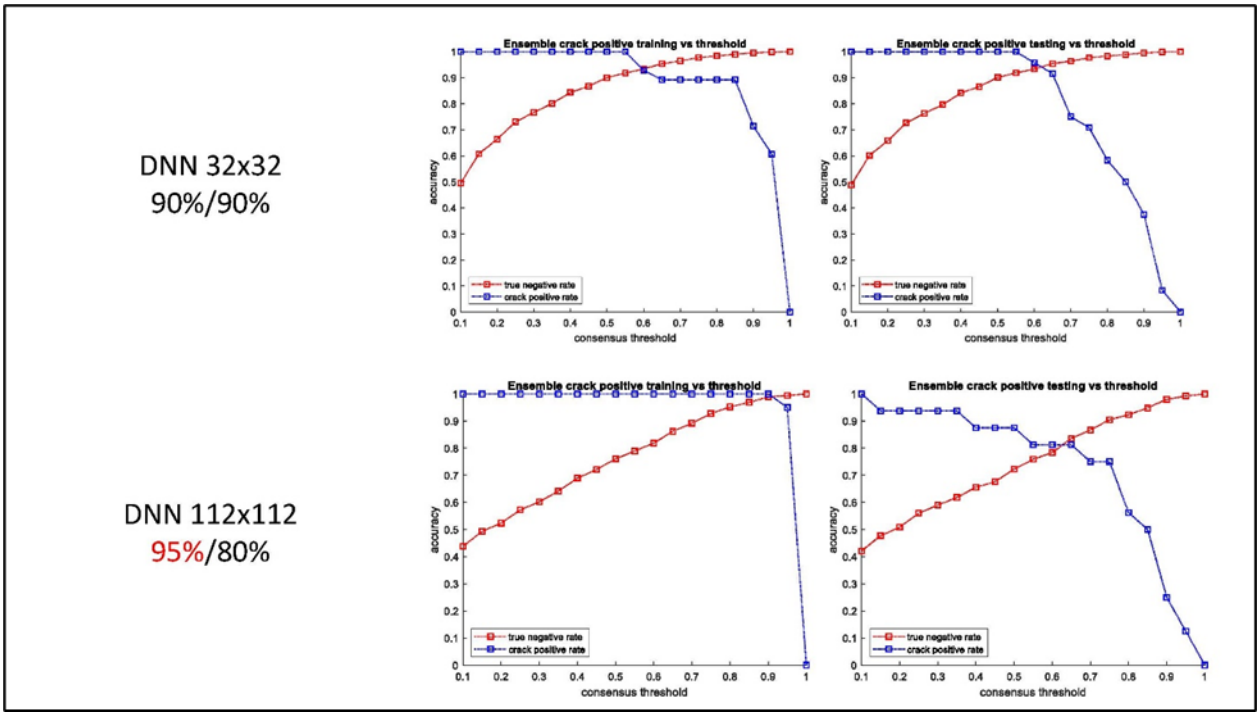


Figure 7. Comparison of the accuracy of crack identification from an ensemble consensus of CNN and DNN algorithms for a tile size of 112X112 pixels as the consensus vote threshold (horizontal axis) is varied from 10% to 100%. The percentages at the left of the plots are the accuracy values corresponding to the intersection of the True Negative Rate (TNR) and the Crack Positive Rate (CPR).

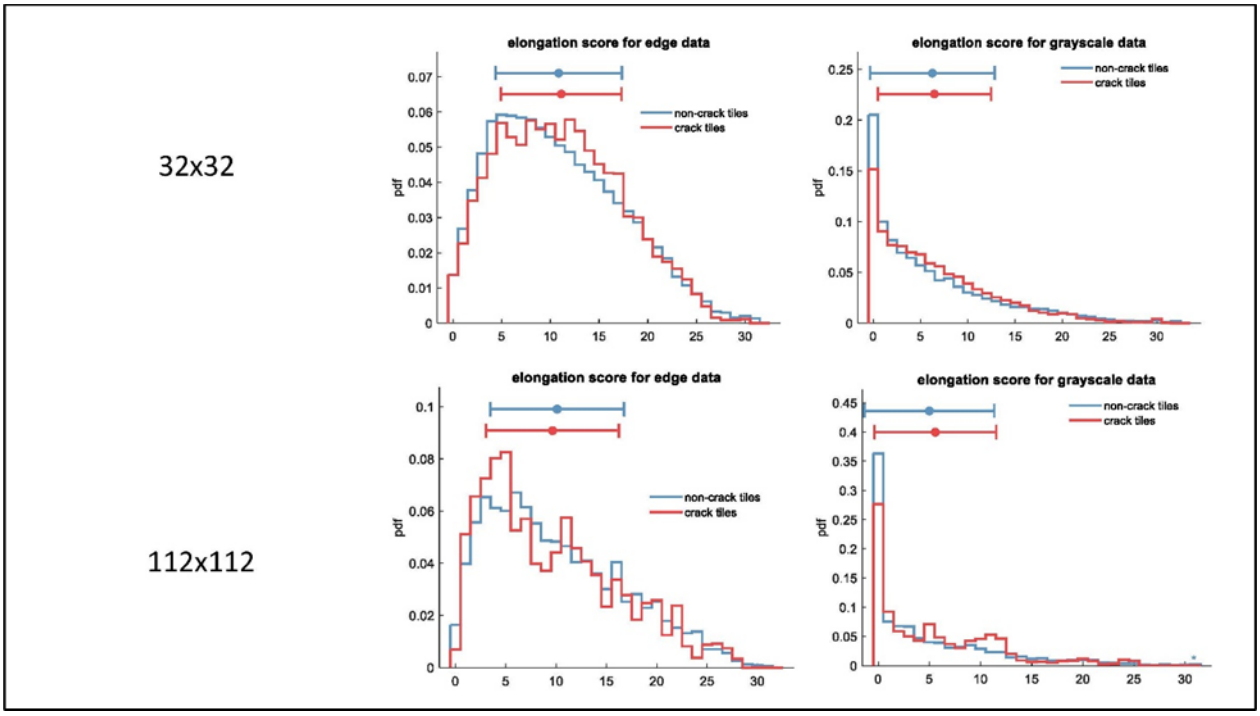


Figure 8. Relation between elongation scores for crack and non-crack tiles. Edge and grayscale data were used for 32 × 32 and 112 × 112 pixel tiles. The figures show that the elongation score displays no discernable difference for tiles with or without cracks.

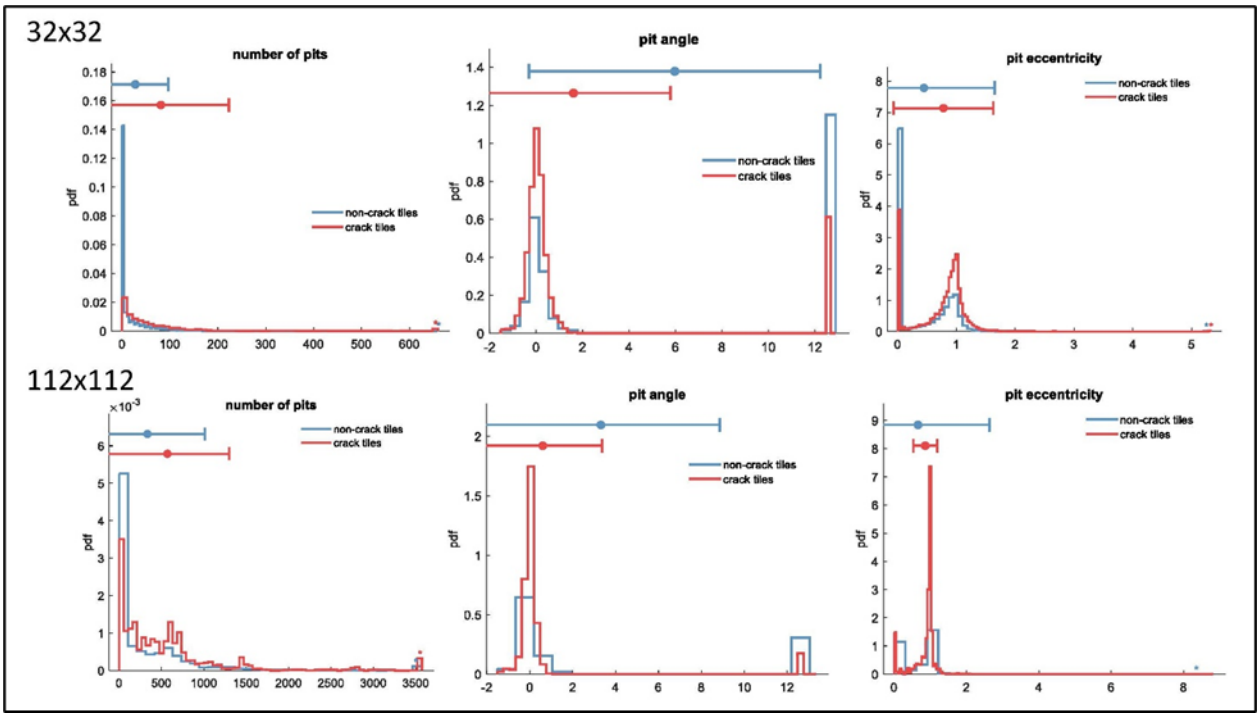


Figure 9. Relation between number of pits, pit angle and pit eccentricity for crack and non-crack tiles. For both 32 × 32 and 112 × 112 pixel tiles, these metrics had peaks at essentially the same locations for tiles with or without cracks.

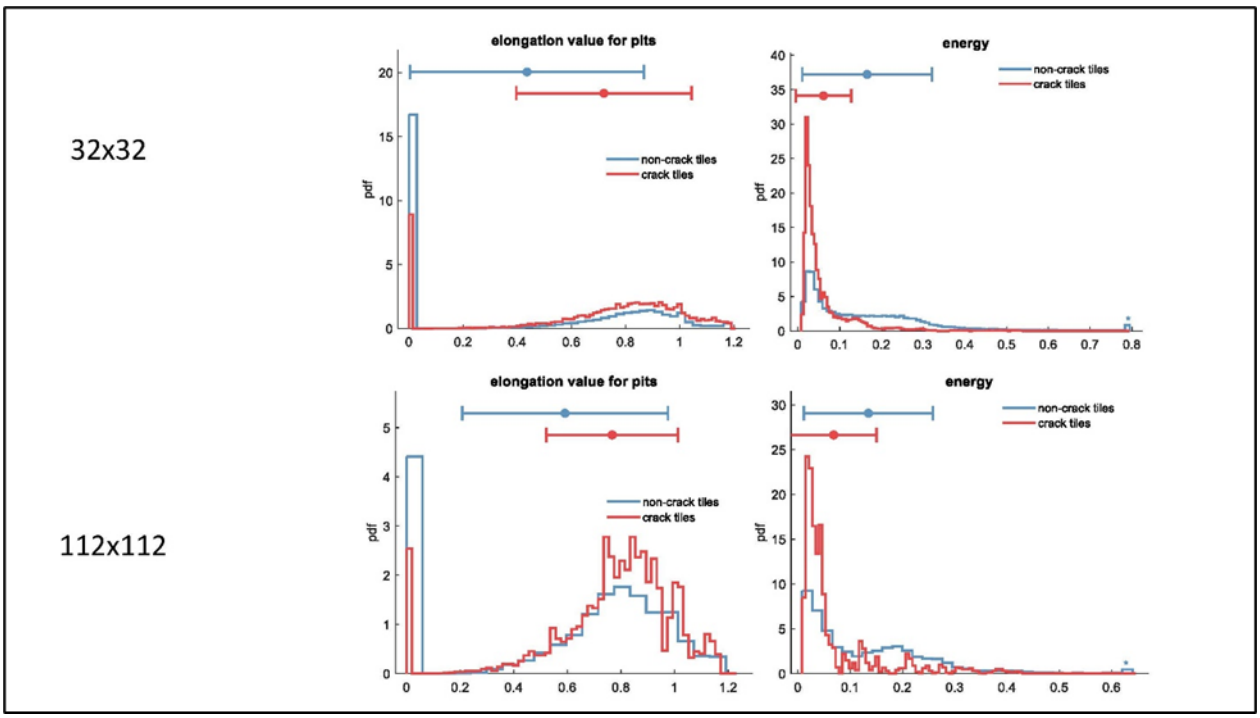


Figure 10. Relation between the pit elongation values and energy for crack and non-crack tiles. For both 32×32 and 112×112 pixel tiles, these metrics have peaks at essentially the same locations for tiles with or without cracks.

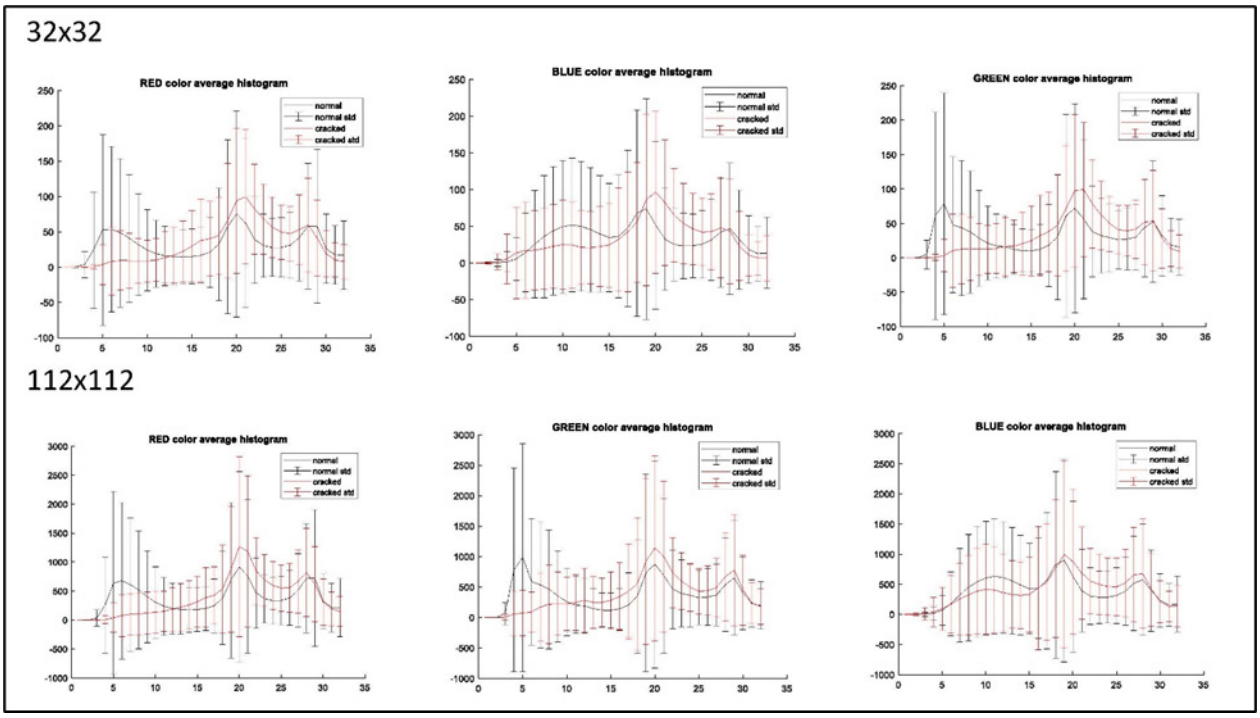


Figure 11. RGB color histograms and their standard deviations for 32×32 and 112×112 pixel tiles with and without cracks. No clearly identifiable difference is displayed for tiles with and without cracks.