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:
This work was prepared under an agreement with and funded by the U.S. Government. Neither the U.S. Government or its employees, nor any of its contractors, subcontractors or their employees, makes any express or implied:

1) warranty or assumes any legal liability for the accuracy, completeness, or for the use or results of such use of any information, product, or process disclosed; or
2) representation that such use or results of such use would not infringe privately owned rights; or
3) endorsement or recommendation of any specifically identified commercial product, process, or service.

Any views and opinions of authors expressed in this work do not necessarily state or reflect those of the United States Government, or its contractors, or subcontractors.
FY22 Progress Report: SRNL Analysis of ICCWR LCM and WAMS data for Corrosion and Cracking

Bruce Hardy (SRNL)
Anna d’Entremont (SRNL)
Stephanie Gamble (SRNL)
Jason Bakos (University of South Carolina)
Joseph Cammarata (University of South Carolina)
Roderick Fuentes (SRNL)

December 31, 2022
SRNL-STI-2022-00470, Revision 0
DISCLAIMER

This work was prepared under an agreement with and funded by the U.S. Government. Neither the U.S. Government or its employees, nor any of its contractors, subcontractors or their employees, makes any express or implied:

1. warranty or assumes any legal liability for the accuracy, completeness, or for the use or results of such use of any information, product, or process disclosed; or
2. representation that such use or results of such use would not infringe privately owned rights; or
3. endorsement or recommendation of any specifically identified commercial product, process, or service.

Any views and opinions of authors expressed in this work do not necessarily state or reflect those of the United States Government, or its contractors, or subcontractors.

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)
Stephanie Gamble (SRNL)
Jason Bakos (University of South Carolina)
Joseph Cammarata (University of South Carolina)
Roderick Fuentes (SRNL)

December 31, 2022
REVIEWS AND APPROVALS

AUTHORS:

__________________________________________________________
Bruce Hardy, SRNL

__________________________________________________________
Anna d'Entremont, SRNL

__________________________________________________________
Stephanie Gamble, SRNL

__________________________________________________________
Jason Bakos, University of South Carolina

__________________________________________________________
Joseph Cammarata, University of South Carolina

__________________________________________________________
Roderick Fuentes, SRNL

TECHNICAL REVIEW:

__________________________________________________________
Larry Deschaine, SRNL
APPROVAL:

Adrian Mistreanu, Manager
SRNL/Advanced Modeling and Simulation

Boyd J. Wiedenman, Manager
SRNL/Separation Sciences and Engineering

Frank M. Pennebaker, Director
SRNL/Chemical Processing and Characterization
EXECUTIVE SUMMARY

Algorithms for machine learning and data analysis for the 3013 Surveillance Program are being developed in an ongoing collaborative effort by the Savannah River National Laboratory (SRNL) and the University of South Carolina (USC). To detect the presence of corrosion and cracking, data is collected from large binary files generated by a Laser Confocal Microscope (LCM) or Wide Area 3D Measurement System (WAMS). Software is being developed to use the physical attributes in the data files (e.g., height, color, and grayscale values; all as functions of a location in a plane projection) to detect the presence of surface corrosion and cracking. 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 for training ML algorithms, flag significant features, execute Machine Learning (ML) algorithms, output parameters from trained ML algorithms, report ML model accuracy with respect to labeled data, and generate graphical representations for various analyses. Surface defects can be called out by setting user-specified thresholds, feature based analysis or machine learning algorithms. Enhancements to data labeling capability have been developed to address this essential precursor to application of ML routines. Efficient labeling is particularly important in view of the very large volume of data required to train ML 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 (for LCM data) and relative surface height, covariances between associated features, and feature histograms.

In FY22, algorithms for hourglass neural networks (HNN’s) were added to the suite of Convolutional Neural Networks (CNN’s) and Deep Neural Networks (DNN’s) being tested for crack identification. Work on the use of HNN’s is in preliminary stages but has yielded promising results.

As in the past, the development of supervised machine learning algorithms has been hindered by a lack of labeled 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 labeled training data currently available, perhaps more so than the structure of the neural networks. At present, the best results are had from:

1. A consensus over an ensemble of randomly generated Deep Neural Network (DNN) or Convolutional Neural Network (CNN) algorithms.
2. Hourglass networks, which are CNN encoder-decoders that employ residual connections between symmetric encoding and decoding layers.

Although both methods have yielded optimum true positive and true negative rates exceeding 80%, additional validation testing is necessary.
# TABLE OF CONTENTS

LIST OF FIGURES .......................................................................................................................... viii
LIST OF ABBREVIATIONS ............................................................................................................... ix

1.0 Introduction .................................................................................................................................. 1
2.0 FY22 Objectives ......................................................................................................................... 2
3.0 Approach ..................................................................................................................................... 2
4.0 Results/Discussion ....................................................................................................................... 4
5.0 FY22 Accomplishments .............................................................................................................. 6
6.0 Conclusions ................................................................................................................................. 8
7.0 Recommendations for Future Work ........................................................................................... 8
8.0 References .................................................................................................................................. 9
9.0 Figures ....................................................................................................................................... 10
LIST OF 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. .......................................................... 10

Figure 2. Crack and corrosion data from LCM images. Data channels include RGB, grayscale and height. ............ ............................................................................................................................................................. 10

Figure 3. Data augmentation by translation of image. Translated offsets are shown in gray. .................................. 11

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). .......................................................................... 11

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. .................................................... 12

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). .................................................... 12

Figure 7. Schematic of hourglass neural network. .................................................................................................. 13

Figure 8. Although the labeled crack is in the red boundary, the ML algorithms classify the depression in the yellow boundary as a crack. ..................................................................................................................... 13

Figure 9. This image contains labeled cracks that are difficult for the ML algorithms to separate from other marks on the image, resulting in low TNR’s. ............................................................................................................. 14

Figure 10. This image that is prone to very low TNR’s due to the large structure on the left of the image. In training the image typically gives reasonable classification accuracy. However, in testing, this proves to be the most difficulty image to classify. .................................................................................. 14

Figure 11. Laser intensity highlights cracks, while height data fails to do so. .......................................................... 15

Figure 12. The crack is detectable with the peak and intensity channels but does not appear clearly in the height channel. .......................................................................................................................................... 15

Figure 13. ML accuracy using grayscale data trained with 25% and 50% crack ratios. The grayscale threshold for pixels containing a crack was 127. ........................................................................................................... 16

Figure 14. ML accuracy using selectively reduced datasets containing 500 or 1000 tiles 25% of which contained cracks. G designates grayscale and H designates height+grayscale. Column “TNR w/out Image 8” denotes removal of a “problem” dataset. Image 8 is the image in Figure 10. .......................................................... 16

Figure 15. Accuracy for randomly generated CNN ensembles with 1 to 5 layers. Training data had 200 tiles/image, with 25% of the tiles containing a crack. 80% of the data was used for training and 20% for testing. .................................................. 17
# LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>CAE</td>
<td>Convolutional Autoencoder</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>CPR</td>
<td>Crack Positive Rate</td>
</tr>
<tr>
<td>CS</td>
<td>Computer Science</td>
</tr>
<tr>
<td>DNN</td>
<td>Deep Neural Network</td>
</tr>
<tr>
<td>DOE</td>
<td>Department of Energy</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HNN</td>
<td>Hourglass Neural Network</td>
</tr>
<tr>
<td>ICCWR</td>
<td>Inner Container Closure Weld Region</td>
</tr>
<tr>
<td>LCM</td>
<td>Laser Confocal Microscope</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
</tr>
<tr>
<td>RELU</td>
<td>Rectified Linear Unit</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green, Blue Colors</td>
</tr>
<tr>
<td>SCC</td>
<td>Stress Corrosion Cracking</td>
</tr>
<tr>
<td>SRNL</td>
<td>Savannah River National Laboratory</td>
</tr>
<tr>
<td>TNR</td>
<td>True Negative Rate</td>
</tr>
<tr>
<td>TPR</td>
<td>True Positive Rate</td>
</tr>
<tr>
<td>USC</td>
<td>University of South Carolina, Columbia, SC</td>
</tr>
<tr>
<td>WAMS</td>
<td>Wide Area 3D Measurement System</td>
</tr>
</tbody>
</table>
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\textsuperscript{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\textsuperscript{2}. With the LCM, a close visual examination is made of the ICCWR surface along with measurements of corrosion-related features\textsuperscript{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.

The objective of this project is the development of computer-facilitated methods to enable 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.

In FY22 the development of ML algorithms and the user interface was continued from the work done in FY21\textsuperscript{3} and FY20\textsuperscript{4}. Notably, in FY22, algorithms for hourglass neural networks (HNN’s) were added to the suite of Convolutional Neural Networks (CNN’s) and Deep Neural Networks (DNN’s) being tested for crack identification. Work on the use of HNN’s is in preliminary stages but has yielded promising results.

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 and color, all as functions of a location in a plane projection. For LCM data, grayscale images that are used in the ML algorithms are generated within the image analysis software by converting RGB optical images. 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 a subject matter expert to label features for training ML algorithms.
• Computes histograms that summarize the distribution of features.
• Incorporates and executes ML algorithms.
  o 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.
  o 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 FY22 Objectives
The FY22 objectives for this project were:

1. Continue development of the image processing software used to identify and quantify surface features from LCM and/or WAMS data related to corrosion on surfaces of 3013 canisters. The software validation will be the true positive-false positive and the true negative rate-false negative rate for event classification that is acceptable to customer.

2. Continue development of the user interface capable of inputting large quantities of 3013 LCM and/or WAMS image data and using the image processing software in Item 1 to identify the location of surface features that meet or exceed thresholds input by the user. This includes improvements to the method of piecewise image evaluation. The overall software development is a continuation of work by SRNL/USC from prior contracts.

3. Use labeled training and test data for 3013 corrosion features (provided by SRNL and LANL experts) to explore the selection and implementation of appropriate machine learning algorithms. Ultimately, the algorithms are to be accessed through the user interface described in Item 2.

4. SRNL and USC will incorporate the machine learning algorithms developed and evaluated in FY 2021 into an end-user data analysis framework that had been developed in parallel during the same period.

5. Add a new front-end to the previously developed software to permit the use of WAMS imagery data files while retaining the capability for using LCM data.

6. Characterize the impact of the difference between WAMS and LCM image resolution on crack and corrosion detection. Investigate ML methods that could improve any loss of accuracy.

3.0 Approach
Corrosion is strongly, but not exclusively, associated with surface pitting and cracking, coloration, along with surface feature geometry, see Figure 2. However, 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 Hourglass Neural Networks (HNN’s), Deep Neural Networks (DNNs), Convolutional Neural Networks (CNN’s), gradient methods, statistical characterization, correlations, and filters\textsuperscript{5,6,7,8}. These methods are embedded in the overall evaluation process 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 CNN and DNN 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. Labeling for HNN networks included a more detailed pixel-level labeling of cracks to effectively use the HNN structure, with any tile that includes part of a labeled crack being classified as a crack tile.

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. The initial approach to 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). In Figure 4, 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. The tiles for HNN’s were selected using a new random sampling approach. As a result, augmentation for these algorithms was applied by prescribing the percentage of crack and non-crack tiles sampled to force oversampling of tiles overlapping the crack.

Augmenting via translation, rotation or oversampling ensures that the cracks in the extra training tiles remain realistic examples and enforces 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 approached through the use of ML methods. Generally, the training of the ML algorithms suffered due to the small amount of labeled 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 a combination of rotation, translation (vertical and horizontal) and/or oversampling 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-FY21, 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 FY22, 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 an argmax (min(CPR, TNR)) 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 Figure 6. 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

\[
CPR \equiv \text{Crack Positive Rate} = \frac{\sum_{i=1}^{N_{Cracks}} 1 \text{ if } TP_i > 0}{N_{Cracks}}
\]

\[
TNR \equiv \text{True negative rate}
\]

\[= \quad \frac{TN}{TN + FP}\]

and

\[
TPR \equiv \text{True positive rate}
\]

\[= \quad \frac{(\text{number of tiles correctly classified as containing a crack})}{(\text{actual number of tiles containing a crack})}
\]
\[ \equiv \frac{TP}{TP + FN} \]

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. Figure 6 shows 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.
- 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 \( \text{argmax} (\min(\text{CPR}, \text{TNR})) \) 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 Figure 6).
- Tile size, which is based on the number of pixels in an image, affects the training and validation accuracy.
  - 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
  - Figure 6 shows the crack positive and true negative rates for a 32x32 tile, for CNNs and DNNs and a range of 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 addition to the ensemble DNN and CNN algorithms, an Hourglass Neural Network (HNN) was applied to the binary data. The hourglass network is a CNN encoder-decoder that employs residual (or skip) connections between symmetric encoding and decoding layers. Encoders, decoders and residual units are discussed in Reference 6. A schematic of the hourglass network is shown in Figure 7.

To improve the accuracy of the ML algorithms an investigation was made of the images that proved to be the most challenging. This study was directed at finding underlying contributions to errors in classification. For example, in Figure 8 the cropped field is much larger than the annotation in the red outline. Additionally, the actual evidence for a crack is questionable. In this case, algorithms typically classify the depression in the yellow outline as a crack.

Figure 9 presents another image that has been very difficult to classify. This image contains 6 cracks that are all very difficult to separate from other marks on the image. ML analysis of this image has never yielded a 100% CPR with any reasonable TNR.

Figure 10 shows an image that is prone to very low TNR’s due to the large structure on the left of the image. For the grayscale, intensity and height channels these feature values are distinct from those associated with a crack. Hence, in training the image typically gives reasonable classification accuracy. However, in testing, this proves to be the most difficulty image to classify because without seeing features associated with the large structure in training, the network assumes them to all be cracks.

A solution to the lack of accuracy might be had from dataset alteration, including selective stacking of feature channels, crop size, feature maps (kernels) for particular channels and care in cropping images with prominent non-crack features. Figures 11 and 12 show examples of the effects of image intensity and channels on elucidation of cracks.

Accuracy of the ML algorithms, and their ensembles, based on selective data (data adaptation) is presented in Figures 13-14. Figure 13 shows the results of ML algorithms applied to a grayscale image where potential crack locations were limited to pixel values of less than 127. In Figure 13, the ML algorithms were trained on cases where the ratio of crack containing tiles was unconstrained, 25% and 50% to compare the impact of different rates of oversampling.

Figure 14 shows the accuracy of ML algorithms by using selectively reduced datasets containing 500 or 1000 tiles, 25% of which contained cracks. G designates grayscale and H designates height. Column “TNR w/out Image 8” denotes removal of a “problem” dataset.

Figure 15 shows preliminary results for training a CNN to classify the (image-format) output of the HNN as crack or non-crack rather than using a fixed threshold criterion. The accuracy for 5 randomly generated CNN ensembles with 1 to 5 layers is shown. Here, the training data for the CNN used 200 tiles/image, with 25% of the tiles containing a crack. For each case, 80% of the data was used for training and 20% for testing.

5.0 FY22 Accomplishments

The FY22 accomplishments in chronological order are:

Oct. 2021:
- Started FY22 using an ensemble of convolutional neural networks with randomized hyperparameters, trained with height+optical channel tiles from a 15-image dataset.
- Evaluated the impact of training parameters on accuracy using sweep analysis.
Nov. 2021:
- Examined individual tiles to assess the features that make individual tiles more difficult to classify.

Dec. 2021:
- Re-wrote the bootstrap sampling code (performs data sampling with replacement) for performance.
- Implemented rotation augmentation of labeled crack image data.

Jan. 2022:
- Developed a method to evaluate the accuracy of individual tiles.

Feb 2022:
- Explored the impact of different train-test mixes on model accuracy.

Mar. 2022:
- Performed an analysis of model accuracy on individual tiles for each crack in the training set to determine the relative classification challenge of each crack relative to the others.
- Began splitting the train and test set by image instead of by tiles to examine the relative classification difficulty of the individual images in the training set.
- Started exploring the use of the hourglass network model.

Apr. 2022:
- Implemented the hourglass network.
- Developed a new method for classifying pixels as opposed to tiles using a tile “score”.

May 2022:
- Refined the implementation of the hourglass network.
- Developed a new result visualization code.
- Developed a new crack labeling tool that allows for more precise crack classification.

June 2022:
- Evaluated hourglass network accuracy with four training images.
- Developed new methods for classifying each tile based on the output pixel values given by the hourglass network.
- Cross-referenced image database against the global image “map” of FY16 DE05 provided by the SRNL team.
- Developed a method to automatically catalog individual discrete crack structures in the dataset.

July 2022:
- Developed a method to generate training datasets by randomly sampling tiles over pre-labeled images.
- Added new training images to the evaluation of the hourglass network.
- Developed a method to train a network against every possible combination of training images.
- Developed and characterized a new method for automatic selection of thresholds for classifying tiles and compared it with fixed-threshold methods.
- Characterized additional classification techniques that used the mean, standard deviation, and elongation of pixel groups.
- Characterized the use of different channels for input data: height only, optical only, intensity only.
- Developed J-score metric to compare different scoring methods.

\[
J = \frac{\text{(mean of score over all crack tiles)} - \text{(mean of score over all non-crack tiles)}}{\text{(std dev of score over all crack tiles)} + \text{(std dev of score over all non-crack tiles)}}
\]
Aug. 2022:
• Developed a method to constrain the ratio between the number of tiles containing a crack and those not containing a crack.
• Evaluated the impact of tile ratio on accuracy and J-score for various scoring functions and input channel combinations.

Sept. 2022:
• Investigated the effect of dataset alterations on ML accuracy.
  o Selective cropping.
  o Ratio of crack tiles in training set.
  o Combination of feature channels.
  o Removal of ambiguous data.
• Tested combination of grayscale and grayscale+intensity features.

6.0 Conclusions
Selective application of data can produce ML accuracy of 80% or greater, which is as good or better than networks discussed in the literature. Cracks occur in a very small fraction of the images but data augmentation by rotation, translation and oversampling of images was found to provide a useful supplement. While CNN and DNN ensembles that have been developed in FY21, and improved in FY22, are showing very good results with available training and testing data, hourglass networks are also quite accurate and need further investigation. The ability to label pixel groups and the development of a new crack labeling tool allows for more precise crack classification.

In studies that investigated the underlying causes for poor classification, it was found that combinations of feature channels improve the detection of cracks, especially if intensity is included. Visually, Figure 11 shows that intensity highlights cracks that are obscured for height data. Because WAMS data does not include intensity, alternate mechanisms will need to be developed to ensure good classification accuracy.

At this point, the ML algorithms have not been applied to WAMS output due to an insufficient amount of labeled training data. It will be a priority to obtain labeled WAMS data in FY23 and use it to train and test the algorithms.

The latest ML algorithms and labeling methods can readily be implemented into the Matlab GUI developed in FY22. The GUI can read data from either LCM or WAMS files and label features for reference, further examination, archival storage, or for development of a training set for machine learning. By using the GUI, features can be called out by user-specified thresholds, manual labeling or machine learning algorithms when they have been completed.

7.0 Recommendations for Future Work
• Application of the ML algorithms to WAMS data.
  o Algorithm modification as needed for classification accuracy.
  o Processing of WAMS image data using the GUI interfaces. ML training using WAMS data (which requires sufficient WAMS files containing identified cracks/corrosion).
  o Evaluate parity between LCM and WAMS measurements for areas associated with depressions, and for pit areal density (individual pits per area).
• Acquisition of additional training data.
  o Augmentation via geometric manipulation.
  o Using data from corrosion coupons.
  o 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.)

- Continue 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.
  - Generate networks using intensity data.
  - Compare different evaluation methods success on different images to determine if it would be beneficial to have an ensemble where crack evaluation is different (i.e., scoring tiles off max pixel value, elongation, binary classifier, etc.).
  - Generate an ensemble using greyscale, height, and intensity data.

- 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.

- 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.

8.0 References

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 to 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).
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. Schematic of hourglass neural network.

Figure 8. Although the labeled crack is inside the red boundary, the ML algorithms classify the depression in the yellow boundary as a crack.
Figure 9. This image contains labeled cracks that are difficult for the ML algorithms to separate from other marks on the image, resulting in low TNR’s.

<table>
<thead>
<tr>
<th>Grayscale</th>
<th>Intensity</th>
<th>Height</th>
<th>Labeled Cracks</th>
</tr>
</thead>
</table>

Figure 10. This image that is prone to very low TNR’s due to the large structure on the left of the image. In training the image typically gives reasonable classification accuracy. However, in testing, this proves to be the most difficulty image to classify.
Figure 11. Laser intensity highlights cracks, while height data fails to do so.

Figure 12. The crack is detectable with the peak and intensity channels but does not appear clearly in the height channel.
Figure 13. ML accuracy using grayscale data trained with 25% and 50% crack ratios. The grayscale threshold for pixels containing a crack was 127.

<table>
<thead>
<tr>
<th>Model</th>
<th>Train</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPR</td>
<td>TNR</td>
<td>CPR</td>
<td>TNR</td>
<td>TNR w/out Image 8</td>
<td>Threshold</td>
</tr>
<tr>
<td>No Crack Ratio</td>
<td>70.1%</td>
<td>98.5%</td>
<td>79.2%</td>
<td>94.1%</td>
<td>95.9%</td>
<td>127</td>
</tr>
<tr>
<td>25% Crack Ratio (G-25)</td>
<td>96.9%</td>
<td>95.7%</td>
<td>81.3%</td>
<td>90.5%</td>
<td>94.0%</td>
<td>127</td>
</tr>
<tr>
<td>50% Crack Ratio (G-50)</td>
<td>100.0%</td>
<td>93.0%</td>
<td>83.3%</td>
<td>87.8%</td>
<td>90.9%</td>
<td>127</td>
</tr>
</tbody>
</table>

Figure 14. ML accuracy using selectively reduced datasets containing 500 or 1000 tiles 25% of which contained cracks. G designates grayscale and H designates height. Column “TNR w/out Image 8 [the image in Figure 10]” denotes removal of a “problem” dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Train</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPR</td>
<td>TNR</td>
<td>CPR</td>
<td>TNR</td>
<td>TNR w/out Image 8</td>
<td>Threshold</td>
</tr>
<tr>
<td>G-25 (1000 tiles)</td>
<td>96.9%</td>
<td>95.7%</td>
<td>81.3%</td>
<td>90.5%</td>
<td>94.0%</td>
<td>127</td>
</tr>
<tr>
<td>G-25 (500 tiles)</td>
<td>90.6%</td>
<td>93.0%</td>
<td>97.9%</td>
<td>84.8%</td>
<td>88.5%</td>
<td>127</td>
</tr>
<tr>
<td>H-25 (1000 tiles)</td>
<td>93.8%</td>
<td>97.5%</td>
<td>95.8%</td>
<td>87.8%</td>
<td>98.6%</td>
<td>127</td>
</tr>
<tr>
<td>H-25 (500 tiles)</td>
<td>90.6%</td>
<td>96.9%</td>
<td>91.7%</td>
<td>88.1%</td>
<td>98.70%</td>
<td>127</td>
</tr>
</tbody>
</table>
Figure 15  Accuracy for randomly generated CNN ensembles with 1 to 5 layers to perform the final classification step after HNN application. Training data had 200 tiles/image, with 25% of the tiles containing a crack. 80% of the data was used for training and 20% for testing.

<table>
<thead>
<tr>
<th>CNN</th>
<th>80% train</th>
<th></th>
<th>20% test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TNR</td>
<td>TPR</td>
<td>TNR</td>
<td>TPR</td>
</tr>
<tr>
<td>1</td>
<td>100%</td>
<td>99.7%</td>
<td>97.9%</td>
<td>73.7%</td>
</tr>
<tr>
<td>2</td>
<td>100%</td>
<td>86.0%</td>
<td>98.4%</td>
<td>70.2%</td>
</tr>
<tr>
<td>3</td>
<td>100%</td>
<td>100%</td>
<td>97.7%</td>
<td>75.4%</td>
</tr>
<tr>
<td>4</td>
<td>80.0%</td>
<td>97.6%</td>
<td>72.2%</td>
<td>82.5%</td>
</tr>
<tr>
<td>5</td>
<td>99.9%</td>
<td>79.5%</td>
<td>98.9%</td>
<td>70.2%</td>
</tr>
</tbody>
</table>