

# Automated Pipeline for Photovoltaic Module Electroluminescence Image Processing and Degradation Feature Classification

Ahmad Maroof Karimi, *Graduate Student, Member, IEEE*, Justin S. Fada , Mohammad Akram Hossain , Shuying Yang, Timothy J. Peshek , Jennifer L. Braid , *Member, IEEE*, and Roger H. French , *Member, IEEE*

**Abstract**—An automated data analysis pipeline is developed to preprocess electroluminescence (EL) module images, and parse the images into individual cells to be used as an input for machine learning algorithms. The dataset used in the study includes EL images of three 60 cell modules from each of five commercial brands at six steps of damp heat exposure, from 500 to 3000 h. Preprocessing of the original raw EL images includes lens distortion correction, filtering, thresholding, convex hull, regression fitting, and perspective transformation to produce planar indexed module and single cell images. Parsing of PV cells from each of the preprocessed 90 EL module images gives us 5400 cell images, which are function of module brand and damp heat exposure step. From the dataset, two unique degradation categories (“cracked” and “corroded”) were observed, while cells that did not degrade were classified as “good.” For supervised machine learning modeling, cell images were sorted into these three classes yielding 3550 images. A training and testing framework with 80:20 sampling ratio was generated using stratified sampling. Three machine learning algorithms (support vector machine, Random Forest, and convolutional neural network) were trained and tuned independently on the training set and then given the test set to predict the scores for each of the three models. Five-fold cross validation was done on training set to tune hyper-parameters of the models. Model prediction scores showed that convolutional neural network outperforms support vector machine and Random Forest for supervised PV cell classification.

**Index Terms**—Computer vision, electroluminescence (EL) imaging, photovoltaic (PV) cell segmentation, prediction, supervised machine learning.

Manuscript received January 23, 2019; revised March 31, 2019; accepted May 23, 2019. This work was supported in part by the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement DE-EE0007140, and in part by the Ohio Third Frontier, under Wright Project Program Award tech 12-004. (*Ahmad Maroof Karimi and Justin S. Fada contributed equally to this work.*) (*Corresponding author: Roger H. French.*)

A. M. Karimi, J. S. Fada, M. A. Hossain, T. J. Peshek, J. L. Braid, and R. H. French are with the Solar Durability and Lifetime Extension (SDLE) Research Center, Case Western Reserve University, Cleveland, OH 44106 USA (e-mail: axk962@case.edu; jsf77@case.edu; mah228@case.edu; tjp3@case.edu; jlb269@case.edu; roger.french@case.edu).

S. Yang is with the SunEdison Inc., Belmont, CA 94002 USA (e-mail: shuyingy@gmail.com).

An EL image dataset of a PV module undergoing step-wise Damp Heat exposure is available for training and testing at <https://osf.io/qt5hj/>, DOI: [10.17605/OSF.IO/QT5HJ](https://doi.org/10.17605/OSF.IO/QT5HJ).

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

Digital Object Identifier 10.1109/JPHOTOV.2019.2920732

## I. INTRODUCTION

THE development of automated feature detection in electroluminescence (EL) images of photovoltaic (PV) devices would bring greater levels of accuracy and speed to cell and module characterization in research, manufacturing, and operation and maintenance sectors of the PV value chain. Herein we discuss a method for using statistical and machine learning algorithms to classify PV cell images based on reported degradation features [1]. The results of our machine learning methods on degradation features, from stepwise EL images of full-size modules exposed to damp heat conditions, demonstrate approaches to quantitative analysis of EL images. Machine learning can be very useful in assessing module performance and reliability in the laboratory, manufacturing, and the field [2], [3].

Currently, popular techniques for PV module degradation analysis include mere visual inspection, current–voltage curve (*I*–*V*) tracing, and imaging techniques, such as UV fluorescence, EL, photoluminescence, and thermography [4]. *I*–*V* curve tracing method is useful for holistic analysis of PV modules as it provides a complete sweep of the electrical properties of a device under test from short circuit to open circuit condition [5]–[7]. However, this measurement is a coarse gauge of mechanistic behavior in a module, as it uses the module’s pair of electrical leads as the single input/output connection, and does not allow for analysis of individual components. Therefore classification of degradation type by *I*–*V* curve tracing can only be narrowed to a few possible degradation modes [4]. Of the remaining module characterization methods, visual inspection and imaging are traditionally done qualitatively for classification of degradation types. The output data from an EL image typically provide millions of spatially resolvable data points registered by the camera sensor as a PV module powered in forward bias emits photons in the near infrared. As a result of the increased data density over other common measurement methods, such as *I*–*V*, EL imaging has been increasingly utilized as a tool for analysis of PV devices [4], [8], [9]. Some researchers have successfully used EL imaging to derive local cell electrical properties (sometimes in conjunction with other image types) [10], [11], with image features, such as cracks, dark spots, and busbar corrosion present for further study.

Recent work in automating PV image analysis have included binary classification of module-level thermography images [12],

and of EL images of cells on a manufacturing production line [13]. Binary PV cell classification after cell segmentation from module-level EL images has also been demonstrated [14], however cells were simply classified into categories of “functional” and “defective” without distinguishing defect type. Complete classification of cell features has not been successfully automated, and most classification is currently done by trained researchers [4]. With qualitative assessment, there is inherently human bias associated with classification particularly regarding matters, such as whether a feature is prominent enough to be recorded, or the feature’s relative degree of severity. These systematic biases are typically reasonable when used as internal lab metrics, however the differences become important when comparing results from different research groups or studies.

Additionally, collection of EL data has become simpler and less expensive with many modified consumer-off-the-shelf cameras present in the literature alongside commercially available equipment [8], [15], [16]. Therefore with the ability to generate larger volumes of EL images that need classification, the task of manual classification is becoming quite laborious and time consuming [17].

We aim to solve these issues for analyzing EL images by validating the use of computer vision and statistical learning methodologies [7] in a distributed computing environment [18] for the reliable quantitative classification of PV cell defect types. In this work, we demonstrate preliminary use of this technique to classify two defect types, cell cracking and busbar corrosion, along with a third category of “good/clean” cells that have no signs of degradation. By confirming the use of this technique herein, expansion of this model to many other defect types will follow.

The data used in this work include fifteen 60-cell commercial modules spanning five brands exposed under accelerated damp heat testing (85 °C and 85% relative humidity) [19]. Images were captured with a Sensovation AG (coolSamBa HR-830) high resolution camera having a large field of view. Each module was measured at six 500-h interval steps from 500 to 3000 h of damp heat exposure. The resulting 90 module images contain 5400 individual cells, of which 3550 cell images exhibited cracking, busbar corrosion, or no degradation, the three cell states studied in this work. The remaining cells exhibited a variety of features which did not yield enough data for classification purposes. The ratio of training to testing data was chosen to be a commonly used ratio of 80:20 [20]. An augmentation method is applied on training images, i.e., 2840 (80%) out of 3550 cell images is transformed into three additional images resulting in 11360 and remaining 20% or 710 PV cell image are assigned as a test set. The test set is used for reporting the results, whereas model training and hyper-parameter tuning are done by five-fold cross validation using the training set.

This work is an extension of our proceedings publication titled “Electroluminescent Image Processing and Cell Degradation Type Classification via Computer Vision and Statistical Learning Methodologies” [21]. In Section II of this updated work, we present the major enhancements from the previous work in EL image processing, including the following:

- 1) lens distortion correction;
- 2) improvement in module edge detection.

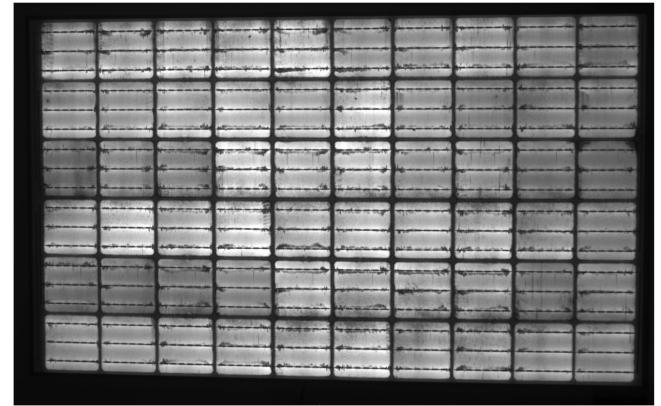


Fig. 1. Unprocessed grayscale EL image of a PV module.

Machine learning models and tuning of their parameters are discussed in Section III. We also present methods for image augmentation in enhancing training for machine learning algorithms and preprocessing to optimize image resolution, then briefly discuss classification algorithms and prediction scores of the models. The major improvements from our previous works are as follows:

- 1) use of cross validation to tune in model parameters;
- 2) use of convolutional neural network (CNN) instead of artificial neural network;
- 3) addition of more comprehensive score metrics to report more accurate results that take into account skewness in the dataset.

## II. IMAGE PROCESSING

When raw EL images are collected, the module typically has some misalignment relative to the boundary of the image (seen in Fig. 1). This can be due to loading variability, the need to avoid operator IR signatures in the image during measurement, or manufacturing variability. Another major image defect is radial distortion, also known as barrel distortion, which is an inherent property of the lens design that deforms an image and makes the subject look curved [22].

An automated image processing pipeline was developed to systematically align and planar index the active module area in the image and allow for a high volume throughput of data. For its computational efficiency, ease of implementation, and rich image processing/analysis toolboxes, the open-source coding language Python [23] was used with the additional packages of NumPy [24], SciPy [25], scikit-image [26], [27], and OpenCV [28], [29].

An image is loaded into memory as a grayscale JPEG image, to reduce computation from the original RGB channel image and the file size for storage. For some image processing problems, all three channels are useful, but for our research, there is no noticeable loss of accuracy by using grayscale. The first step in the image processing step is to correct the lens distortion. Straight lines from undistorted image are transformed to an arc of a circle in a distorted image. Research work on lens distortion correction has been an important topic in the field of computer

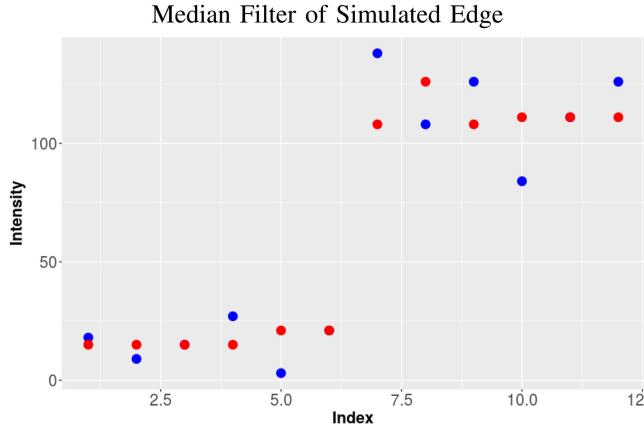


Fig. 2. Blue data points represent a vector of simulated noisy data with an edge step, such as when intensity values increase at the edge of a cell in an EL module image from the background to active cell area. Red points are the postmedian filter data. A kernel size of 3 was used for this simulation [21].

vision, with some published works on barrel correction schema [30]–[32]. In our observation, barrel distortion caused straight lines to curve approximately 5–7 pixels over a 1500 pixel length. Equation (1) describes the radial distortion division model to correct a distorted image. Let  $(x_d, y_d)$  be the coordinates of a pixel in a distorted image,  $(x_i, y_i)$  are the coordinates of the same pixel in ideal or undistorted image,  $r$  is the radial distance of the points  $(x_d, y_d)$  from the center point of radial distortion  $O(x_0, y_0)$ , and  $k$  is the parameter of radial distortion. In the simplest form, the model equations can be written as

$$\begin{aligned} x_i &= x_d(1 + k_1 r^2 + k_2 r^4) \\ y_i &= y_d(1 + k_1 r^2 + k_2 r^4) \end{aligned} \quad (1)$$

where  $r^2 = x_d^2 + y_d^2$  because  $O(0, 0)$  is considered as the center of radial distortion. To remove distortion from PV image, we used OpenCV [28] and applied an iterative method to find the values of  $k_1$  and  $k_2$  in (1). The resultant image is free from radial distortion or has small distortion error. The selected values of  $k_1$  and  $k_2$  are  $-1.5e^{-5}$  and  $-1.0e^{-7}$ , respectively, for this dataset.

The next step after correction of radial distortion is perspective transformation. For this step, the image is filtered using a  $3 \times 3$  pixel kernel that convolves on the image to perform a median filtering, which reduces any noise spikes across the pixel matrix and retains feature edges [33]. Fig. 2 demonstrates the filtering process visually using a simulated vector of noisy data with an edge step (seen as blue points) and filtered result (seen as red points). This edge step represents the boundary of an image feature, such as an increase in values from the background to the active cell regions. The most important characteristic of this median filter is that the edge is retained, which would not be the case if, for instance, a Gaussian filter were used, which “blurs” the image when smoothing [34].

With our interest in identifying the active cell regions of the PV module EL image only, the background values are removed. Using a histogram [represented in two dimensions as a spectral map of binned pixel values in Fig. 3(a)], a global threshold of

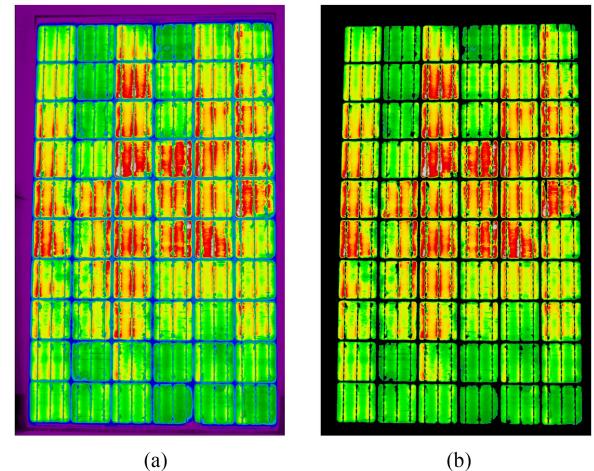


Fig. 3. (a) Median filter applied to the original image, displayed using a spectral color mapping of pixel intensity values to binned color ranges. From this image, it can be seen that the purple and blue colors (pixel intensity ranges) represent the background of the image. (b) Setting those background pixels in Fig. 3(a) to black (value of 0) results in only the active cell region as nonzero values [21].

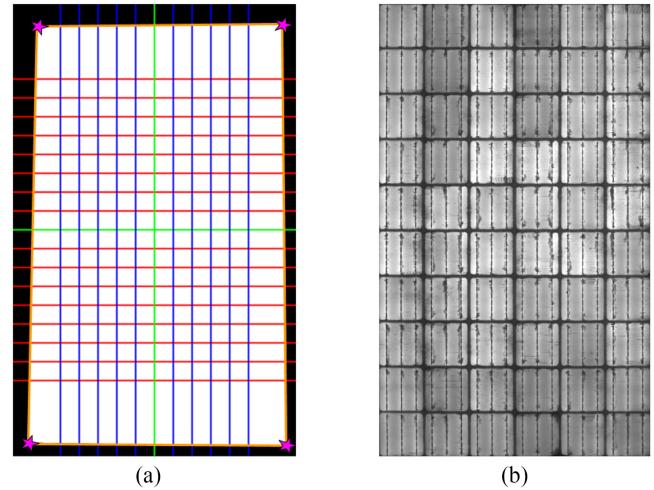


Fig. 4. (a) Visualization of the steps used to identify the four corners of the active cell region. The white area is the identified active module area from the convex hull algorithm. The green lines are reference lines for the midpoints in the  $x$  and  $y$  dimensions. Blue lines indicate horizontal vector slices (relative to the original image in Fig. 1). Red lines indicate vertical vector slices. Orange lines represent linear regression fits of the active area edges. Magenta stars represent the four corners of the active cell regions where the linear regression fits intersect. (b) Perspective transformed image using the four corners identified in Fig. 4(a) [21].

the image is used to reassign all noncell pixels [purple and blue in Fig. 3(a)] to 0. The result of the background threshold is seen in Fig. 3(b). With only cell areas as nonzero, a convex hull algorithm generates the smallest possible convex set of pixels containing all the nonzero data points (“shrink-wrapping” the nonzero data points) [35]. In the next step, the pixels pertaining to the convex hull identified area are set to 1, shown as the white area in Fig. 4(a), creating a binary matrix.

Next, it is important to find the edges of the PV module, which seem to be out of plane or at some small angle of rotation from

the sides of the original image (see Fig. 1). To find the edges in a manner that is reliable and tolerant to variability due to degradation of the outer cells, the edge is found by sampling the set of points that lie on the expected edge boundary and then fitting of those points using linear regression.

To find left and right edges of a module, as in Fig. 4(a), we scan a row (represented by red line) of pixels and detect the points where pixel value changes from 0 to 1 (index of the left edge) and when pixel value changes from 1 to 0 (index of the right edge). We repeat this method 100 times on equally spaced rows in the center 70% of the module image, because we have observed that the cells on the corners of the module degrade faster and are therefore unreliable for edge detection. Similarly, we scan a column of pixels to detect top and bottom edges of the module. Again in Fig. 4(a), 100  $x$ -axis and  $y$ -axis parallel slices are selected (exemplary red and blue lines) and the points where these lines move from black region to white or vice-versa are detected as edge points. After finding 100 edge points on each of the four sides, we run a linear regression for each side to get a straight line for each edge of the module. The four intersection points of the lines resulting from the regression fits of the module edges correspond to the four corners of the module active area.

With these four points the surface plane of the PV module is defined in the image. Using domain knowledge that PV modules are rectangular in shape, a perspective transformation for that geometry is performed on the data points of the median filtered image. The perspective transformation is a method by which a two-dimensional image in one plane can be projected into another two-dimensional plane [36]. For example, the module in Fig. 1 is tilted out of the image plane, but using perspective transformation, the PV module can be projected in the image plane [see Fig. 4(b)]. The corresponding points in the two planes are related by (2), where  $\mathbf{H}$  is a homography matrix. Let a point in original plane be represented by vector  $\mathbf{I} = (I, J, 1)^T$ , and  $\mathbf{i} = (i, j, 1)^T$  represent the corresponding point in the image plane

$$\mathbf{I} = \mathbf{H}\mathbf{i}. \quad (2)$$

In matrix form, (2) is represented as

$$\begin{bmatrix} IW \\ JW \\ 1W \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad (3)$$

where  $W = h_{31}i + h_{32}j + 1$ . Equation (3) can be simplified to yield the following equations as:

$$\begin{aligned} h_{11}i + h_{12}j + h_{13} &= Iih_{31} + Ijh_{32} + I \\ h_{12}i + h_{22}j + h_{23} &= Jih_{31} + Jjh_{32} + J. \end{aligned} \quad (4)$$

If we add missing zero terms in (4) and write the elements of  $\mathbf{H}$  matrix as vector, then (4) can be rewritten in the form  $\mathbf{Ah} = \mathbf{b}$ . In matrix form, (5) has eight unknown terms and the dimension of matrix  $\mathbf{A}$  for  $n$  points is  $2n \times 9$ . Thus, every point  $((I, J) \leftrightarrow (i, j))$  makes up two rows of matrix  $\mathbf{A}$ , and we need

four points to solve the following:

$$\begin{bmatrix} i & j & 1 & 0 & 0 & 0 & -Ii & -Ij \\ 0 & 0 & 0 & i & j & 1 & -Ji & -Jj \\ \vdots & & & & & & & \end{bmatrix} \begin{bmatrix} h_{11} \\ \vdots \\ h_{33} \end{bmatrix} = \begin{bmatrix} I \\ J \\ \vdots \end{bmatrix}. \quad (5)$$

To solve (5), the vector  $\mathbf{h}$  that minimizes the residual  $R = \|\mathbf{Ah}\|$  (zero for four points) subject to smallest value of  $\|\mathbf{h}\|$  is given by an eigenvector corresponding to the smallest eigenvalue of  $\mathbf{A}^T \mathbf{A}$ , calculated from singular value decomposition [37]. Once the coordinates of the corners are found, all images can be transformed as matrices throughout these calculations have the same dimensions. This maps the active regions (contained within the image hull) to a uniformly sized canvas of  $1500 \times 2500$  pixels, which is approximately the original image size. We refer to this as the planar indexed EL image.

Before cell classification, individual cell images must be extracted into their own files. Since the PV module images are now in uniform alignment, the images can be sliced into equal areas using the number of column-wise and row-wise cells. For the  $6 \times 10$  cell modules used herein, 5  $x$ -axis and 9  $y$ -axis divisions successfully obtain the cell areas desired. There can be very minor variations in cell spacing, which appear around the borders of the extracted cell images. However, it is appropriate to cease further manipulations and refinement as the entire active cell area is contained within each extracted image file. This meets our requirement for input into learning algorithms, which have a high tolerance for random variations across the dataset. Additionally, for a highly degraded cell that contains dead areas along the edge of the cell, an algorithm, such as the convex hull used above would only capture the active cell region, losing valuable information regarding the cell's degradation.

### III. MODELING AND PREDICTION

For this study, we developed a supervised model for image classification of PV cells by degradation mode following the framework described in the following.

#### A. Image Classification and Augmentation

Using the pipeline outlined above, 90 images of 60-cell modules were processed to yield 5400 individual cell images. For supervised learning, each of these cell images was classified manually into a single category of good/clean, cracked, busbar corroded, or other, of which the first three groups were used for modeling (see Fig. 5). To reduce the bias, all cells were labeled by a single author, and then cross verified by another author. It should be noted that since a researcher performed the initial classification of 5400 cell images, there may be some misclassification in the training set used for modeling. However, statistical models like those outlined below are tolerant to a small amount of error from supervised classification [20], [38]. In addition, after initial classification and model training, use of this method will result in systematic quantitative classification contained in a scalable data analytics pipeline.

After sorting, the three selected categories resulted in 3550 images in total. The remaining 1850 cells, some of which are

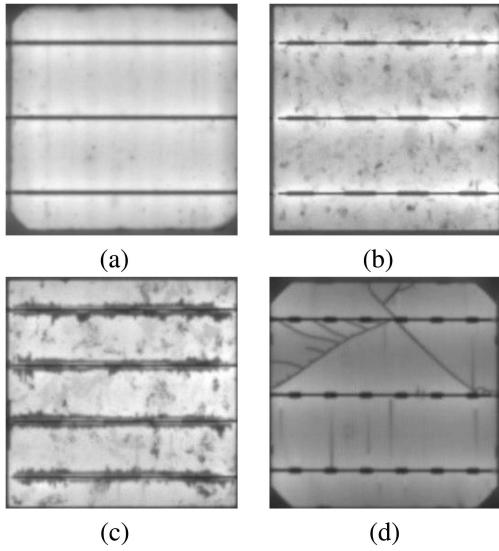


Fig. 5. Examples of the three cell types used for classification modeling [21]. (a) Good. (b) Good. (c) Corroded. (d) Cracked.

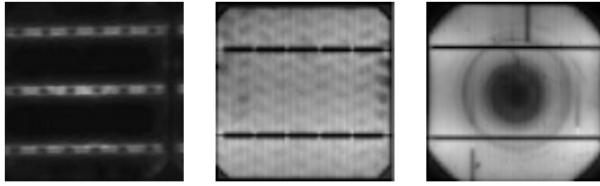


Fig. 6. Examples of cells which did not conform to the three categories used for classification modeling.

shown in Fig. 6 could not be strictly classified into the three categories. Some of the cells among 1850 cells were degraded even more than the left cell in Fig. 6 or were completely non-functional. The other two cells in this figure also show degradation other than corrosion or cracking and thus were not included for classification modeling.

The training set was augmented to produce four times the data and achieve a total of 11 360 cell images. The augmentation process function takes each image as an input and flips it about its center  $x$ -axis, center  $y$ -axis, and rotates it  $180^\circ$ , as seen in Fig. 7. These augmented images are all equally possible configurations in real-world cells and are perceived as unique additional data for modeling.

#### B. Image Preprocessing

Corrosion and cracking only become apparent after 2500 h of damp heat exposure, limiting the portion of the dataset representing those features. When corrosion appears in an EL image, it is common that a high percentage of the cells per module (if not all) exhibit corrosion. However, while cracking appears more often in longer exposed modules, the number of cracked cells is only a small to moderate percentage of all cells in the module (depending on module brand). Therefore of the 12 070 (training and testing dataset) postprocessed cell images, only 3.5% cells are cracked, compared with 12.0% corroded and 84.5%

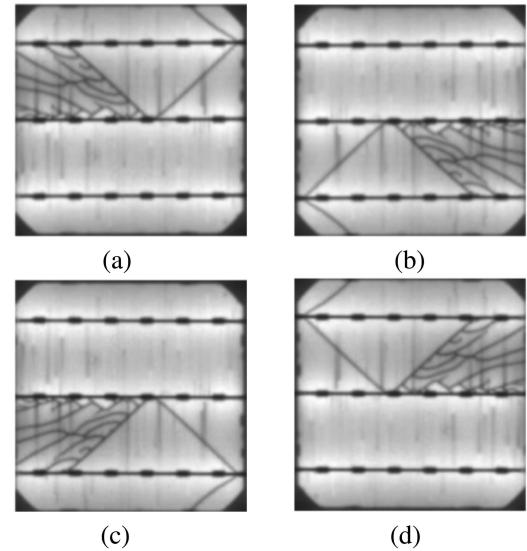


Fig. 7. Cell image augmentation. (a) Original. (b) Rotated  $180^\circ$ . (c) Flipped about  $x$ -axis. (d) Flipped about  $y$ -axis [21].

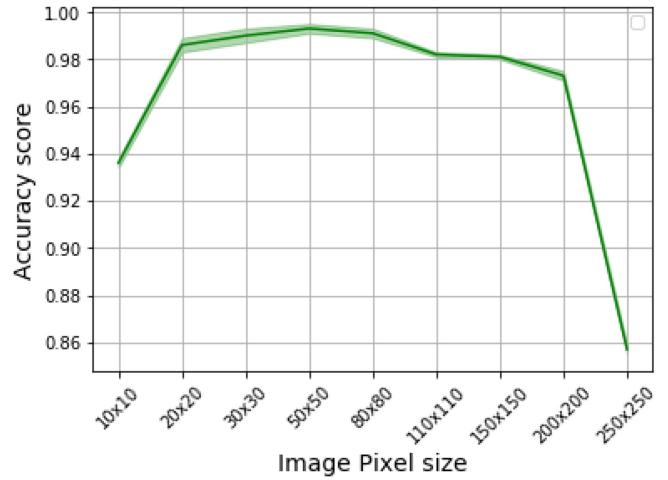


Fig. 8. Classification accuracy score metric for different image dimensions. The trend shows the optimal value of accuracy score when the image is of size  $50 \times 50$  pixels.

good/clean. As a result of unbalanced categories and to ensure that all data are represented in the training and testing sets, stratified sampling was used to generate training and testing datasets [39]. This commonly used method samples from each of the three categories of data to ensure that the percentage of samples in each group is maintained [40].

The individual cells after cell extraction from module images were  $250 \times 250$  pixels in size. In order to find the optimum image resolution for cell classification, the cell images were resized to dimensions ranging from  $10 \times 10$  to  $250 \times 250$  pixels. The process of optimizing cell image size has two main advantages: Higher classification accuracy by removing noise, and reduced computation time. From the results in Figs. 8–11, we find that the cell images of dimension  $50 \times 50$  have the best combined metrics for accuracy, precision, recall, and f-measure. The scores

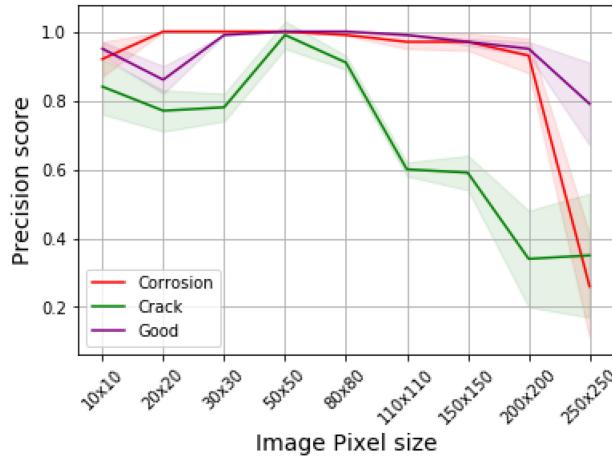


Fig. 9. Precision score measure for different image dimensions. The trend shows the optimal value of precision score when the image is of size  $50 \times 50$  pixels.

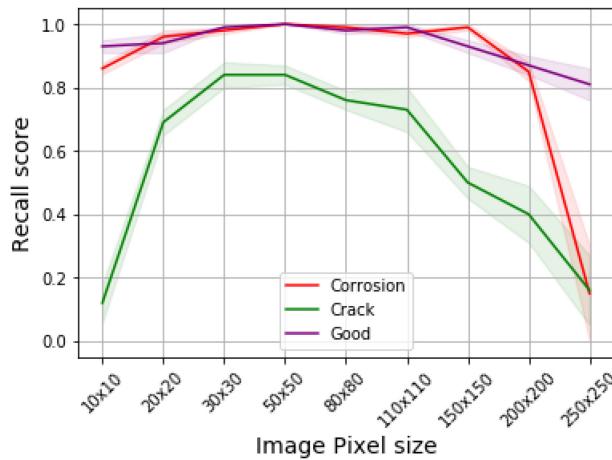


Fig. 10. Recall score measure for different image dimensions. The trend shows the optimal value of recall score when the image is of size  $50 \times 50$  pixels.

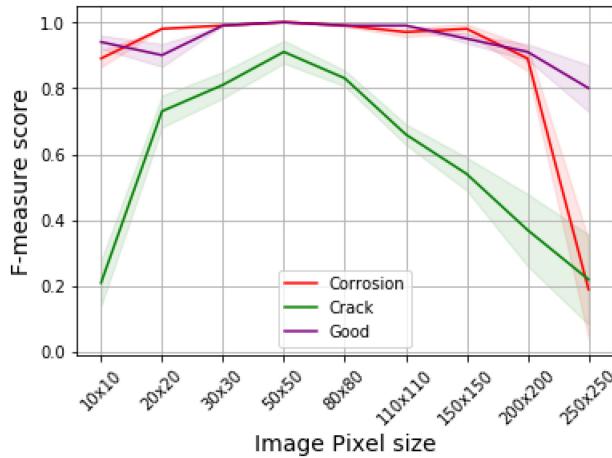


Fig. 11. F-measure score metric for different image dimensions. The trend shows the optimal value of f-measure score when the image is of size  $50 \times 50$  pixels.

in Figs. 8–11 were calculated for the support vector machine (SVM) algorithm, and similar trends were observed for Random Forest (RF). For CNN, the trend changes slightly; metric scores are low for low image resolution and improve until the image size becomes  $50 \times 50$  pixels, at which point the metrics saturate and give the same result for higher resolutions. Therefore all cell images are resized to  $50 \times 50$  pixels and then normalized using min–max intensity scaling. Normalization standardizes the dataset, so that the classification algorithm becomes less sensitive to small intensity changes and more likely to detect features correctly. The score metrics used in Figs. 8–11 are defined as follows. *Accuracy* is the number of data points classified correctly in each class over the total number of data points. *Precision* is the fraction of the relevant class that was predicted correctly. In other words, it is the ratio of data points that were classified correctly over all data points that were correctly or incorrectly classified for a particular class. *Recall* is the proportion of the actual relevant class identified correctly. *F-measure* combines precision and recall, and is the harmonic mean of the two.

In case of multiple classes, we take the mean of the values across three classes as shown later in the results in Table VI. Mathematically, these measures are represented as

$$\begin{aligned} \text{accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \\ f - \text{measure} &= \frac{2(\text{precision} * \text{recall})}{\text{precision} + \text{recall}} \end{aligned} \quad (6)$$

where TP is True Positive, FP is False Positive, TN is True Negative, and FN is False Negative. Four different result metrics are used because the dataset is highly skewed, so accuracy alone is not sufficient to characterize the model's performance.

### C. Classification Algorithms

Using supervised learning in a training and testing framework, we explored three state-of-the-art algorithms to classify cell-level EL images: SVM, RF, and CNN [41], [42]. As the input for SVM and RF, we convert an image matrix of size  $n \times n$  to a vector of length  $n \times n$ , while for CNN, the input is the image matrix itself. We trained and tested the learning models on the high performance computing (HPC) cluster at Case Western Reserve University, Cleveland, OH, USA. The configuration of the system for SVM and RF was an Intel(R) Xeon(R) CPU E5-2450 @ 2.10 GHz processor, 24 GB memory, and 12 CPU cores  $\times$  2.69 GHz. For CNN, we employed a GPU with configuration of NVIDIA GP100GL (Tesla P100) card, 12 GB memory @ 33 MHz.

SVM is a supervised machine learning method based on statistical learning theory and structural risk minimization invented by Vladimir Vapnik to solve both classification and regression problems [43]. For classification, SVM constructs hyperplanes in a multidimensional feature space that can separate different types of class labels [20], [38]. SVM is represented

with the model form

$$y_i = \mathbf{w}^T \cdot \mathbf{x}_i + b \quad (7)$$

where  $\mathbf{w}$  is a weight vector,  $\mathbf{x}_i$  is an input vector,  $b$  is a bias term,  $y_i \in \{1, -1\}$ , and  $i \in 1 \dots N$ ,  $N$  being the number of data points in the training set. To determine the optimum hyperplane, a quadratic programming problem to minimize the objective function  $\frac{1}{2} \|\mathbf{w}\|^2$  is solved under the inequality constraint  $y_i(\mathbf{w}^T \cdot \mathbf{x}_i - b) \geq 1$ . However, when the dataset is nonlinearly separable, a slack variable  $\xi_i \geq 0$  is used for each training point to introduce a penalty for misclassification. Then, the objective function becomes

$$C \sum_{i=1}^N \xi_i + \frac{1}{2} \|\mathbf{w}\|^2 \quad (8)$$

where  $C$  is a parameter that controls the tradeoff of the penalty term and the margin of the hyperplane under the constraint  $y_i(\mathbf{w}^T \cdot \mathbf{x}_i - b) \geq 1 - \xi_i$ . If the value of  $C$  is too large, the resulting model will over fit and loses its generalization property. If  $C$  is too small, the model is biased with high training error. The Lagrangian function used in (8) under the given constraint to solve for the minimum point is given by

$$\begin{aligned} L(\mathbf{w}, b, \mathbf{a}) = & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ & - \sum_{i=1}^N a_i \{y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) - 1 + \xi_i\} - \sum_{i=1}^N \mu_i \xi_i \end{aligned} \quad (9)$$

where  $a_i \geq 0$  and  $\mu_i \geq 0$  are Lagrangian multipliers. To find the minima, we take the partial derivative of (9) with respect to  $\mathbf{w}$ ,  $b$ , and  $\xi_i$  and equate them to 0. By solving, we get the value of weight  $\mathbf{w} = \sum_{i=1}^N a_i y_i \mathbf{x}_i$ , thus replacing  $\mathbf{w}$  in (7), predicted value of new data point  $\mathbf{x}$  is calculated as:

$$y = \sum_{i \in \text{Support Vectors}} a_i y_i \mathbf{x}_i \cdot \mathbf{x} + b \quad (10)$$

where Support Vectors are the data points in the training set for which  $a_i \neq 0$ . The general form of function  $\mathbf{x}_i \cdot \mathbf{x}$  is  $\phi(\mathbf{x}_i)\phi(\mathbf{x})$  can be rewritten in kernel formal as follows:

$$y = \sum_{i \in \text{Support Vectors}} a_i y_i k(\mathbf{x}, \mathbf{x}_i) + b. \quad (11)$$

The function  $k(\mathbf{x}, \mathbf{x}_i)$  is a kernel function and the radial basis kernel function is defined as

$$k(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2), \quad \gamma > 0. \quad (12)$$

A grid search approach was used to find the optimum values of  $C$  in (8) and  $\gamma$  in (12). The range of hyper-parameters for grid search for  $C$  and  $\gamma$  are shown in Table I. The optimized values of hyper-parameters  $\gamma$  and  $C$  are 0.01 and 1000, respectively. We also observe from Figs. 12 and 13 how the training score changes with  $C$  and  $\gamma$ , respectively.

TABLE I  
VALUES OF HYPER-PARAMETERS  $C$  AND  $\gamma$  FOR TRAINING SVM MODEL

Parameters	Search Vector
$C$	$[10^1, 10^2, 10^3, 10^4, 10^5]$
$\gamma$	$[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0]$

In the grid search approach, all possible combinations of hyper-parameters are tested by five-fold cross validation, and the combination that gives the highest score is selected for class prediction of the test dataset.

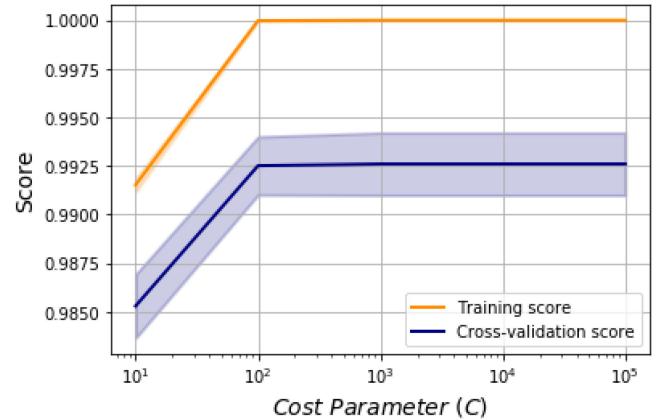


Fig. 12. Training score and cross-validation score versus  $C$ .  $C$  is 1000 where the training score has leveled off indicating an optimized value.

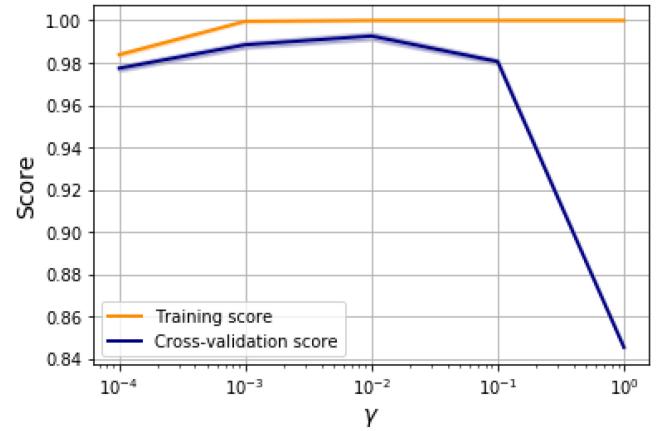


Fig. 13. Training score and cross-validation score versus  $\gamma$ .  $\gamma$  is 0.01 where the training score has leveled off indicating an optimized value.

The RF classifier is an ensemble machine learning technique that builds a number of decision tree classifiers on various random sub-samples of the training dataset, and then selects its output from the mean prediction of the individual decision trees. RF utilizes the idea of bagging [44] with a random subset of features [45]. We used Gini impurity criterion to split the nodes of decision trees. Hyper-parameters like *number of trees* and *depth of tree* are chosen using grid search five-fold cross validation. For this study, the RF classifier was optimized for number of trees (400), maximum tree depth (60), and maximum features (50). The range of hyper-parameters for grid search for RF are shown in Table II.

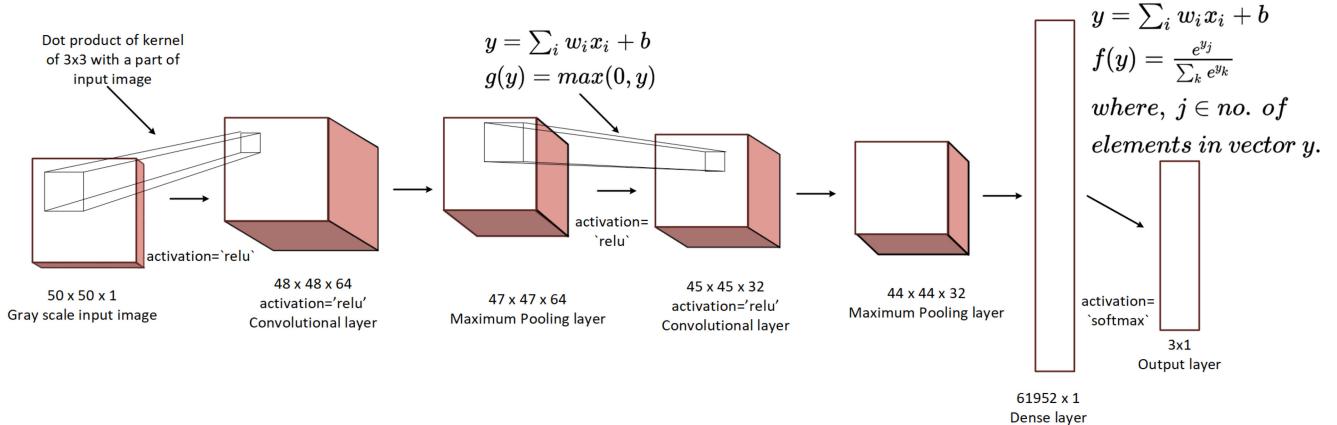


Fig. 14. Architecture of the CNN. Input data is an image of dimensions  $50 \times 50 \times 1$ . The feature maps after the first and second convolutional layer have dimensions of  $48 \times 48 \times 64$  and  $45 \times 45 \times 32$ , respectively, formed by convolving sixty four  $3 \times 3$  kernels. The maximum pooling layers are of dimensions  $47 \times 47 \times 64$  and  $44 \times 44 \times 32$  using a  $2 \times 2$  filter. The dimensions of the dense layer are  $61952 \times 1$ , and last layer is the output layer for the three classes.  $g(y)$  and  $f(y)$  are the ReLU and Softmax activation functions.

TABLE II  
VALUES OF RF HYPER-PARAMETERS USED IN THE GRID SEARCH METHOD

Parameters	Search Vector
Max. tree depth	[10, 20, 40, 60, 80, 100, None]
Number of trees	[50, 100, 200, 400, 600, 900, 1200]
Max. number of features	[10, 50, 100, 500, 1000, 2500]

Five-fold cross validation was applied on the training set to identify the best combination of parameters. Test data was used to predict the scores of the RF model on the best parameters.

CNN is a variant of deep neural network and lately, it is one of the most sought-after machine learning techniques used in the field of computer vision for classification [46]–[50]. We trained the CNN model on a system with an NVIDIA GP100GL (Tesla P100) GPU with 12 GB of memory. When compared with the training time on a system with CPU Intel(R) Xeon(R) E5-2450 @ 2.10 GHz processor, 24 GB memory, the GPU system was approximately 20 times faster. A typical CNN is made up of convolutional, pooling and dense layers, with input and output layers at the two extreme ends of the network. TensorFlow [51] and Keras [52] python libraries were used to train and custom build the CNN model shown in Fig. 14. We started building the model from a simple four layer (input, convolutional, dense, and output) model, to which we added complexities and filters until saturation for accuracy was achieved. The input image was read as a grayscale and thus dimension of the input layer was  $50 \times 50 \times 1$ . The first convolutional layer used 64 filters with a  $3 \times 3$  kernel, which convolves over the image and thus the output is a tensor of size  $48 \times 48 \times 64$ . The equation for the convolution operation on image  $I$  with filter  $K$  is given by

$$(I * K)_{i,j} = \sum_{m=1}^k \sum_{n=1}^k K_{(m,n)} \cdot I_{(i+m-1, j+n-1)} + b \quad (13)$$

where  $k$  is length and breadth of the filter,  $i$  and  $j$  are the indices of the image on which filter can convolve, and  $b$  is the bias. The activation function used in the model was rectified linear

TABLE III  
CONFUSION MATRIX FOR THE PREDICTED AND ACTUAL/MANUAL LABELS FOR THE TEST DATASET FOR THE SVM MODEL

		Predicted		
Actual		Corroded	Cracked	Good
		Corroded	Cracked	Good
Corroded	Corroded	85	0	0
Cracked	Cracked	0	21	4
Good	Good	0	0	600

unit (ReLU), which is expressed as

$$f\left(\sum_i w_i x_i + b\right) = \max\left(0, \left(\sum_i w_i x_i + b\right)\right) \quad (14)$$

where  $w_i$  and  $x_i$  are the weight and input values from the tensor in the previous layer that contribute to the cell index of tensor in current layer. The second layer is a maximum pooling layer for which a kernel of size  $2 \times 2$  is convolved over the convolutional layer to generate a new feature map of dimension  $47 \times 47 \times 64$ , which returns the maximum value for a kernel. Similarly, the next two layers are also combinations of convolutional and pooling layers. Then we flatten the output from the second maximum pooling layer to a vector of 61952 neurons. The final layer is an output layer for three classes. The weights of the networks are trained by minimizing an error function through back propagation using the stochastic gradient descent method.

#### D. Prediction and Results

To characterize the performance of each model, a testing set of data (held out from model training) was used to test classification prediction of each cell image. That prediction by the model was then compared with the manual supervised classification, and the set of score metrics (precision, recall, f-measure, and accuracy percentage) were calculated.

The classification results of all three models (SVM, RF, and CNN) are shown by confusion matrices in Tables III–V, respectively. Among the three models, we find CNN to outperform the

TABLE IV  
CONFUSION MATRIX FOR THE PREDICTED AND ACTUAL/MANUAL  
LABELS FOR THE TEST DATASET FOR THE RF MODEL

		Predicted		
Actual		Corroded	Cracked	Good
	Corroded	84	0	1
	Cracked	0	8	17
	Good	0	0	600

TABLE V  
CONFUSION MATRIX FOR THE PREDICTED AND ACTUAL/MANUAL  
LABELS FOR THE TEST DATASET FOR THE CNN MODEL

		Predicted		
Actual		Corroded	Cracked	Good
	Corroded	85	0	0
	Cracked	0	23	2
	Good	0	0	600

TABLE VI  
AVERAGE SCORE METRICS FOR CORRODED, CRACKED, AND  
GOOD CLASSES FOR SVM, RF, AND CNN MODELS

Model	Precision	Recall	F-measure	Accuracy
SVM	0.99	0.94	0.96	99.43%
RF	0.99	0.77	0.82	97.46%
CNN	0.99	0.97	0.98	99.71%

other two methods, with SVM performing second best, and RF showing the lowest scores. Across three different classes, good, and corroded cells were classified more accurately than cracked cells. This can be attributed to the disparity in the data with only 3.5% images belonging to the cracked category.

Metric values of precision, recall, and f-measure calculated in Table VI are the average values for the three different classes. We calculate the metrics using (6) iteratively for each class, considering one class as a positive and the other two classes as negative and then taking the mean values of all three iterations. From the table we observe that the RF model has a high accuracy of 97.46%, which is misleading because the RF model is specifically poor in classifying the majority of cracked cells correctly. To get an overall reflection of model behavior, we collectively look at all metrics in Table VI and find that the recall value of 77% better reflects the model performance.

In Fig. 15, we observe how growth of the training dataset improves model performance. As the field of PV degradation and modeling matures, greater volumes of data will further improve these metrics. Fig. 16 shows the performance of the CNN model as the number of training epoch increases. Model accuracy improves in the beginning and then stabilizes, meaning that the model has been trained for the given dataset and further improvement can be achieved by adding new datasets. Fig. 17 shows the two cell images which were wrongly classified as good by CNN, but were actually labeled as cracked. The image on the left has a very light crack, parallel, and near to the busbar. The right image has even smaller crack near edge of the cell, which caused it to be classified as good cell.

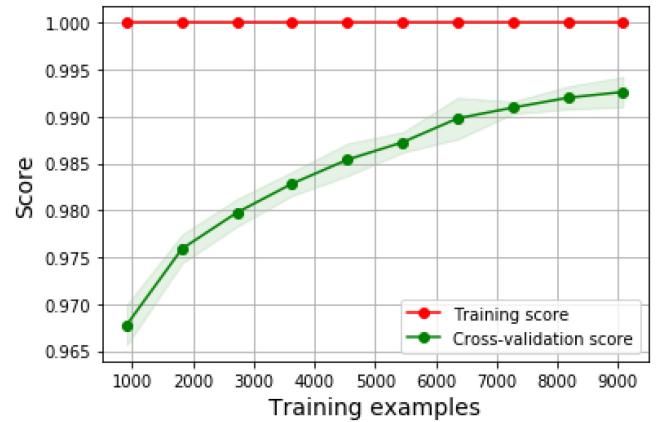


Fig. 15. Training score and accuracy score at the optimized  $C$  and  $\gamma$  values. Score converges with consistent training score indicating a validated model. For the training model, we randomly selected part of the training set for each of the five iterations for 1000, 2000, ..., 9000 data points, and the test set was used for validation results.

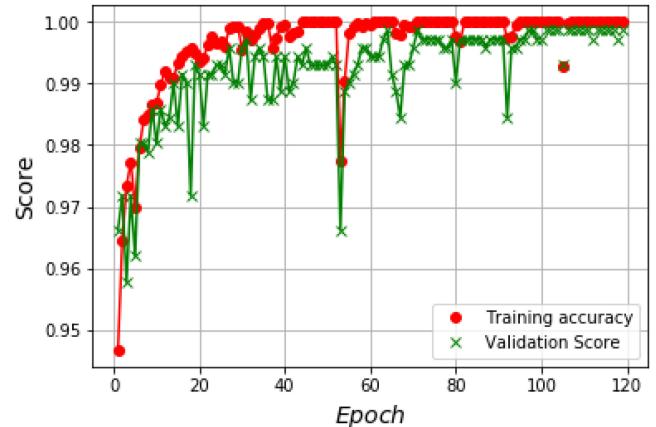


Fig. 16. Training and validation accuracy for the CNN model versus epoch. Training initially improves model results, until the curve flattens to show that the model has become stable and further training will not improve the performance for the given training dataset.

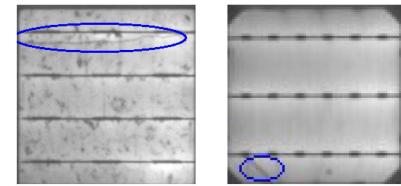


Fig. 17. Two cell images classified as good by CNN, but were labeled as cracked.

#### IV. DISCUSSION

EL image measurements of PV modules contain vast and diverse information about the condition of cells and performance of the module. For the PV industry, this type of data has the potential to yield great insight into the complexities of system

level degradation from mechanistic evidence. Additionally, reduced costs of imaging systems and data storage have drastically increased the volume of PV image data that is available for analysis. Even with this great potential, very little work has been done to create scalable image analysis pipelines, which are tailored for PV data. Feature classification and tracking of feature formation is also an important capability to be developed for PV system monitoring and lifetime performance analysis.

In this work, we have demonstrated a high-fidelity pipeline that can systematically process thousands of images in minutes. We have also developed algorithmic sorting of EL cell images on a small number of features to validate this method, providing the opportunity for other features to be added from larger datasets.

Intuitively, it is expected that the good/clean and corroded cells would classify with high accuracy. In both cases, the feature (or lack thereof) is well defined. For good/clean cells, there is simply no feature compared with the other two classes. And for corroded cells, all images are slight variations of the rest with defects localized to the busbar region where the only significant variation is the width of the corroded area. Conversely, crack formations on the surface of a cell are nonuniform. Each crack is unique with a particular propagation pattern that is a function of the specific heterogeneous structure of the individual wafer, as well as the external factors causing initiation and propagation. Cracks also exhibit variable geometries, including thickness and length. The issue of uniqueness can be resolved with large amounts of data, however with the cracked cells in our training dataset being only 3.5% of the 11 360 cells, there is no sufficient data to predict at higher scores using the current learning algorithms.

The accuracy results for SVM and CNN classification models of over 99% are improvement over the previous works and the other advancement is the classification of dataset into cracking and corrosion [14]. The data augmentation described in Section III-A has helped enhance the size of input training dataset to four folds, such that all the four images have equal chances of occurrence. In this work and the work presented in [14], CNN model has outperformed other algorithms, such as SVM or RF. To make models much more robust to variety in the data, and to improve accuracy for future unseen data, various combinations of training and testing datasets could be built, such as conducting testing and training on different brands. Herein, we have avoided partitioning the data by brand because of the low percentage of cracked cells overall, and that percentage being even fewer for some brands compared with others. With more data and advanced modeling, the classifier models could be trained to yield higher classification performances, particularly for cracked cells.

## V. CONCLUSION

In this work, we demonstrated an automated image processing pipeline used to process PV module-level EL images into single cell images with augmentation employed to increase the dataset size. Cell images were manually categorized into three subsets: good, cracked, and corroded. Using a supervised statistical learning approach, three modern classification algorithms

(SVM, RF, and CNN) were compared with four different metrics (accuracy, precision, recall, and f-measure) for each classification. Hyper-parameters were trained for SVM and RF using five-fold cross validation. For SVM, tuned values of  $C$  and  $\gamma$  parameters were calculated to be 1000 and 0.01, respectively. In RF modeling, hyper-parameters of maximum tree depth, number of trees, and maximum number of features were tuned final values of 60, 400, and 50, respectively. The CNN model was trained to a large number of epochs until the scores became stable and further training no longer improved the score. Five-fold cross validation also demonstrated that the models were stable. The results showed that CNN performs best at classifying cell EL images into corroded, cracked, and good classes based on all four score metrics.

## ACKNOWLEDGMENT

Research was performed at the SDLE Research Center. This work made use of the HPC Resource in the Core Facility for Advanced Research Computing at Case Western Reserve University.

## REFERENCES

- [1] D. C. Jordan, T. J. Silverman, J. H. Wohlgemuth, S. R. Kurtz, and K. T. VanSant, "Photovoltaic failure and degradation modes," *Progress Photovolt.: Res. Appl.*, vol. 25, no. 4, pp. 318–326, Apr. 2017. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/pip.2866/abstract>
- [2] DOE-EERE, "EERE success story—Researchers illuminate solar panel wear and tear to improve longevity," Mar. 2018. [Online]. Available: <https://www.energy.gov/eere/success-stories/articles/eere-success-story-researchers-illuminate-solar-panel-wear-and-tear>
- [3] U. Jahn *et al.*, "Review on IR and EL imaging for PV field applications," IEA-PVPS Task 13, Tech. Rep. IEA-PVPS T13-10:2018, May 2018. [Online]. Available: <http://www.iea-pvps.org/index.php?id=480>
- [4] M. Köntges *et al.*, "Review of failures of photovoltaic modules," International Energy Agency, Paris Cedex, France, Tech. Rep. IEA-PVPS T13-01:2014, 2014. [Online]. Available: [http://www.iea-pvps.org/fileadmin/dam/intranet/ExCo/IEA-PVPS\\_T13-01\\_2014\\_Review\\_of\\_Failures\\_of\\_Photovoltaic\\_Modules\\_Final.pdf](http://www.iea-pvps.org/fileadmin/dam/intranet/ExCo/IEA-PVPS_T13-01_2014_Review_of_Failures_of_Photovoltaic_Modules_Final.pdf)
- [5] D. Pysch, A. Mette, and S. Glunz, "A review and comparison of different methods to determine the series resistance of solar cells," *Sol. Energy Mater. Sol. Cells*, vol. 91, no. 18, pp. 1698–1706, Nov. 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0927024807002255>
- [6] T. M. Pletzer, J. I. van Möhlen, S. Rißland, O. Breitenstein, and J. Knoch, "Influence of cracks on the local current–voltage parameters of silicon solar cells," *Progress Photovolt.: Res. Appl.*, vol. 23, no. 4, pp. 428–436, Apr. 2015. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/pip.2443/abstract>
- [7] T. J. Peshek *et al.*, "Insights into metastability of photovoltaic materials at the mesoscale through massive  $I$ – $V$  analytics," *J. Vacuum Sci. Technol. B*, vol. 34, no. 5, p. 050801, Sep. 2016. [Online]. Available: <http://scitation.aip.org/content/avs/journal/jvstb/34/5/10.1116/1.4960628>
- [8] J. S. Fada *et al.*, "Democratizing an electroluminescence imaging apparatus and analytics project for widespread data acquisition in photovoltaic materials," *Rev. Scientific Instrum.*, vol. 87, no. 8, p. 085109, Aug. 2016. [Online]. Available: <http://scitation.aip.org/content/aip/journal/rsi/87/8/10.1063/1.4960180>
- [9] T. Trupke, J. Nyhus, and J. Haunschild, "Luminescence imaging for inline characterisation in silicon photovoltaics," *Physica Status Solidi (RRL)-Rapid Res. Lett.*, vol. 5, no. 4, pp. 131–137, 2011. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/pssr.201084028/full>
- [10] K. Ramspeck *et al.*, "Recombination current and series resistance imaging of solar cells by combined luminescence and lock-in thermography," *Appl. Phys. Lett.*, vol. 90, no. 15, p. 153502, 2007. [Online]. Available: <http://scitation.aip.org/content/aip/journal/apl/90/15/10.1063/1.2721138>

- [11] D. Hinken, K. Ramspeck, K. Bothe, B. Fischer, and R. Brendel, "Series resistance imaging of solar cells by voltage dependent electroluminescence," *Appl. Phys. Lett.*, vol. 91, no. 18, p. 182104, 2007. [Online]. Available: <http://scitation.aip.org/content/aip/journal/apl/91/18/10.1063/1.2804562>
- [12] T. W. Teo and M. Z. Abdullah, "Detection of oxygen precipitate dark rings in solar cell luminescence using gray level co-occurrence matrix," in *Proc. 9th Int. Conf. Robot. Vision Signal Process. Power Appl.*, 2017, pp. 307–312.
- [13] M. Alt *et al.*, "Electroluminescence imaging and automatic cell classification in mass production of silicon solar cells," in *Proc. IEEE 7th World Conf. Photovolt. Energy Convers.*, pp. 3298–3304, Jun. 2018.
- [14] S. Deitsch *et al.*, "Automatic classification of defective photovoltaic module cells in electroluminescence images," Jul. 2018, arXiv: 1807.02894. [Online]. Available: <http://arxiv.org/abs/1807.02894>
- [15] T. Fuyuki *et al.*, "Electroluminescence," in *Proc. 31st Photovolt. Specialists Conf., Lake Buena Vista, FL, USA*, 2005, pp. 1343–1345.
- [16] T. Potthoff, K. Bothe, U. Eitner, D. Hinken, and M. Koentges, "Detection of the voltage distribution in photovoltaic modules by electroluminescence imaging," *Progress Photovolt.: Res. Appl.*, vol. 18, no. 2, pp. 100–106, Mar. 2010. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/pip.941/abstract>
- [17] R. H. French *et al.*, "Degradation science: Mesoscopic evolution and temporal analytics of photovoltaic energy materials," *Current Opinion Solid State Mater. Sci.*, vol. 19, no. 4, pp. 212–226, Aug. 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1359028614000989>
- [18] Y. Hu *et al.*, "A nonrelational data warehouse for the analysis of field and laboratory data from multiple heterogeneous photovoltaic test sites," *IEEE J. Photovolt.*, vol. 7, no. 1, pp. 230–236, Jan. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7763779/>
- [19] *Terrestrial Photovoltaic (PV) Modules—Design Qualification and Type Approval—Part 2: Test Procedures*, IEC 61215-2:2016. [Online]. Available: <https://webstore.iec.ch/publication/24311>
- [20] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R* (Springer Texts in Statistics), 1st ed. New York, NY, USA: Springer, Aug. 2013. [Online]. Available: <http://www-bcf.usc.edu/~gareth/ISL/index.html>
- [21] J. S. Fada *et al.*, "Electroluminescent image processing and cell degradation type classification via computer vision and statistical learning methodologies," in *Proc. IEEE 44th Photovolt. Specialist Conf.*, Jun. 2017, pp. 3456–3461.
- [22] C. S. Fraser, "Digital camera self-calibration," *ISPRS J. Photogrammetry Remote Sens.*, vol. 52, no. 4, pp. 149–159, Aug. 1997. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924271697000051>
- [23] Python, Python.org. (2013). [Online]. Available: <https://www.python.org/>
- [24] NumPy, "NumPy — numpy." (2014). [Online]. Available: <http://www.numpy.org/>
- [25] SciPy, SciPy.org — SciPy.org. (2014). [Online]. Available: <http://www.scipy.org/>
- [26] S. van der Walt *et al.*, "Scikit-image: Image processing in Python," *PeerJ*, vol. 2, p. e453, Jun. 2014. [Online]. Available: <https://peerj.com/articles/453>
- [27] Scikit-image. [Online]. Available: <https://pypi.org/project/scikit-image/>
- [28] OpenCV library. [Online]. Available: <https://opencv.org/>
- [29] K. Pulli, A. Baksheev, K. Konyakov, and V. Eruhimov, "Real-time computer vision with OpenCV," *Commun. ACM*, vol. 55, no. 6, pp. 61–69, Jun. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2184319.2184337>
- [30] A. Wang, T. Qiu, and L. Shao, "A simple method of radial distortion correction with centre of distortion estimation," *J. Math. Imag. Vision*, vol. 35, no. 3, pp. 165–172, Nov. 2009. [Online]. Available: <http://link.springer.com/10.1007/s10851-009-0162-1>
- [31] P. Drap and J. Lefèvre, "An exact formula for calculating inverse radial lens distortions," *Sensors*, vol. 16, no. 6, p. 807, Jun. 2016. [Online]. Available: <http://www.mdpi.com/1424-8220/16/6/807>
- [32] S. Deitsch, C. Buerhop-Lutz, A. Maier, F. Gallwitz, and C. Riess, "Segmentation of photovoltaic module cells in electroluminescence images," Jun. 2018, arXiv: 1806.06530. [Online]. Available: <http://arxiv.org/abs/1806.06530>
- [33] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications* (The Springer International Series in Engineering and Computer Science), vol. 84. Springer-Verlag, Mar. 2013.
- [34] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1001–1013, Jul. 2014.
- [35] M. A. Jayaram and H. Fleyeh, "Convex hulls in image processing : A scoping review," *Amer. J. Intell. Syst.*, vol. 6, no. 2, pp. 48–58, 2016. [Online]. Available: <http://urn.kb.se/resolve?urn=urn:nbn:se:du-21491>
- [36] A. Criminisi, I. Reid, and A. Zisserman, "A plane measuring device," *Image Vision Comput.*, vol. 17, no. 8, pp. 625–634, Jun. 1999. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0262885698001838>
- [37] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," in *Linear Algebra (Handbook for Automatic Computation)*, J. H. Wilkinson, C. Reinsch, and F. L. Bauer, Eds. Berlin, Heidelberg, Germany: Springer, 1971, pp. 134–151. [Online]. Available: [https://doi.org/10.1007/978-3-662-39778-7\\_10](https://doi.org/10.1007/978-3-662-39778-7_10)
- [38] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning* (Springer Series in Statistics). New York, NY, USA: Springer, 2009. [Online]. Available: <http://link.springer.com/10.1007/978-0-387-84858-7>
- [39] S. K. Thompson, *Sampling* (Wiley Series in Probability and Statistics), 3rd ed. Hoboken, NJ, USA: Wiley, 2012.
- [40] A. Gramfort, G. Varoquaux, O. Grisel, and R. RV, "Sklearn: StratifiedShuffleSplit().," [Online]. Available: [http://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.StratifiedShuffleSplit.html](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html)
- [41] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [42] A. Gramfor *et al.*, "Sklearn: accuracy\_score()." [Online]. Available: [http://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy\\_score.html](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html)
- [43] C. Cortes and V. Vapnik, "Support-vector networks," in *Mach. Learn.*, vol. 20, pp. 273–297, Sep. 1995. [Online]. Available: <http://dx.doi.org/10.1023/A:1022627411411>
- [44] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
- [45] Leo Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [47] P. Simard, D. Steinkraus, and J. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. 7th Int. Conf. Document Anal. Recognit. Proc.*, vol. 1, Edinburgh, U.K.: IEEE Comput. Soc., 2003, pp. 958–963. [Online]. Available: <http://ieeexplore.ieee.org/document/1227801/>
- [48] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.
- [49] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—A review," *Pattern Recognit.*, vol. 35, no. 10, pp. 2279–2301, Oct. 2002. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320301001789>
- [50] W. Liu *et al.*, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231216315533>
- [51] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," Mar. 2016, arXiv: 1603.04467. [Online]. Available: <http://arxiv.org/abs/1603.04467>
- [52] F. Chollet *et al.*, Keras. (2015). [Online]. Available: <https://github.com/fchollet/keras>



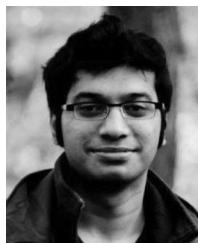
**Ahmad Maroof Karimi** (GSM'19) received the B.S. degree in electrical engineering from Aligarh Muslim University, Aligarh, India, in 2011, the M.S. degree in computer science from the University of Toledo, Toledo, OH, USA, in 2016. He is currently working toward the Ph.D. degree in computer science from the Case Western Reserve University, Cleveland (CWRU), OH, USA.

He joined the Solar Durability Lifetime Extension Research Center, CWRU, in 2017, as a Research Assistant. His research focuses on the application of machine learning, pattern recognition and data mining in the field of energy applications. His research interests include working on large-scale distributed computing cluster for real-time machine learning problems.



**Justin S. Fada** received the B.S. degree with dual majors in mechanical and aerospace engineering in 2017 and the M.S. degree in mechanical engineering from Case Western Reserve University, Cleveland, OH, USA, in 2018.

He is currently with the National Aeronautics and Space Administration, Washington, DC, USA, as an Aerospace Engineer. He joined the Solar Durability and Lifetime Extension Research Center in 2014 as an Undergraduate Research, working on various projects related to the degradation of devices and materials subject to natural and induced environments. He developed imaging systems and analytic methods for understanding how PV devices degrade.



**Mohammad A. Hossain** received the B.Sc. degree in mechanical engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2009, and the M.Sc. and Ph.D. degrees in mechanical engineering from Case Western Reserve University, Cleveland, OH, USA, under Professor Roger H. French and Professor Alexis R. Abramson in 2014 and 2018, respectively.

He is a Senior Data Scientist at Hanwha Q CELLS USA, Inc. Prior to joining Hanwha Q CELLS, he was a Data Scientist with Cisco Systems, Inc., for 2 years. His current research interests include PV analytics, Building energy efficiency and optimization, data mining, time series analytics, prediction modeling, and machine learning.

**Shuying Yang**, photograph and biography not available at the time of publication.

**Timothy J. Peshek** received the Ph.D. degree in physics from Case Western Reserve University, Cleveland, OH, USA, in 2008.

He was a Postdoctoral Scholar with Arizona State University and the National Renewable Energy Laboratory. He is currently Senior Research Electrical Engineer with NASA's John Glenn Research Center performing research on space photovoltaics. Prior to joining NASA, he was a Research Assistant Professor of Materials Science and Engineering with Case Western Reserve University.



**Jennifer L. Braid** (M'18) received the B.S. degrees in physics and mathematics from Ohio University, and the M.S. degree in physics from the Ohio State University, and the Ph.D. degree in applied physics from Colorado School of Mines, Golden, CO, USA, in 2016.

She conducted the Ph.D. research in organic photovoltaics at the National Renewable Energy Laboratory. In 2016, she joined the SDLE Research Center, Case Western Reserve University, as a Postdoctoral Researcher. In this position, she has led several projects investigating the durability and active degradation mechanisms in silicon cell technologies and commercially viable module materials. Her research focuses on multi-level, multi-scale study integration for mechanistic understanding and discovery in electronic devices.



**Roger H. French** (M'08) received the Ph.D. degree in materials science from Massachusetts Institute of Technology, Cambridge, MA, USA, under Prof. R. L. Coble.

In 1985, he joined DuPont Co., Central Research, and, in 1996, he joined the University of Pennsylvania Adjunct Professor of Materials Science. He has been the Kyocera Professor with the Materials Science Department with secondary appointments in macromolecular science, computer and data sciences, biomedical engineering, and physics at Case Western Reserve University (CWRU), Cleveland, OH, USA, since August 2010. He is the Director of the SDLE Research Center, and the Faculty Director of the CWRU Applied Data Science Program. His research focuses on lifetime and degradation science, and reliability, of photovoltaic materials, modules, and power plants, using data science and high-performance/distributed computing approaches.