Feature Extraction, Supervised and Unsupervised Machine Learning Classification of PV Cell Electroluminescence Images

Ahmad Maroof Karimi *‡, Justin S. Fada*, JiQi Liu*†, Jennifer L. Braid*†,
Mehmet Koyutürk‡, Roger H. French*†

* SDLE Research Center, Case Western Reserve University, Ohio
†Department of Material Science Engineering, Case Western Reserve University, Ohio
‡Department of Electrical Engineering and Computer Science, Case Western Reserve University, Ohio
Email: ahmadmaroof.karimi@case.edu

***Abstract*—Lifetime performance and degradation analysis of laboratory and field deployed PV modules is paramount to the continued success of solar energy. Image characterization techniques capture spatially resolved macroscopic manifestations of microscopic mechanistic behavior. Automated data processing and analytics allow for a large-scale systematic study of PV module health. In this study, degradation features seen in periodic EL images taken during test-to-failure damp-heat, thermal cycling, ultra-violet irradiance, and dynamic mechanical loading accelerated exposures are extracted and classified using supervised and unsupervised methods. Image corrections, including planar indexing to align module images, are applied. On extracted cell images, degradation states such as busbar corrosion, cracking, wafer edge darkening, and between-busbar dark spots can be studied in comparison to new cells using supervised and unsupervised machine learning. The systematic feature groupings provide a scalable method without bias to quantitatively monitor the degradation of laboratory and commercial systems alike. The evolution of these degradation features through varied exposure conditions provides insight into mechanisms causing degradation in field deployed modules. The supervised algorithms used in this application are Convolutional Neural Networks (CNN) and Support Vector Machines (SVM). With the increase in data and diversity of features, unsupervised learning can be employed to find relations between inherent image properties. Feature extraction techniques help identify intrinsic geometric patterns formed inthe images due to degradation. Principal component analysis is then applied to the extracted set of features to filter the most relevant components from the set, which are then passed to an agglomerative hierarchical clustering algorithm. Google's Tensorflow library was utilized to enhance the computational efficiency of the CNN model by providing GPU-based parallel matrix operations. Using supervised methods on 5 features an accuracy greater than 98% was achieved. For unsupervised clustering, the classification was done into two clusters of degraded and non-degraded cells with 66% coherence.**

***Index Terms*—computer vision, electroluminescence imaging, feature extraction, machine learning, PCA, supervised classification, unsupervised clustering,**

I. Introduction

The power output from a photovoltaic (PV) module is the most important metric for the measurement of performance of PV. The expected lifetime of a field deployed photovoltaic (PV) module is over two decades, but the power output decreases gradually over a system's lifetime due to multiple degradation factors.[1], [2]. Factors that affect the long term stability of PV modules include mechanical stresses during the transportation and field installation which can lead to cell cracking [3]. Environmental variables such as moisture, temperature and it's fluctuation, and wind contribute to degradation features which can be expressed in module characterization data.

Different strategies are used for determining the degradation in PV panels. Particular insight into mechanistic behavior can be captured using imaging systems such as electroluminescence (EL), which provides spatially resolvable information through the front side of the module [4]. Supervised and unsupervised machine learning approach can be employed in classification of features in cell EL images. *Supervised machine learning:* Application in which the dataset consists of feature vectors as well as target vectors/labels and the goal of an algorithm is try to learn a target function such that every feature vector is assigned a correct predefined label. *Unsupervised machine learning:* When the dataset consists of feature vectors but are devoid of labels, the aim of an algorithm is to identify groups/clusters with examples of similar characteristics.

In our previous work [5], we have introduced advanced image processing techniques to analyze EL images of solar modules and cells. Supervised machine learning algorithms were used to classify cells as cracked, busbar corroded or in a new/featureless state. In this work, we have broadened the supervised classification algorithms to study more cell degradation features. We also present feature extraction and unsupervised clustering technique to correlate intrinsic patterns existing in the images with supervised labels. The idea behind unsupervised learning is to reduce the amount of time needed for human labeling of thousands of cell images into different categories. In addition, human judgment in attaching labels introduces subjectiveness in dataset groupings. Growth of patterns and characteristics of the cell images due to degradation are leveraged for feature extraction. Clustering is done based on the distance matrix calculated from the extracted features of cell images. Some of the new patterns observed in the images include darkening of the edges and cell areas between busbars.

978-1-5386-8529-7/18/$31.00 ©2018 IEEE

We provide an overview of the experimental study, data collection and computational tasks like image processing, feature extraction and machine learning algorithms in section II. In section III, we present the results and discussion of various classification algorithms. The final section contains conclusions of the research.

II. Methodology

A. Test-to-failure Study

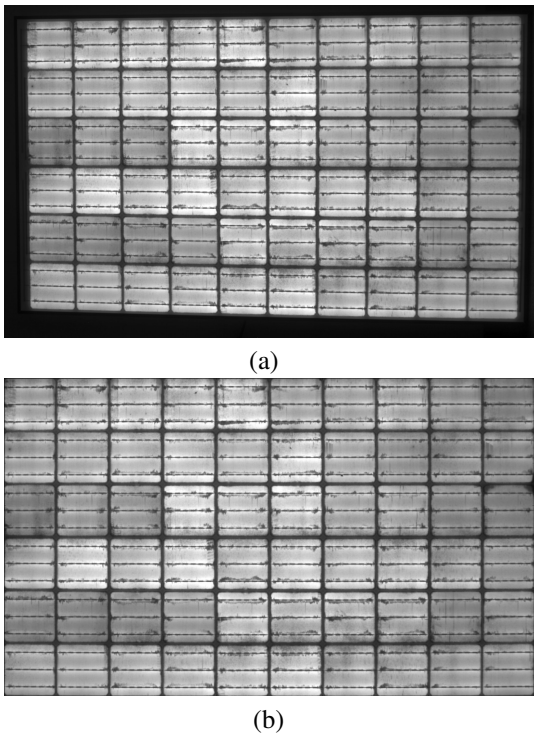
90 full-size commercial modules spanning 5 brands were split into 6 groups equally representing each brand. Each of these 6 groups were exposed to a different IEC indoor accelerated test [6]. These included damp-heat, thermal-cycling, ultra-violet irradiance, dynamic mechanical loading, potential induced damage (PID) +1000 V, and PID -1000 V. For the work outlined herein, the PID biased modules were not used as they did not contain enough degraded cells to be useful to the analysis.

Damp-heat modules underwent 4200 hours of total exposure with current-voltage ($I-V$) curves and electroluminescence (EL) images being measured every 500 hours until 3000 hours and every 300 hours after 3000 hours. Thermal-cycling exposed modules underwent 1000 cycles in total with $I-V$ and EL measurements were taken every 200 cycles until 600 cycles and every 100 cycles after 600 cycles. Ultra-violet (UV) exposed modules underwent 90 kWh of irradiance with measurements taken every 30 kWh. And lastly, the dynamic mechanical loading tests applied 2 rounds of dynamically actuated loading. These tests induced a myriad of degradation features to be analyzed.

The OpenCV [7] Python library was used extensively for image processing. Some of the functions for machine learning applications were used from the scikit-learn and scipy Python libraries [8], [9]. We implemented Google's TensorFlow library with Graphical Processing Unit (GPU) support to increase computational efficiency over our previous work [10][5]. TensorFlow provides an abstraction over learning classifiers and is an efficient image analysis library for a large dataset having numerous feature vectors, especially for computationally expensive matrix operations such as Deep Neural Networks (DNN).

B. Image Pre-processing

A sample of a raw EL image of a PV module is shown in Figure 1(a). As outlined in previous work [5], an automated image processing pipeline has been developed in Python and enhanced further to reduce noise and planar index the module in the image (Figure 1(b)). The processed image allows extraction of individual cells from the module image [5]. Figure 2 outlines the salient stages in the processing pipeline. The original images are filtered using a median filter. A custom algorithm built using the open-source language Python is then employed to find the edges of the module in the EL image and then a perspective transform of the module into the image plane with consistent dimensions for streamlined analysis. The cell images in Figure 3, which are studied herein, are then sliced from the planar indexed images [5]. Cell images were rotated by 180^o as a part of image augmentation process for increasing the training set.



(a)

(b)

Fig. 1. Figure (a) is a raw EL image captured by electroluminescence imaging method, and (b) is the processed image post filtering and planar indexing.

C. Supervised Classification

The first step in training a supervised learning algorithm is to have a well defined, large, training set. After assessing and identifying the characteristics, we decided to label dataset into five feature groups: cracked, busbar corroded, edge darkened, between-busbar darkened and new/featureless, which can be seen with their representative images in Figure 3. Sorting resulted in 6264 images across all brands and 4 accelerated exposure conditions.

Two supervised machine learning algorithms, Convolutional Neural Networks (CNN) and Support Vector Machine (SVM), were compared to determine which was best for PV cell dataset.[11] Stochastic gradient descent was used for convergence of the model, and a cross-validation technique was used to increase confidence in the results.

1) Convolutional Neural Network: CNN is a feed-forward DNN classifier, and is the state-of-the-art training model architecture particularly for image classification problems. In this method, a series of filters are applied to the raw images to extract and learn high level intrinsic features, which the model can then use for classification purposes [12]. CNNs typically have three sets of layers:

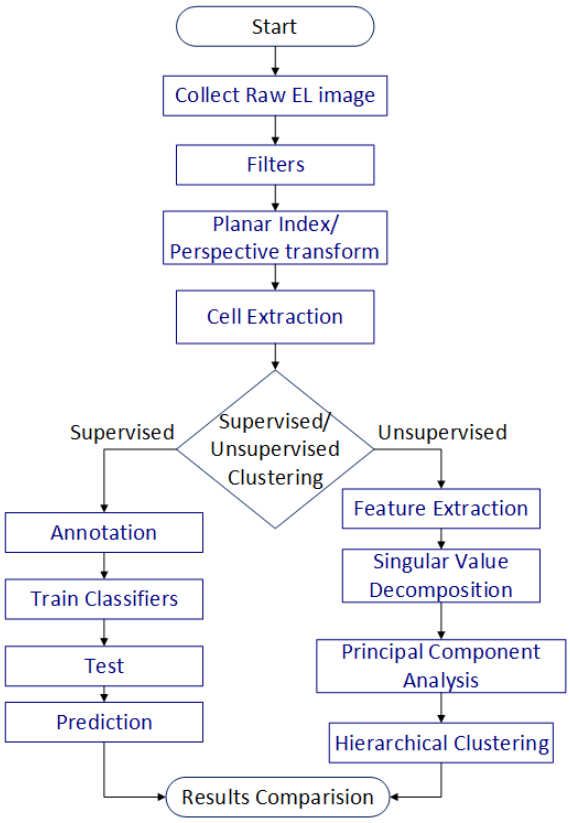


Fig. 2. Flowchart showing various stages in our image processing pipeline.

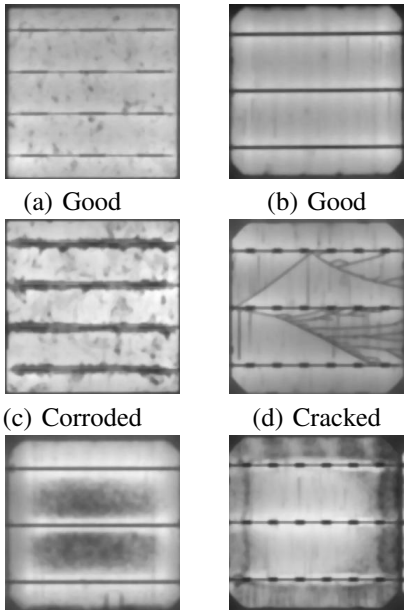


Fig. 3. These 6 cells are examples of the 5 cell types used for this classification model: (a-b) good, (c) busbar corrosion, (d) cracked, (e) between busbar darkening and (f) cell edge darkening.

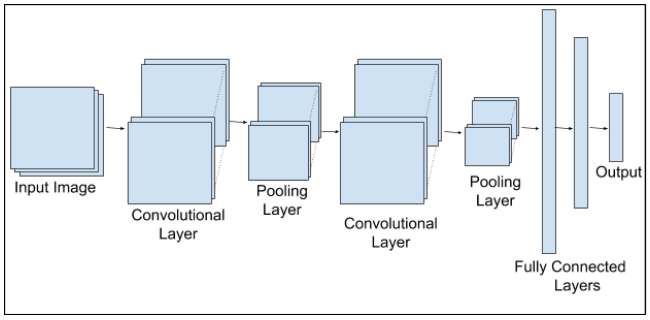


Fig. 4. General architecture of convolutional neural network.

1) Convolutional layers: A specified number of convolutional filters are applied to the images. Each filter is a kernel of a typical shape of 5x5 grid pixels, which convolves across an image. For each subregion the kernel covers, the layer performs a set of mathematical operations to produce a single value in the output feature map. A Leaky-ReLU (Rectified Linear Unit) activation function is applied on the output value to introduce non-linearity into the model.
2) Pooling layers: Placed between the convolutional layers, the purpose of pooling layer is to reduce the spatial dimension and avoid over-fitting. The most common type of pooling methods are max pooling and average pooling, which takes the maximum or the average value respectively of the pixels covered by the kernel.
3) Fully Connected (FC) layers: Also known as dense layers, are similar to regular neural network where each input node is connected to each output node. These final learning layers map extracted features to visual output. FC layers help correlate features across a complete image.

In Figure 4, we have shown a general architecture of a CNN. The first layer is a 3-channel input image, then any number of inner layers can be added into the model depending on the classification problem, with care taken to avoid over-fitting in the model. The output layer has a size equal to the number of classes in the dataset.

2) Support Vector Machine: SVM classifiers are based on the decision function that divides the hyperspace into two classes by a hyperplane [13] given by an equation $wx + b = 0$ where $w \epsilon \mathbb{R}^n$, $b \epsilon \mathbb{R}$ and n is the number of feature vectors. The two-class classifier is extended to a multiclass classifier using one-versus-rest classifier, and for a datum we select a class which gives the largest margin. There are many kinds of kernel substitution methods in SVM, here we have used the radial basis function (RBF) kernel. The regularization parameter in a kernel function lets user avoid over-fitting, kernel function takes low dimensional input space and transforms the input to higher dimensional space making separable boundaries,and kernel method also eases the calculations by simplifying the optimization problem in a higher dimension space.

D. Feature Extraction

Feature extraction is a process by which hidden patterns are extracted by analyzing neighboring pixels in an image. In this step, cell images of 250x250 pixels, examples shown in Figure 3, were taken as input. Then fourteen Haralick features[14] were calculated from the images, in four different directions called $0^o, 45^o, 90^o, 135^o$ as shown in Figure 5. These features give the details of textural patterns in the images. In Table I, all the extracted features are listed with their description [15].

TABLE I
FEATURES EXTRACTED FROM THE PV CELL IMAGES

Features	Description
Energy/ Angular Second Moment (ASM), $\sum_{i,j}^{N_g} p(i,j)^2$ N_g is the no. of gray scale levels	ASM is also called energy, it's value is 1 for constant image & range [0,1]. $p(i,j)$ is $(i,j)^{th}$ value in a normalized gray-tone
Contrast, $\sum_{i,j}^{N_g} (i-j)^2 p(i,j)$	Measure of contrast in intensity between adjacent pixels
Correlation, $\sum_{i,j} \frac{(i.j)p(i,j)-\mu_x\mu_y}{\sigma_x\sigma_y}$ μ and σ are mean and std. deviation of p_x p_y	Measure of how correlated a pixel to it's neighbor in an image. p_x, p_y are marginal probabilities
Sum of square, $\sum_{i,j}(i-\mu)^2 p(i,j)$	squared distance from the mean pixel intensity
Inverse Difference Moment $\sum_{i,j} \frac{p(i,j)}{1+(i-j)^2}$	Also called Homogeneity, measures values by the inverse of the contrast weight
Sum Average, $\sum_{k=2}^{2N_g} kp_{(i+j)}(k)$	sum of the average values of whole image
Sum Variance, $\sum_{k=2}^{2N}(k-\mu_{x+y})^2 p_{(x+y)}(k)$	sum of the variance values of whole image. N is number of distinct gray level
Entropy, $-\sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j) log P(i,j)$	measure of randomness that can be used to characterize the texture of the input image.
Sum Entropy, $-\sum_{k=2}^{2n} p_{(x+y)}(k) log(p_{x+y}(k))$	sum of the entropy.
Difference Variance, $\sum_{k=2}^{2n}(k-\mu_{x-y})^2 p_{(x-y)}(k)$	Difference between the variance.
Difference Entropy, $-\sum_{k=2}^{2n} p_{(x-y)}(k) log(p_{x-y}(k))$	Difference between the entropy.
Information Measure of Correlation I, $\frac{HXY-HXY1}{max(HX,HY)}$	HX and HY are entropies of $P_x and P_y$ $HXY = -\sum_i \sum_j p(i,j) log(p(i,j))$ $HXY1 = -\sum_i \sum_j p(i,j) log(p_x(i)p_y(j))$ $HXY2 = -\sum_i \sum_j p_x(i)p_y(j) log(p_x(i)p_y(j))$
Information Measure of Correlation II, $(1-exp[-2.0(HXY2-HXY)])^{\frac{1}{2}}$	
Maximum Correlation coefficient $Q(i,j) = \sum \frac{p(i,k)p(j,k)}{p_x(i)p_y(k)}$	Method for identifying shape or numerical parameter for probability distribution

After deriving features from the images, singular value decomposition (SVD) [16] is applied on the extracted feature matrix to get the most prominent principal component (PC) vectors in a direction that explains the variability most in the dataset. To find the most dominant set of PCs, we use Equation 1 to get the three matrices.

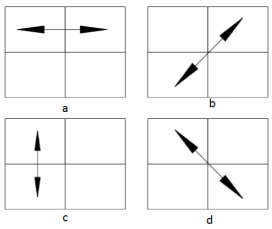


Fig. 5. Adjacent pixels for the calculation of haralick features in four directions a) 0, b) 45 c) 90 and d) 135 degrees of orientation.

$$U,\ D,\ V^T = SVD(X) \tag{1}$$

where U is a set of left singular vectors, V is a set of right singular vectors and V^T is a transpose of V, D is a diagonal matrix whose singular values are in a descending order with largest at the top of matrix and X is the features dataset on which the decomposition is applied. The singular values in diagonal matrix D give the contribution of each PCs in explaining the variability of the dataset.

In Principal Component Analysis (PCA)[17], 14 PCs are calculated from 14 Haralick features by taking the product of matrices U^T and X as given in Equation 2.

$$Z = U^T X \tag{2}$$

where Z is a matrix in which columns are principal component vectors. PCs are vectors in a domain different from the original feature space and are in the direction of maximum variation in the dataset. When X is represented by PCs, it helps to minimize noise and makes it easier to observe the broader structure in the dataset. We found PC1 and PC2 were the two most dominant components from the 14 PCs calculated in Equation 2.

E. Unsupervised Algorithms

Unsupervised machine learning was applied to the Haralick features extracted in the above section. The agglomerative hierarchical clustering [18] algorithm was applied to cluster all degradation types vs. good/featureless cells as shown in Figure 3. The benefit of a hierarchical clustering over other algorithms is that we don't need to know the number of groups available in the datasets prior to running the algorithm. Important steps in the algorithm are:

- All the data points in the algorithm are assigned to their own clusters, such that for N points there will be N number of clusters.
- Calculate the similarity measure between all the clusters from above step.
- Merge the two closest pairs, so that total number cluster reduces by 1.
- Repeat steps 2 & 3 until all data points belong to one cluster.

In this algorithm, the dissimilarity measure was a Euclidean distance metric between the feature vectors.

III. Results and Discussion

The results obtained from two supervised classification algorithms and agglomerative unsupervised clustering are presented below.

A. Supervised

For supervised methods, five-fold cross validation was used to confirm the stability of the algorithms. 80% of the data was used for training and the remaining 20% was used for validation. For the RBF kernel function in SVM, grid search is applied on the regularization parameter (C) and gamma value (γ). The optimum value of the model was chosen to be 50 for C and $1e^{-5}$ for the *gamma*.

In CNN modeling, two convolutional layers, three pooling layers and three fully connected layers were chosen and the model is shown in Figure 6. The first layer is the 2D convolutional layer which has 32 filters, so the output from the first layer is an object of shape 250x250x32. The second layer is a pooling layer with a filter kernel of size 5x5 with stride (1,1), meaning that in each step, the kernel filter will convolve 1 pixel from its original position in the horizontal or vertical direction. We again add a similar pooling layer to reduce the spatial dimension to 63x63x32. The next layer is a convolutional layer with output of 128 channels followed by a pooling layer. In the end, there are two fully connected layers and the output result layer.

The accuracy and standard deviation of the results for the above two models are shown in Table II. The other performance measures[19] of the two models are shown in Table III. The three measures are defined as follows: *Precision* is defined as fraction of the relevant class predicted correct.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall is proportion of the actual relevant class identified correctly.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

F-measure combines precision and recall and is the harmonic mean of two.

$$F - measure = \frac{2(Precision * Recall)}{Precision + Recall}$$

TABLE II
Accuracy of the 2 tested supervised classifier models

	SVM	CNN
Mean Accuracy	98.95%	98.24 %
Standard Deviation	0.67	1.8

TABLE III
Performance measures of two tested supervised models

	Precision	Recall	F-measure
SVM	0.97	0.99	0.98
CNN	0.96	0.98	0.97

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_11 (Conv2D)           (None, 250, 250, 32)      320
_________________________________________________________________
leaky_re_lu_27 (LeakyReLU)   (None, 250, 250, 32)      0
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 125, 125, 32)      0
_________________________________________________________________
leaky_re_lu_28 (LeakyReLU)   (None, 125, 125, 32)      0
_________________________________________________________________
max_pooling2d_17 (MaxPooling (None, 63, 63, 32)        0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 63, 63, 128)       102528
_________________________________________________________________
leaky_re_lu_29 (LeakyReLU)   (None, 63, 63, 128)       0
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 32, 32, 128)       0
_________________________________________________________________
flatten_6 (Flatten)          (None, 131072)            0
_________________________________________________________________
dense_17 (Dense)             (None, 256)               33554688
_________________________________________________________________
leaky_re_lu_30 (LeakyReLU)   (None, 256)               0
_________________________________________________________________
dense_18 (Dense)             (None, 128)               32896
_________________________________________________________________
leaky_re_lu_31 (LeakyReLU)   (None, 128)               0
_________________________________________________________________
dense_19 (Dense)             (None, 4)                 516
=================================================================
Total params: 33,690,948
Trainable params: 33,690,948
Non-trainable params: 0
```

Fig. 6. CNN model applied for classification.

B. Unsupervised

In unsupervised clustering, features are extracted from the cell images in four different orientations of 0^o, 45^o, 90^o, and 135^o as shown in Figure 5. The means of all fourteen features in four orientations were used for SVD decomposition and PCA feature calculation. Eigenvalues from the matrix D in Equation 1 are plotted in Figure 7. We observe from the large eigenvalues of the first two PCs that the maximum variability of the data is captured between them.

Thus, the first two principal features were used for clustering the dataset. A threshold of the Euclidean distance was considered as a similarity measure to cluster the datasets. In Figure 8, we have shown a scatter plot of PC1 vs PC2. Each data point represents a cell image extracted from the PV module, the points are colored into two groups of degraded and good/non-degraded. We can also observe two clusters in the plot reflecting the degraded and good/non-degraded cells. It suggests that PC1 and PC2 are the good representation of the 14 features extracted earlier. Hierarchical clustering coherence is calculated to be 66%, i.e. 66% of cell images were clustered in the group that was labeled same as the image. A perfect cluster would haves a coherence value of 1, which means all of the data in the cluster belongs to the

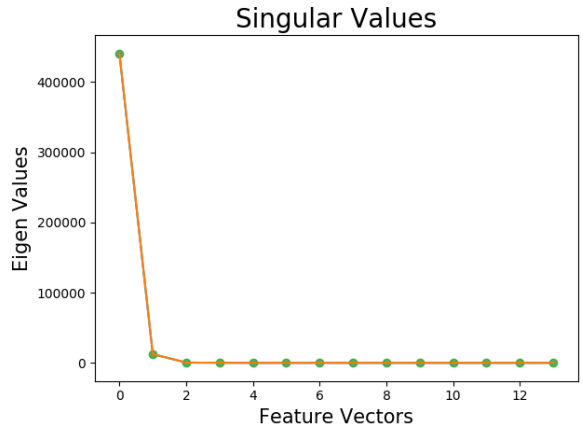


Fig. 7. Eigen values plot for all 14 feature vectors.

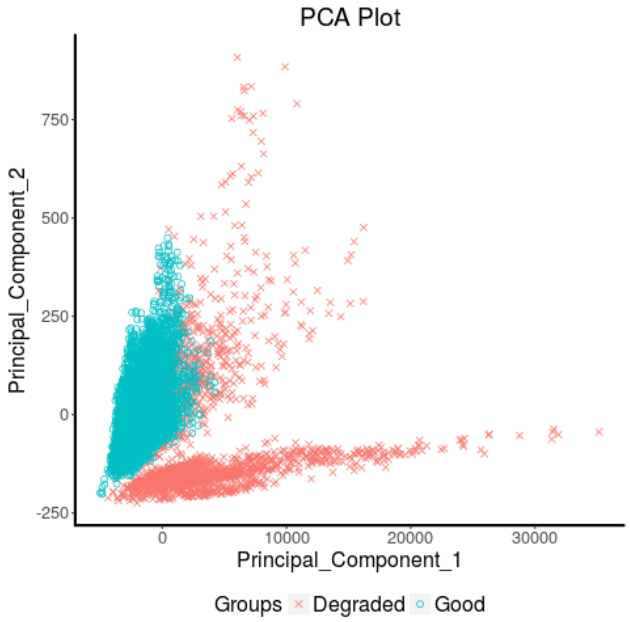


Fig. 8. PCA showing scatter plot of principal component 1 vs principal component 2.

same label.

IV. Conclusion

Supervised machine learning classification algorithm was trained to classify five different degradation types found in PV cells with SVM and CNN using TensorFlow as the classifiers. Tensorflow implementation with Graphical Processing Unit (GPU) efficiently reduced the computations time of CNN from hours to minutes. Identifying a good CNN model is not straightforward, multiple permutations and combinations of layer need to be tested before deciding on the model that gives the good result on both training and test dataset. To verify the robustness of the algorithm, we performed 5-fold cross-validation and observed that the standard deviation of all the results is near the mean values. We explained the feature extraction method to identify intrinsic patterns in the images, which were helpful in clustering. PCA method was used to filter the most dominant features from the Haralick features set. We also implemented an unsupervised algorithm to cluster images into degraded and non-degraded groups. Results show promising values for application on larger datasets which can be expanded to more diverse degradation features in the future.

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy Office for Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement Number DE-EE0007140. We are also grateful to the Rider High-Performance Computing Resource infrastructure in the Core Facility for Advanced Research Computing at Case Western Reserve University.

References

[1] V. Sharma and S. S. Chandel, "Performance and degradation analysis for long term reliability of solar photovoltaic systems: A review," *Renewable and Sustainable Energy Reviews*, vol. 27, pp. 753–767, Nov. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032113004917

[2] C. R. Osterwald, A. Anderberg, S. Rummel, and L. Ottoson, "Degradation analysis of weathered crystalline-silicon PV modules," in *Conference Record of the Twenty-Ninth IEEE Photovoltaic Specialists Conference, 2002.*, May 2002, pp. 1392–1395.

[3] U. Eitner, S. Kajari-Schrder, M. Kntges, and H. Altenbach, "Thermal Stress and Strain of Solar Cells in Photovoltaic Modules," in *Shell-like Structures: Non-classical Theories and Applications*, H. Altenbach and V. A. Eremeyev, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 453–468, dOI: 10.1007/978-3-642-21855-2_29.

[4] Fada et. al, "Democratizing an electroluminescence imaging apparatus and analytics project for widespread data acquisition in photovoltaic materials," *Review of Scientific Instruments*, vol. 87, no. 8, p. 085109, Aug. 2016.

[5] J. S. Fada, M. A. Hussain, J. L. Braid, S. Yang, T. J. Peshek, and R. H. French, "Electroluminescent image processing and cell degradation type classification via computer vision and statistical learning methodologies," *44th PVSC*, 2017.

[6] International Electrotechnical Commission, "IEC 61215 Terrestrial photovoltaic (PV) modules - Design qualification and type approval," International Electrotechnical Commission, International Standard, 2016. [Online]. Available: https://webstore.iec.ch/publication/24312

[7] I. Willow Garage, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[8] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[9] SciPy, "SciPy.org SciPy.org," 2014. [Online]. Available: http://www.scipy.org/

[10] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv:1603.04467 [cs]*, Mar. 2016, arXiv: 1603.04467. [Online]. Available: http://arxiv.org/abs/1603.04467

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY: Springer New York, 2009. [Online]. Available: http://link.springer.com/10.1007/978-0-387-84858-7

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3098997.3065386

[13] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, Jul. 1998.

[14] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[15] L. P. Coelho, "Mahotas: Open source software for scriptable computer vision," *Journal of Open Research Software*, vol. 1, no. 1, Jul. 2013. [Online]. Available: http://openresearchsoftware.metajnl.com/articles/10.5334/jors.ac/

[16] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 164–176, Apr. 1980.

[17] H. Abdi and L. J. Williams, "Principal component analysis: Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, Jul. 2010. [Online]. Available: http://doi.wiley.com/10.1002/wics.101

[18] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967. [Online]. Available: http://link.springer.com/10.1007/BF02289588

[19] D. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," p. 24.