Automatic Crack Segmentation and Feature Extraction in Electroluminescence Images of Solar Modules

Xin Chen *, Todd Karin *, Cara Libby †,
Michael Deceglie ‡, Peter Hacke ‡, Timothy J Silverman ‡, Anubhav Jain *

*Lawrence Berkeley National Laboratory, Berkeley, CA, U.S.A
†Electric Power Research Institute, Palo Alto, CA, U.S.A
‡National Renewable Energy Laboratory, Golden, CO, U.S.A

***Abstract*—The effect of cracks in solar cells on the long-term degradation of photovoltaic (PV) modules remains to be determined. To investigate this effect in future studies, it is necessary to quantitatively describe the crack features (*e.g.* length) and correlate them with module power loss. Electroluminescence (EL) imaging is a common technique for identifying cracks. However, it is currently challenging and time-consuming to identify cracks in a large number of EL images and quantify complex crack features by human inspection. This article introduces a fast semantic segmentation method (∼0.18 s/cell) to automatically segment cracks from EL images and algorithms to extract crack features. We fine-tuned a UNet neural network model using pre-trained VGG16 as the encoder and obtained an average F1 score of 0.875 and an intersection over union (IoU) score of 0.782 on the testing set. With cracks and busbars segmented, we developed algorithms for extracting crack features, including the crack-isolated area, the brightness inside the isolated area, and the crack length. We also developed an automatic preprocessing tool for cropping individual cell images from EL images of PV modules (∼0.72 s/module). Our codes are published as open-source software, and our annotated data set composed of various types of cells is published as a benchmark for crack segmentation in EL images.**

***Index Terms*—PV module, crack feature extraction, semantic segmentation, deep learning**

I. Introduction

Global photovoltaic (PV) installation, with crystalline silicon as the most commonly used material for solar cells, has experienced a rapid increase to meet the high demand for clean energy. Solar modules installed in utility-scale PV systems are expected to have a long operational lifetime to be competitive with conventional energy sources [1]. However, solar modules in the field are subject to multiple environmental stresses like temperature variation, mechanical loading, *etc.* [2] These environmental stresses lead to various degradation modes [2], [3] in solar modules and limit their durability. Cell fracture is one of the dominant degradation modes in the early stage of PV systems [4] and may cause a delayed worsening of annual degradation rate [3]. Brittle silicon cells in glass/backsheet modules will fracture under tensile stress caused by front loading (*e.g.*, snow) in the field. For cracked modules, oscillatory field wind causes a long-term cyclic loading of solar modules. The existence of cracks may cause temporary or permanent disconnected areas in a cell, but some research shows that cracked modules do not have severe power loss [5]. Also, the cyclic loading in cracked modules may cause a "recovery" behavior in crack-isolated areas [6], [7]. Currently, the effect of cell cracks on module electric power and the evolution of cracks in long-term cyclic loading is under active investigation [2], [8], [9]. Previous work [10], [11] reports that the influence of cracks may depend on various features, including the number, position of cracks, and electrical resistance introduced by cracks. It is essential to not only detect cracked cells but also quantitatively describe crack features which can be used for future research on crack effects.

Micro-cracks are almost invisible to bare human eyes. Electroluminescence (EL) imaging is a fast, non-destructive method for detecting cracks in solar modules [12], [13], but manual inspection and extraction of complex crack features from a large number of EL images is not practical. As computer vision technology becomes increasingly fast and accurate, such tasks can be shifted to computers. Previous research [14], [15] has reported various computer vision methods to segment cracks from solar cell images. However, those methods may be susceptible to variations in types of

The project was primarily funded and intellectually led as part of the Durable Modules Consortium (DuraMAT), an Energy Materials Network Consortium funded under Agreement 32509 by the U.S. Department of Energy (DOE), Office of Energy Efficiency & Renewable Energy, Solar Energy Technologies Office (EERE, SETO). Lawrence Berkeley National Laboratory is funded by the DOE under award DE-AC02-05CH11231.a

The authors declare no conflicts of interest. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. government. Instruments and materials are identified in this paper to describe the experiments. In no case does such identification imply recommendation or endorsement by LBL. The U.S. government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. government purposes.

solar modules and may not perform well on small cracks from polycrystalline silicon cells because crystallographic defects appear similar to crack lines [16]. Recently, deep learning models have been widely used on computer vision tasks, *e.g.*, VGG [17], ResNet [18], UNet [19], DeepLab [20], *etc.* The application of deep learning models to EL image analysis covers cell classification [21]–[24], defect object detection [25]–[28] and semantic segmentation of defects [29]–[31].

For cell classification work, solar cell images are classified into different categories based on defect types. Deitsch *et al.* [21] trained a convolutional neural network (CNN) model with an accuracy of 88.42% to classify cells into "functional" or "defective". The data set, consisting of 2,624 cell images, was published as a benchmark [32], [33]. However, the models are limited to binary classification without determining the specific defect categories. Karimi *et al.* [23] trained three models, including random forest (RF) [34], SVM, and CNN to classify cells into "good", "cracked" and "corroded". The CNN model outperformed other models with a 99.71% accuracy on their own testing set.

For object detection tasks, the type and position of defective cells are predicted. Zhao *et al.* [26] developed an object detection model to identify 19 types of defects in half-cell solar modules, achieving 70.2% mean Average Precision (mAP). The model can localize target defects and draw a bounding box around them. Meng *et al.* [27] developed a YOLO-PV model based on YOLO model [35] to identify defects in half-cell PV modules. The authors' model achieved an average precision of 94.55%, and the inference speed exceeded 35 fps. In one of our previous works [28], we also designed a pipeline with a YOLO model to automatically inspect millions of solar cells in the field and reported the effects of bush fire damage on the solar farm.

Both the cell classification and object detection methods mentioned above can automatically identify cracked cells. However, they cannot provide more detailed information on crack features (*e.g.*, size, shape, *etc.*), which are essential in correlating cracks with PV performance. Therefore, semantic segmentation is needed to identify defects at the pixel level and provide annotation masks for feature extraction.

Balzategui *et al.* [36] utilized a sliding window method to obtain a segmented defective region (micro-crack region) in the cell image. The authors trained the model on 542 cell images and achieved a recall of 92% and a precision of 85% at the image level. Using the same data set, Balzategui *et al.* [37] trained a UNet model to segment defects. This end-to-end method only requires the input of annotated images. The authors compared the performance of a UNet model without modification, a UNet model with some hidden layers dropped, and a pre-trained UNet model. The UNet model without modification achieved a 72.2% true positive rate and 92.8 % positive predicted value. However, such high scores were only calculated on severe cracks (over 800 pixels). Also, additional metrics like F1 score and intersection over union (IoU) are necessary to evaluate semantic segmentation models.

Mayr *et al.* [29] employed a modified ResNet50 as the backbone to drive segmentation from activation maps and L_p normalization to aggregate the activation maps. The model obtained an F1 score of 0.83 to segment cracks from mono- and polycrystalline solar cells.

Rahman *et al.* [38] incorporated multi-attention networks into the UNet model (MAU-net) to extract effective multi-scale features for defects inspection. This modified model was trained on 620 polycrystalline cell images and obtained an IoU of 0.699 and an F1 score of 0.799 to segment cracks and finger interruption. However, this model was only trained on polycrystalline cells and might lack generalization on other types of solar cells.

Tian *et al.* [39] designed a pipeline to select defective cells with VGG16 and then segment the defects with the UNet++ model. However, the segmentation targets in their work were not clearly distinguished. Pratt *et al.* [40] used the UNet model with VGG16 as the encoder to perform an end-to-end segmentation of cracks. They also correlated the model's output (*e.g.* predicted masks) with EL images taken during accelerated aging tests. Their model has a similar architecture to the model introduced in this paper. However, as we will show, our work also introduces algorithms that can post-process such data to extract crack features based on the model's output.

Fioresi *et al.* [31] trained a Deeplabv3 model with a ResNet50 backbone on 17,064 EL images, including 256 simulated images of PV cells. The model achieved an IoU score of 57.3% to localize non-defective areas, cracks, contact defects, corrosion, and interconnect defects. The authors published their data set, but at the time of this writing, the link provided in the paper leads to an empty data repository.

Based on the reviewed publications related to defect segmentation of solar cells, we could see that the UNet model and its variants were commonly used because the "skip-connection" architecture in the UNet model might help with the "gradient vanishing" problem and improve the model performance in semantic segmentation [41]. However, those publications focus on segmenting defects without providing further algorithms for post-processing model's outputs. Some previous research without using deep learning models reported algorithms for crack feature extraction. Bedrich [42] designed quantitative descriptors of crack length and crack orientation. Whitaker *et al.* [43] extracted the length and convex envelope area of cracks and correlated the crack features with the current-voltage (IV) parameters of mini-modules. However, as mentioned above, non-deep learning methods may not perform well on polycrystalline silicon cells, and more crack features like crack-isolated areas can be extracted.

Most of the work mentioned above presented promising potential in the defect segmentation of solar cells. However, further exploration is still needed in several areas: (i) most work only focuses on segmenting objects (*e.g.* crack) from EL images. Feature extraction based on the segmented defects is needed; (ii) most data sets are not published, so the comparisons of various models do not use the same testing set. A benchmark is needed for related research in the future;

(iii) most training sets contain only one or two specific types of solar cells and lack diversity. More comprehensive benchmarks are still needed.

This paper demonstrates a variant of the UNet model using pre-trained VGG16 as the encoder and algorithms designed to extract crack features from model-predicted masks. We first introduced our automatic cell cropping tool, which cropped out 29,664 individual cell images from module images. The average processing rate is ∼0.72 s/module. The cropped cell images were used to train our model to segment objects, including busbars, crack lines, cross cracks, and dark areas (*i.e.*, degradation area) separately. We reported the performance of our model on the testing set with an average F1 score of 0.875 and an IoU score of 0.782. The average segmentation rate is ∼0.18 s/cell. With the segmented cracks and busbars from the model, we designed algorithms for predicting the worst-case inactive area on solar cells caused by cracks. We also introduce our algorithms for extracting crack length and brightness in the crack-isolated area indicating the disconnection status. The model and algorithms we designed have been utilized to quantitatively correlate crack features with module power loss in accelerated testing and field data sets [9], and we expect they may serve as a basis for future analysis in the field. The tools and models presented in this paper are available as part of the “pv-vision” open-source software [44]. Our cell images and annotations are also published as a benchmark [45] to compare segmentation models. We also uploaded our module images.

This paper primarily focuses on the segmentation of cracks and the extraction of crack features. The main contributions of this work are:

(1) We demonstrated a preprocessing tool to crop out cell images from module images that is robust to complex backgrounds (*e.g.* modules mounted on racks).

(2) We published our images and annotations as a benchmark to compare segmentation models for analyzing EL images in future research. The data set contains various types of solar modules (mono and polycrystalline Si, different busbar configurations, and different cell number configurations).

(3) We developed a semantic segmentation model to separately segment busbars, cracks, *etc*. The model’s performance is competitive with previous work, although quantitative comparisons are difficult due to the lack of standardized benchmarks and a lack of available models with published hyperparameter information.

(4) We designed algorithms to extract crack features including the crack-isolated area, brightness in the crack-isolated area and crack length based on object masks predicted by our model. Those features can be used to describe cracks quantitatively and further correlate cracks with the degradation of PV modules.

(5) We published our model and algorithms as open-source software to analyze EL images of solar modules. The tools have been utilized to analyze crack growth in accelerated testing [9].

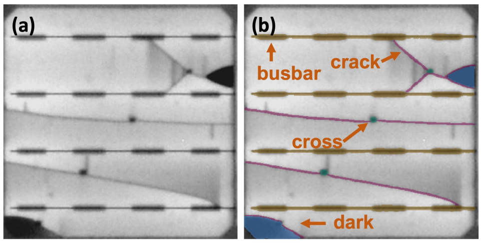

Fig. 1. Example of annotation targets. (a) An original image of mono-Si cells. (b) The annotated image. The four classes are annotated in different colors (purple: “crack”; cyan: “cross”; brown: “busbar”; blue: “dark”).

The remainder of this paper is organized as follows: Section II presents the description of our data set. Section III provides details of the image preprocessing tool, model training and feature extraction algorithms. Section IV shows the performance of our model and tools. The conclusions and outlook are reported in Section V.

II. Data set

Our data set contains 1,837 individual cell images for training and evaluation. The individual cell image is shown in Figure 1(a). To prepare the data set, we cropped out 445 EL images of solar modules into 29,664 cell images and randomly sampled 1,837 of them. The cropping method is demonstrated in Section III. Those module images were collected from various data providers, which increases the diversity of data. Among the 1,837 cell images, 705 of them are monocrystalline silicon (mono-Si) solar cells and the remaining 1,132 are polycrystalline silicon (poly-Si) cells; 54 of them have 2 busbars, 441 have 3 busbars, and the remaining 1342 have 4 busbars; 1,078 of them were cropped from modules of 12 × 6 cells, with the remaining 759 from 10 × 6 modules.

From the 1,837 cell images, we randomly sampled 1,272 images for the training set, 206 for the validation set, and 359 for the testing set. The ratio of the train-val-test set is 7:1:2. The objects that we want to segment from cell images are “busbar” lines, “crack” lines, “cross” dots, and “dark” areas, as shown in Figure 1. “Cross” dot is the dark dot of a cross crack that may be caused by point loads on cells. “Dark” area refers to the degradation areas on cells that appear darker due to lack of electroluminescence. Although our deep learning model is trained to predict “dark” area, we designed an alternative post-processing method to extract the brightness of the crack-isolated area to better process the partially disconnected area. This post-processing algorithm is introduced in Section III. The four objects were annotated on pixel scale, as shown in Figure 1, and the distribution of each category in the data set is shown in Table I. The “Total images” column indicates the number of cell images in each set. The following four columns (“Crack” to “Dark”) show the number of images that contain the specific object to segment. Annotations were performed on Supervisely [46], an online annotation software, by a group of 8 experienced annotators. Two annotators verified all the annotation results to ensure the quality of annotations.

TABLE I
DISTRIBUTION OF CELL IMAGES BY VARIOUS CATEGORIES IN DATA SETS.

Data set	Total images	Crack	Busbar	Cross	Dark
Train	1272	493	1272	323	270
Train oversample	1765	986	1765	598	538
Val	206	79	206	57	40
Test	359	120	359	97	68
Test crack	322	290	322	225	171

The training set is imbalanced, where the "crack" class accounts for 38.8% of the training set. Therefore, we oversampled the images with the "crack" class so that the "crack" class accounts for 55.9%, which is comparable to uncracked cells. The performance was compared between the model trained on the original training set and the oversampled data set. Only 133 (37%) images in the testing set have "crack" or "cross" crack objects. To further evaluate the performance of our model on cracks, we built another testing set named "test crack" with 322 images that only have "crack" lines or "cross" cracks. Those images are sampled from the 29,664 cell images and do not overlap with the training and validation sets.

III. METHODS

A. Automatic cropping of cells from module images

An example of the module image to be processed is shown in Figure 2. The lens distortion of those images has been corrected by data providers prior to being processed by us. The inner edges of each solar cell in the solar module are automatically detected to split a module image into individual cell images. Since images from different data providers have various sizes (*e.g.* 4096×2500 pixels, 4240×2832 pixels, *etc.*), we first transformed the original module image into a uniform size of 4000×2500 pixels. We then binarized the image using the adaptive thresholding method in OpenCV [47] to retain edges, as shown in Figure 2(b). After that, we split the binary image into 100 horizontal strips, each with a size of 4000×25 pixels, as schematically illustrated in Figure 2(c). For each strip, we summed up the grayscale values in the vertical direction to obtain a cell edge distribution as shown in Figure 2(d). The grayscale intensity was normalized and ranged from 0 (approaching black) to 1 (approaching white). The edges of solar cells are white in Figure 2(c) and appear as peaks in Figure 2(d). Afterward, we determined the positions of cell edges in each strip by detecting the peaks using "signal.find_peaks" tool from Scipy [48] and fit lines to these peak positions that represent each edge. We performed an analogous procedure to fit edges in vertical strips. The detected edges are demonstrated in Figure 2(e). Figure 2(f) is another example of detected inner edges in a poly-Si solar module. This result demonstrates that the cropping algorithm remains robust when racks are in the background. With inner edges detected, we crop out each cell by using intersections of edges as corners of cells. The size is selected to maintain the cell's original size in module images, which in our case is 400×400 pixels. The cropped cells are shown in Figure 2(g).

The script was executed with Python3 [49] on a MacBook Pro (2.8 GHz Quad-Core Intel Core i7 processor, 16 GB 2133 MHz LPDDR3 memory).

B. Model training and evaluation

The training data was augmented by horizontal flip, vertical flip, and random crop. Before performing the random crop, the images were duplicated three times. Then the images were cropped at random to 70% - 90% of width and height. The augmented training set contains 11,448 images, and the augmented, oversampled training set contains 15,885 images.

We fine-tuned a UNet model with a pretrained VGG16 model [17] as the encoder (pretrained on ImageNet [50]). The model was deployed and trained with Supervisely for annotation and training convenience. The architecture of our model is shown in Figure 3. The encoder performs convolution and max-pooling operations to transform the input images into deep feature maps. The decoder performs upsampling on the feature map and outputs the probability map with five layers, corresponding to the five objects (*i.e.*, "background", "crack", "busbar", "cross", "dark") in our task. The probability map is used to compute loss during the training process or make predictions during the inference process. We used negative log likelihood loss (NLLLoss) [51] as the loss function. The size of the input layer is 256×256 pixels, which is the default size of the pretrained model. A larger size is possible, but it exceeds the limitation of our GPU device. The learning rate was set to 0.001, and the training batch size was 12. The model was trained for 10 epochs, and the epoch with the lowest loss on the validation set yielded the final model. We used precision, recall, F1 score, and IoU to report the performance of our model. Each score is described as:

$$\text{Precision} = \frac{\sum_I \text{TP}}{\sum_I \text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\sum_I \text{TP}}{\sum_I \text{TP} + \text{FN}} \tag{2}$$

$$\text{F}_1 = \frac{\sum_I 2\text{TP}}{\sum_I 2\text{TP} + \text{FN} + \text{FP}} \tag{3}$$

$$\text{IoU} = \frac{\sum_I \text{TP}}{\sum_I \text{TP} + \text{FN} + \text{FP}} \tag{4}$$

where TP represents true-positives (model correctly identifies a pixel as an existing object), FP represents false-positives (model detects a pixel as a target object but is wrong). FN represents false negatives (the model fails to detect a pixel labeled as a target object). The subscript I represents the sum over all images in the data set. Our model was trained on 4 NVIDIA Tesla K80 GPUs (48 GB GDDR5 memory) and tested on 1 NVIDIA Tesla K80 GPU (12 GB GDDR5 memory).

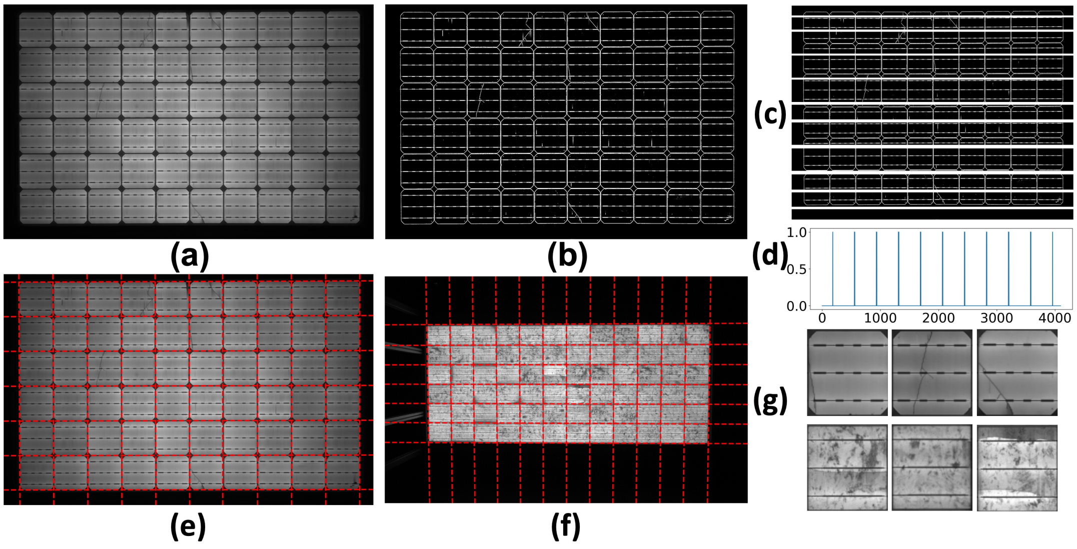

Fig. 2. Process of automatic cell cropping. (a) Original EL image of a mono-Si solar module. (b) Binarized image with edges retained. (c) A schematic illustration of horizontal image strips. The actual strips are much thinner than the schematic image here. (d) Detected peaks on a horizontal strip. The x-axis is the width of a horizontal strip. The y-axis is the normalized grayscale value. Each peak corresponds to a vertical edge of the solar module. (e) Detected edges of individual solar cells. (f) Another example of detected inner edges on a poly-Si solar module. (g) Examples of cropped individual solar cells.

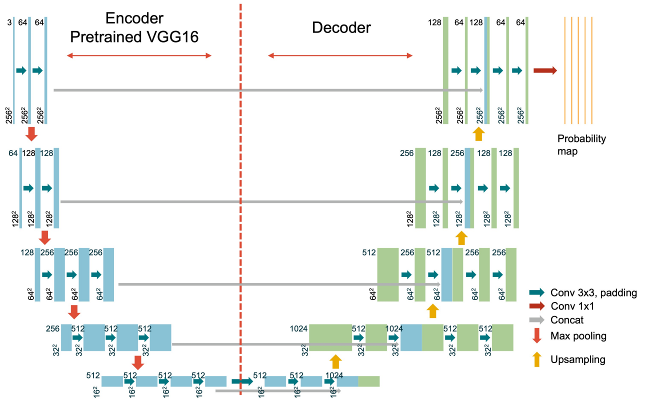

Fig. 3. Architecture of the UNet model. The encoder is a VGG16 model pretrained on ImageNet. The decoder outputs a probability map with five layers, corresponding to the five objects (*i.e.*, “background”, “crack”, “busbar”, “cross”, “dark”) in our task.

C. Crack feature extraction

1) Crack-isolated area prediction: The existence of cracks in solar cells may cause isolated (electrically disconnected or degraded) areas that impede the flow of charge carriers. In the worst case, all charge carriers generated in isolated areas are blocked from busbars, and those areas subsequently become inactive parts of solar cells, making no contributions to output power. Figure 4(a) schematically shows the example of isolated areas within a solar cell. In this figure, the blue area indicates the worst-case inactive area because cracks prevent charge carriers generated in this area from flowing to busbars. The green arrows represent possible paths connected to busbars. Herein, we assume that charge carriers can only reach busbars through fingers that are perpendicular to busbars. If cracks block all paths to potential busbars, isolation can occur. However, not all cracks lead to isolation. In the second row from bottom of the schematic solar cell, although a diagonally oriented crack exists, it does not result in isolation because carriers can still reach a busbar by traveling perpendicularly along fingers. The schematic diagram shows that charge carriers generated above or below this crack can still flow to the nearest busbars and contribute to the output current.

Based on cracks and busbars detected by the UNet model, we developed an algorithm to predict the worst-case isolation areas, as illustrated in Figure 4(b)-(f). We first skeletonized the annotation masks (Figure 4(c)) of cracks and busbars using scikit-image [52], as shown in Figure 4(d). Figure 4(d) is essentially a 400×400 matrix, with grayscale values of background as 0 and crack and busbar annotations other than 0. We used different grayscale values to distinguish crack and busbar. In each column of this matrix, our algorithm assigned the same grayscale value of busbar to background pixels vertically connected to busbars (vertically growing the busbars until they reached a crack). After iterating over all the columns, only the isolated areas retained a grayscale value of 0. Figure 4(e) shows the result of the isolated area. Note that we inverted the grayscale value for better visualization so that the isolated area appears white. In summary, areas that cannot be reached perpendicularly from busbars are considered as isolated by cracks. The equation of calculating the proportion of isolated area in the binary image Figure 4(e) is:

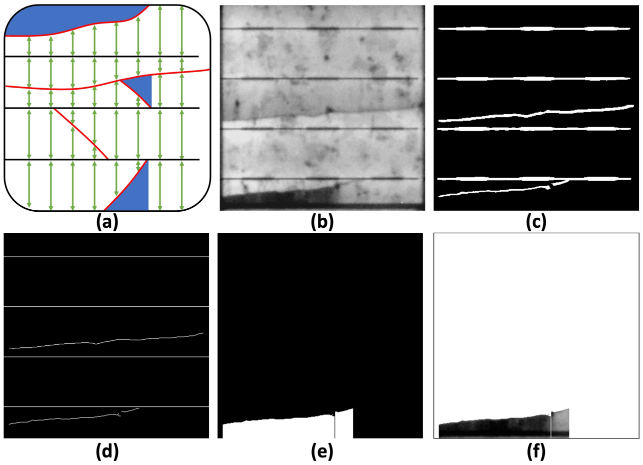

Fig. 4. (a) Schematic diagram to explain inactive area prediction algorithm. Red lines represent cracks. Green arrows represent the flow path of charge carriers. The blue part indicates the isolation area. (b) The original image of a poly-Si cell. (c) Annotation masks of cracks and busbars. The masks are combined in one image for visualization. The UNet model can output masks of each object separately. (d) Skeletonized masks of cracks and busbars combined in one image. (e) Crack-isolated area using our algorithm. (f) Original image in the crack-isolated area. For clarification, (a) is not the schematic diagram of (b).

$$p(\%) = \frac{\text{isolated area}}{\text{cell area}} \times 100\% = \frac{\sum_{i \in \text{cell}} G_i^{binary}}{\text{cell area}} \times 100\% \quad (5)$$

where G_i^{binary} is the gray-scale value of pixel i that belongs to the binary image Figure 4(f). "Cell area" in Figure 4(f) is 400×400 pixels. This is a simplified computation because, in the actual situation, the fingers around cracks between busbars can increase series resistance and influence the electric power of solar modules.

2) Brightness of crack isolated area: Cracks can cause a change of break resistance in solar cells and lead to power loss [10]. The crack resistance can also cause a decrease in EL intensity in crack-isolated areas. Therefore, the brightness of the crack-isolated regions in EL images correlates with power loss, although a well-established relation is still under investigation. Here we extracted the average gray-scale value of the isolated area based on our crack-isolated area calculation algorithm. The equation is:

$$\bar{\Phi}_{isolated} = \frac{\sum_{i \in \text{isolated area}} G_i^{cell}}{\text{isolated area}} \quad (6)$$

where G_i^{cell} is the gray-scale value of pixel i that belongs to the isolated area in the original image, as shown in Figure 4(f). This method acts as an alternative method to quantify the severity of the "dark" area in EL images, especially when the area is partially disconnected.

3) Crack length: The skeletonized crack mask is a binary (0-1) image with the gray-scale value of cracks of 1 and the background of 0, as shown in Figure 4(d). Since the width of the skeletonized crack is 1, the overall crack length per cell is equal to the number of pixels of the skeletonized crack, which is equal to the sum of pixel values in the binary figure. A similar method was also used in previous research [43]. The equation can be described as:

$$\text{crack length} = \sum_{i \in \text{cell}} G_i^{binary} \quad (7)$$

where G_i^{binary} is the gray-scale value of pixel i that belongs to the binary image of skeletonized crack masks.

IV. Results and discussion

A. Cell cropping tool

Figure 2(g) illustrates the results of cell cropping. We first applied the cropping tool to 576 images of solar modules that were randomly sampled, and 445 of them were successfully cropped into 29,664 cell images. Figure 5 demonstrates some poorly processed modules. Among failed images, $\sim$50% can be considered as poorly shot based on manual inspection. The solar modules in those images are truncated, severely tilted or poorly exposed. Figure 5(a) is a truncated module that cannot be used for analysis. The module in Figure 5(b) is severely tilted. We estimate that only 12 out of 576 ($\sim$2%) images are severely tilted so the perspective transform of severely tilted modules was not included in the beginning stage of research. However, related perspective transform algorithms [28] have been added into the latest version of "pv-vision". In Figure 5(c), although the photo is clear, the edges and corners of the module are dark and blend into the background. The algorithm failed to determine the boundary between the module and the background. Figure 5(d) was not transformed, possibly because the contrast between bright cells and dark inner edges is low, so the algorithm failed to detect the inner edges. Another possible reason is that its size (3550×1810 pixels) is smaller than others. After transformation, some information may be lost. We consider Figure 5(a) a data acquisition error, while errors in Figure 5(b)(c)(d) can be fixed in the updated version of the open-source tool.

B. Model performance

We trained separate CNN models on the original training set and oversampled training set. The performance of each model on the validation set is illustrated in Figure 6. Recall is the fraction of actual pixels of objects that were correctly labeled by the model, whereas precision is the fraction of predicted pixels of objects that are identified correctly. F1 score and IoU scores balance both precision and recall. IoU score penalizes bad segmentation (FP and FN) more than F1 score. In Figure 6, both macro F1 and IoU scores show that the oversampling method can improve the model's performance. Therefore, the model we finally selected is the model trained on the oversampled training set. Its average F1 score is 0.882 and average IoU score is 0.795, as listed in Table II.

We also tested the performance of our model on the two testing sets (*i.e.*, original testing set and testing set with cracked cells only) to investigate the generalization of the

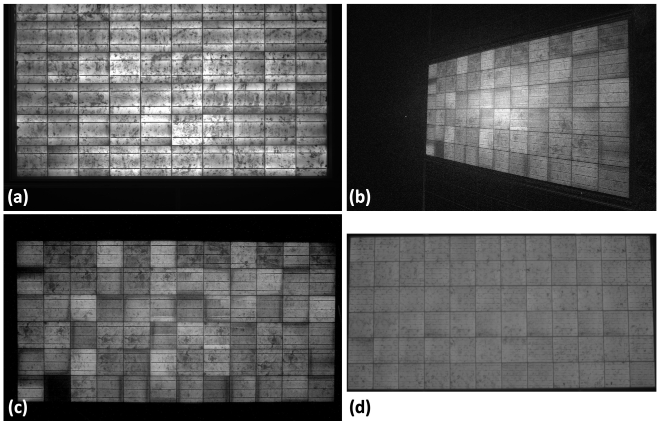

Fig. 5. Examples of challenges in processing module images. (a) is a truncated module. (b) is a severely tilted image. (c) is a module with dark edges and corners. (d) was not cropped, probably due to low image contrast or smaller image size.

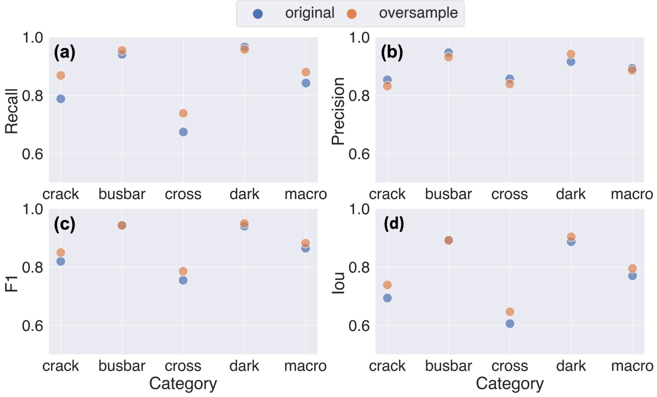

Fig. 6. Performance of models on the validation set. The scores (a) recall, (b) precision, (c) F1, (d) IoU are computed based on each category. The “macro” represents the average score of the four target categories. The two models were trained on the original and oversampled training sets respectively. Note that some points overlap with each other.

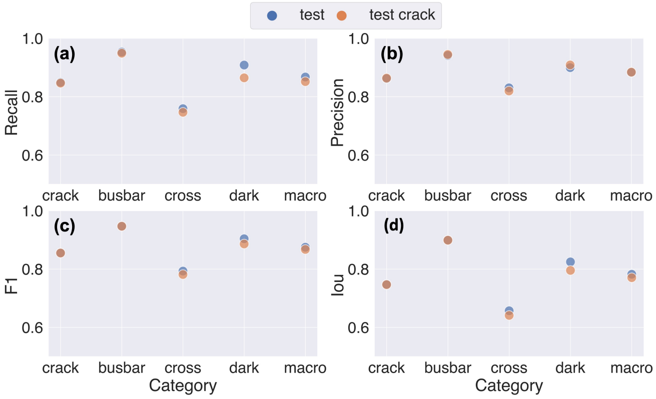

Fig. 7. Performance of the model on the testing set. This model was trained on the oversampled training set. “Test” means the original testing set with the same distribution of categories as the population. “Test crack” means the testing set that merely contains cracked cells. Note that some points overlap with each other.

TABLE II
MACRO AVERAGE METRICS OF THE OPTIMAL MODEL ON THE VALIDATION SET AND TWO TESTING SETS.

Data set	Avg Precision	Avg Recall	Avg F1	Avg IoU
Val	0.886	0.879	0.882	0.795
Test	0.883	0.867	0.875	0.782
Test crack	0.884	0.852	0.867	0.770

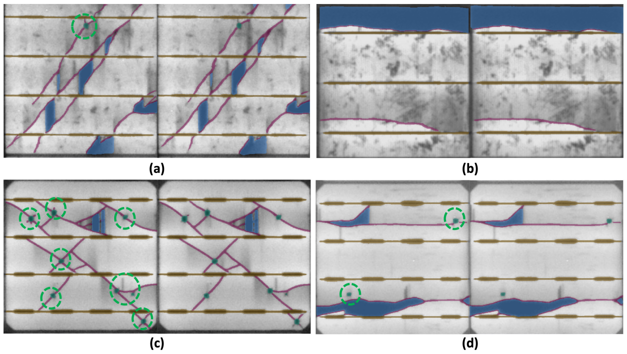

Fig. 8. Examples of segmented objects from UNet model. For each subfigure, the left is the ground truth, and the right is the prediction. (a)(b) are poly-Si cells, and (c)(d) are mono-Si cells. The four classes are predicted in different colors (purple: “crack”; cyan: “cross”; brown: “busbar”; blue: “dark”). The “cross” type defect is circled for better visualization.

model. The performance is shown in Table II and Figure 7. Table II illustrates that our model retains good performance when generalizing to new data, *i.e.*, the F1 (0.875) and IoU (0.782) scores on the original testing set are not significantly lowered compared to scores on the validation set. Also, Figure 7 shows that when dealing with cracked cells only, our model still obtained similar F1 and IoU scores to the original testing set.

Figure 8 illustrates some example predictions from the final model on the testing set. Figure 8(a)(b) are poly-Si cells and (c)(d) are mono-Si cells. The left image of each sub-figure shows the manually annotated objects and the right image shows the prediction from the UNet model. It can be seen that the predicted annotations show high consistency with the manual ones. Figure 8(b) illustrates that the model can still identify cracks correctly when there are crack-like textures on poly-Si cells. Also, the model can precisely detect cracks of different orientations, especially in Figure 8(c) where cracks have horizontal, obliquely upward and obliquely downward orientations. Figure 8(d) shows that the model can detect the “Y” shaped cracks.

Some shortcomings of the model are shown in Figure 9. The model does not identify small cracks around the “cross” objects very accurately, as shown in Figure 9(a). The model may also miss some cracks for severely shattered cells, like Figure 9(b). Figure 9(c) shows that the “dark” area is not well identified, although cracks are identified correctly. The main reason is that it is hard to define precisely whether areas with medium brightness should be defined as “dark” areas. For example, the crack-enveloped area in Figure 9(c) is

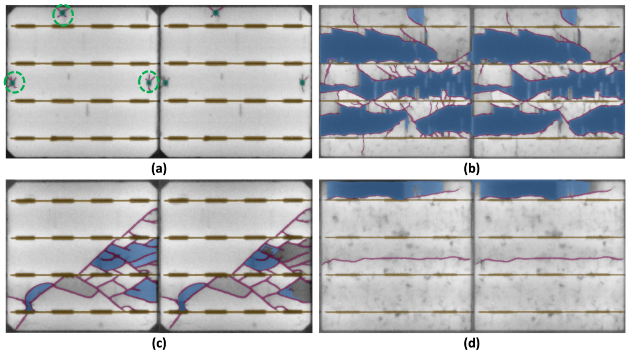

Fig. 9. Examples of shortcomings of the model. For each subfigure, the left is the ground truth, and the right is the prediction. The four classes are predicted in different colors (purple: "crack"; cyan: "cross"; brown: "busbar"; blue: "dark"). The "cross" type defect is circled for better visualization.

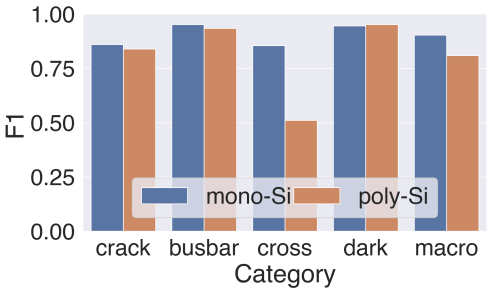

Fig. 10. Performance of the model on mono-Si and poly-Si cells. The F1 score was computed based on the validation set.

partially disconnected and not totally dark, so it is ambiguous even for annotators to determine the "dark" area. Another example is shown in Figure 9(d), where the model counts some dark margins as "dark" areas which are inconsistent with annotation. Also, the brightness of the disconnected area is influenced by the injected current. IEC TS 60904-13 [53] shows that a partially disconnected area caused by mode B cracks exhibits high contrast with I_{sc} applied but lower contrast with $10\% I_{sc}$ applied. Therefore, the lack of a rigorous definition in such analyses of what constitutes a disconnected area led to inconsistent manual annotations and further justifies the use of the feature extraction of isolated area brightness as an alternate method. Another shortcoming we noticed is that the model may perform worse on poly-Si cells than mono-Si cells, especially for "cross" object as shown in Figure 10. The possible reason is that poly-Si cells show more complex surface textures (e.g., dislocation clusters). This could be improved by using more poly-Si cells during the training process or modifying the loss function to add more penalties when the model wrongly processes the poly-Si cells.

C. Crack-isolated area prediction

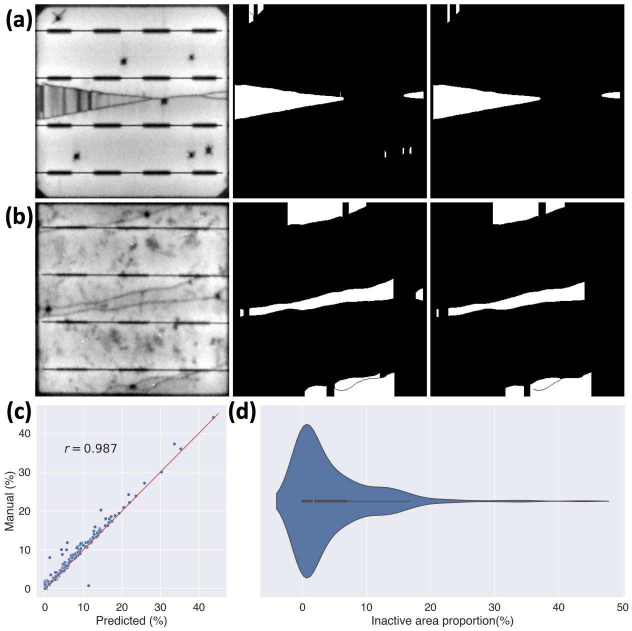

Fig. 11. (a)(b) Results from inactive area prediction tool. Figures from left to right are the original image, the isolated area (white area) generated from manual annotations and the isolated area from UNet-predicted crack annotations. (c) Comparison of isolated area proportion from manual and UNet-predicted crack annotations across all cells in the "test crack" data set. The red line is a reference line where $y = x$. (d) Distribution of predicted inactive area proportion in the "test crack" set.

Both crack length and brightness of isolated area extraction algorithms depend on the process of the crack-isolated area prediction algorithm, so we tested the performance of the isolated area prediction algorithm. An example of the predicted crack-isolated area is shown in Figure 11(a)(b). We applied this algorithm to the "test crack" set where all cells have cracks. By comparing areas generated from true and predicted crack annotations across the entire "test crack" set, the F1 score is 0.905, and the IoU score is 0.827. Figure 11(c) shows a high consistency between the inactive area proportion calculated from manual and UNet-predicted crack annotations, with a Pearson correlation coefficient [54] of 0.987. The predicted inactive area proportion in the testing set has a distribution shown in Figure 11(d) where 75% of cells have inactive areas lower than 6.86%, and the average proportion is 4.54%. 31.4% of those cracked cells have zero inactive areas, which means they have potentially insignificant cracks such as the one in the middle of Figure 4(b). Thus, an important result of this analysis is that, at least in the context of our data set, the presence of cracks does not imply a large power loss area. Further analysis and comparison to measured power data may help to separate more problematic cracks from those that are less likely to induce power loss.

V. Conclusion and Outlook

This paper introduced methods for automatic crack segmentation and crack feature extraction tools to process EL images of solar modules. We trained a UNet model to perform semantic segmentation of solar cells to segment "crack" lines, "cross" cracks, "busbars", and "dark" areas. The model exhibits good performance, with a macro-average F1 score of 0.875 and an IoU score of 0.782 on the testing set. We further tested the model's performance specifically on cracked cells and obtained an F1 score of 0.867 and an IoU score of 0.770.

We developed algorithms to extract crack features based on the annotation masks predicted from our model, including crack-isolated area, brightness of the isolated area and crack length. We evaluated our algorithm for predicting the worst-case crack-isolated area on the testing set and obtained an F1 score of 0.905 and an IoU score of 0.827. With this algorithm, we analyzed the inactive area proportion of the "test crack" data set and found that 31.4% of cells have cracks with no isolated areas, suggesting that some cracks may not result in power loss, at least initially.

The open-source tools and model weights are published on Github [44]. Our data is published on the DuraMat Data Hub [45]. Future work could further refine the model by training with more EL images with other types of cells (*e.g.*, half-cut cells) by transfer learning and develop circuit models translating the worst-case isolated area into a maximum power loss value. Also, additional EL images with $10\%I_{sc}$ applied might be needed for better crack identification since previous research [10] shows that some cracks can be hidden in images with I_{sc} applied but appear more obvious in $10\%I_{sc}$ images. With the crack segmentation model and crack feature extraction algorithms, we can correlate those features with current-voltage (IV) parameters to quantify how cracks influence the degradation of PV modules. One could further apply the tools to time-series images to investigate how crack features evolve over time, *e.g.*, how cracks may progressively grow larger and ultimately lead to the formation of inactive areas. We also plan to apply the tools to large-scale field data to quantify the possible degradation of PV modules caused by cracks.

Acknowledgment

The authors would like to thank Andrew Gabor for sharing part of the EL images in our data set. We also would like to thank Matthew Jones, Rajalakshmi Sundaramoorthy, Will Hobbs, Xan McPherson for their efforts in the manual annotation.

References

[1] K. Ardani, P. Denholm, T. Mai, R. Margolis, E. O'Shaughnessy, T. Silverman, and J. Zuboy, "Solar futures study," U.S. Department of Energy, Tech. Rep., 2021.

[2] M. Aghaei, A. Fairbrother, A. Gok, S. Ahmad, S. Kazim, K. Lobato, G. Oreski, A. Reinders, J. Schmitz, M. Theelen *et al.*, "Review of degradation and failure phenomena in photovoltaic modules," *Renewable and Sustainable Energy Reviews*, vol. 159, p. 112160, 2022.

[3] D. C. Jordan, T. J. Silverman, B. Sekulic, and S. R. Kurtz, "Pv degradation curves: non-linearities and failure modes," *Progress in Photovoltaics: Research and Applications*, vol. 25, no. 7, pp. 583–591, 2017.

[4] M. Köntges, G. Oreski, U. Jahn, M. Herz, P. Hacke, K.-A. Weiss, G. Razongles, M. Paggi, D. Parlevliet, T. Tanahashi *et al.*, "Assessment of photovoltaic module failures in the field," International Energy Agency: IEA, Tech. Rep., 2017.

[5] M. Dhimish and Y. Hu, "Rapid testing on the effect of cracks on solar cells output power performance and thermal operation," *Scientific Reports*, vol. 12, no. 1, pp. 1–11, 2022.

[6] M. Paggi, I. Berardone, A. Infuso, and M. Corrado, "Fatigue degradation and electric recovery in silicon solar cells embedded in photovoltaic modules," *scientific reports*, vol. 4, no. 1, pp. 1–7, 2014.

[7] J. Käsewieter, F. Haase, and M. Köntges, "Model of cracked solar cell metallization leading to permanent module power loss," *IEEE Journal of Photovoltaics*, vol. 6, no. 1, pp. 28–33, 2015.

[8] T. J. Silverman, M. G. Deceglie, M. Owen-Bellini, W. B. Hobbs, and C. Libby, "Cracked solar cell performance depends on module temperature," in *2021 IEEE 48th Photovoltaic Specialists Conference (PVSC)*. IEEE, 2021, pp. 1691–1692.

[9] C. Libby, B. Paudyal, X. Chen, W. B. Hobbs, D. Fregosi, and A. Jain, "Analysis of pv module power loss and cell crack effects due to accelerated aging tests and field exposure," *IEEE Journal of Photovoltaics*, pp. 1–9, 2022.

[10] M. Köntges, I. Kunze, S. Kajari-Schröder, X. Breitenmoser, and B. Bjørneklett, "The risk of power loss in crystalline silicon based photovoltaic modules due to micro-cracks," *Solar Energy Materials and Solar Cells*, vol. 95, no. 4, pp. 1131–1137, 2011.

[11] A. Morlier, F. Haase, and M. Köntges, "Impact of cracks in multicrystalline silicon solar cells on pv module power—a simulation study based on field data," *IEEE Journal of Photovoltaics*, vol. 5, no. 6, pp. 1735–1741, 2015.

[12] M. Abdelhamid, R. Singh, and M. Omar, "Review of microcrack detection techniques for silicon solar cells," *IEEE Journal of Photovoltaics*, vol. 4, no. 1, pp. 514–524, 2013.

[13] U. Jahn, M. Herz, M. Köntges, D. Parlevliet, M. Paggi, I. Tsanakas, J. Stein, K. Berger, S. Ranta, R. French, M. Richter, T. Tanahashi, E. Ndrio *et al.*, "Review on Infrared (IR) and Electroluminescence (EL) Imaging for Photovoltaic Field Applications," International Energy Agency: IEA, Tech. Rep., 2018.

[14] S. A. Anwar and M. Z. Abdullah, "Micro-crack detection of multicrystalline solar cells featuring an improved anisotropic diffusion filter and image segmentation technique," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, pp. 1–17, 2014.

[15] S. Spataru, P. Hacke, and D. Sera, "Automatic detection and evaluation of solar cell micro-cracks in electroluminescence images using matched filters," in *2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC)*. IEEE, 2016, pp. 1602–1607.

[16] M. Köntges, S. Kurtz, C. Packard, U. Jahn, K. A. Berger, K. Kato, T. Friesen, H. Liu, M. Van Iseghem, J. Wohlgemuth *et al.*, "Review of failures of photovoltaic modules," International Energy Agency: IEA, Tech. Rep., 2014.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December. IEEE Computer Society, dec 2016, pp. 770–778. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[20] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[21] S. Deitsch, V. Christlein, S. Berger, C. Buerhop-Lutz, A. Maier, F. Gallwitz, and C. Riess, "Automatic classification of defective photovoltaic module cells in electroluminescence images," *Solar Energy*, vol. 185, pp. 455–468, 2019.

[22] M. W. Akram, G. Li, Y. Jin, X. Chen, C. Zhu, X. Zhao, A. Khaliq, M. Faheem, and A. Ahmad, "Cnn based automatic detection of photovoltaic cell defects in electroluminescence images," *Energy*, vol. 189, p. 116319, 2019.

[23] A. M. Karimi, J. S. Fada, M. A. Hossain, S. Yang, T. J. Peshek, J. L. Braid, and R. H. French, "Automated pipeline for photovoltaic module electroluminescence image processing and degradation feature classification," *IEEE Journal of Photovoltaics*, vol. 9, no. 5, pp. 1324–1335, 2019.

[24] W. Tang, Q. Yang, K. Xiong, and W. Yan, "Deep learning based automatic defect identification of photovoltaic module using electroluminescence images," *Solar Energy*, vol. 201, pp. 453–460, 2020.

[25] X. Zhang, Y. Hao, H. Shangguan, P. Zhang, and A. Wang, “Detection of surface defects on solar cells by fusing multi-channel convolution neural networks,” *Infrared Physics & Technology*, vol. 108, p. 103334, 2020.

[26] Y. Zhao, K. Zhan, Z. Wang, and W. Shen, “Deep learning-based automatic detection of multitype defects in photovoltaic modules and application in real production line,” *Progress in Photovoltaics: Research and Applications*, vol. 29, no. 4, pp. 471–484, 2021.

[27] Z. Meng, S. Xu, L. Wang, Y. Gong, X. Zhang, and Y. Zhao, “Defect object detection algorithm for electroluminescence image defects of photovoltaic modules based on deep learning,” *Energy Science & Engineering*, vol. 10, no. 3, pp. 800–813, 2022.

[28] X. Chen, T. Karin, and A. Jain, “Automated defect identification in electroluminescence images of solar modules,” *Solar Energy*, vol. 242, pp. 20–29, 2022.

[29] M. Mayr, M. Hoffmann, A. Maier, and V. Christlein, “Weakly supervised segmentation of cracks on solar cells using normalized l p norm,” in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1885–1889.

[30] E. Sovetkin, E. J. Achterberg, T. Weber, and B. E. Pieters, “Encoder–decoder semantic segmentation models for electroluminescence images of thin-film photovoltaic modules,” *IEEE Journal of Photovoltaics*, vol. 11, no. 2, pp. 444–452, 2020.

[31] J. Fioresi, D. J. Colvin, R. Frota, R. Gupta, M. Li, H. P. Seigneur, S. Vyas, S. Oliveira, M. Shah, and K. O. Davis, “Automated defect detection and localization in photovoltaic cells using semantic segmentation of electroluminescence images,” *IEEE Journal of Photovoltaics*, vol. 12, no. 1, pp. 53–61, 2021.

[32] C. Buerhop-Lutz, S. Deitsch, A. Maier, F. Gallwitz, S. Berger, B. Doll, J. Hauch, C. Camus, and C. J. Brabec, “A benchmark for visual identification of defective solar cells in electroluminescence imagery,” in *35th European PV Solar Energy Conference and Exhibition*, vol. 12871289, 2018.

[33] S. Deitsch, C. Buerhop-Lutz, E. Sovetkin, A. Steland, A. Maier, F. Gallwitz, and C. Riess, “Segmentation of photovoltaic module cells in uncalibrated electroluminescence images,” *Machine Vision and Applications*, vol. 32, no. 4, pp. 1–23, 2021.

[34] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[36] J. Balzategui, L. Eciolaza, N. Arana-Arexolaleiba, J. Altube, J.-P. Aguerre, I. Legarda-Ereño, and A. Apraiz, “Semi-automatic quality inspection of solar cell based on convolutional neural networks,” in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2019, pp. 529–535.

[37] J. Balzategui, L. Eciolaza, and N. Arana-Arexolaleiba, “Defect detection on polycrystalline solar cells using electroluminescence and fully convolutional neural networks,” in *2020 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2020, pp. 949–953.

[38] M. R. U. Rahman and H. Chen, “Defects inspection in polycrystalline solar cells electroluminescence images using deep learning,” *IEEE Access*, vol. 8, pp. 40 547–40 558, 2020.

[39] S. Tian, W. Li, S. Li, G. Tian, L. Sun, and X. Ning, “Image defect detection and segmentation algorithm of solar cell based on convolutional neural network,” in *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*. IEEE, 2021, pp. 154–157.

[40] L. Pratt, D. Govender, and R. Klein, “Defect detection and quantification in electroluminescence images of solar pv modules using u-net semantic segmentation,” *Renewable Energy*, vol. 178, pp. 1211–1222, 2021.

[41] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, “The importance of skip connections in biomedical image segmentation,” in *Deep learning and data labeling for medical applications*. Springer, 2016, pp. 179–187.

[42] K. G. Bedrich, “Quantitative electroluminescence measurements of pv devices,” Ph.D. dissertation, Loughborough University, 2017.

[43] C. M. Whitaker, B. G. Pierce, A. M. Karimi, R. H. French, and J. L. Braid, “Pv cell cracks and impacts on electrical performance,” in *2020 47th IEEE Photovoltaic Specialists Conference (PVSC)*. IEEE, 2020, pp. 1417–1422.

[44] X. Chen, “pv-vision,” 2022. [Online]. Available: https://github.com/hackingmaterials/pv-vision

[45] X. Chen, T. Karin, C. Libby, M. Deceglie, P. Hacke, and A. Jain. A benchmark for crack segmentation in el images. [Online]. Available: https://datahub.duramat.org/dataset/crack-segmentation

[46] Supervisely. [Online]. Available: https://supervise.ly/

[47] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[48] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020.

[49] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[51] Pytorch nllloss. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.NLLLoss.html

[52] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, “scikit-image: image processing in python,” *PeerJ*, vol. 2, p. e453, 2014.

[53] I. E. Commission *et al.*, “Photovoltaic devices-part 13: Electroluminescence of photovoltaic modules,” International Electrotechnical Commission, Tech. Rep., 2018.

[54] R. Walpole, R. Myers, S. Myers, and K. Ye, *Probability and Statistics for Engineers and Scientists*, 9th ed. Pearson Education, 2017.