Corrigendum

Corrigendum: GAN-Based Augmentation for Improving CNN Performance of Classification of Defective Photovoltaic Module Cells in Electroluminescence Images (IOP Conf. Ser.: Earth Environ. Sci. 354 012106)

Z Luo¹, S Y Cheng¹ and Q Y Zheng¹

¹Institute of Micro/Nano Devices and Solar Cells, College of Physics and Information Engineering, Fuzhou University, Fuzhou 350116, China

Published 29 November 2019

The corresponding author acknowledges not correctly referencing the content taken from https://github.com/zae-bayern/elpv-dataset. The author apologises for the error.
GAN-Based Augmentation for Improving CNN Performance of Classification of Defective Photovoltaic Module Cells in Electroluminescence Images

Z Luo¹, S Y Cheng¹ and Q Y Zheng¹,²
¹Institute of Micro/Nano Devices and Solar Cells, College of Physics and Information Engineering, Fuzhou University, Fuzhou 350116, China
E-mail: zhengqy@vip.sina.com

Abstract. Electroluminescence (EL) imaging is an effective way for the examining of photovoltaic (PV) modules. Compared with manual analysis, using Convolutional Neural Network (CNN) for classification is much more convenient but it requires a certain amount of annotated training samples which cannot be acquired handily. In this paper, we present a method for augmenting the existing dataset of EL images using Generative Adversarial Networks (GANs) and propose a model called AC-PG GAN aiming at this. Three chosen CNN models are used to examine the effectiveness of the proposed GAN model and have achieved an improvement of the classification accuracy with the augmented dataset after some adjustment and the maximum improvement is up to 14%.

1. Introduction
Defects in solar cells are closely related to module efficiency. To guarantee the quality of the PV module, detection of defects is a must. One of the methods of detecting defects is by measuring the current and voltage which called current-voltage measurements, but it may not be able to reveal some types of defects such as small cracks. Compared with current-voltage measurements, Electroluminescence (EL) imaging is a more effective way and has been widely used in the industry. EL images are acquired by imaging the EL emission induced by a current applied PV module using a silicon Charge-coupled Device (CCD). In order to examine defects, image classification algorithms including Support Vector Machine [1] (SVM) and Convolutional Neural Networks (CNN) are employed on the captured EL images to detect defects and classify them if exist. EL images acquisition requires a specific environment like complete darkness, as the consequence of which, limits the number of acquired samples to a great extent. Besides, when employ supervised machine learning algorithms such as CNN, the training samples should be annotated which require expert knowledge and is also time-consuming. Lacking annotated training data will disturb the performance of supervised machine learning algorithms, and maybe even worse, causing overfitting.

Although we can easily access some public EL images datasets nowadays, most of them do not meet the requirements of the supervised machine learning algorithms still due to the limitation of size. Since it is not easy to gain new samples, data augmentation is the most reasonable approach. Traditional data augmentation techniques perform a simple transformation on the original image to obtain a unique view of it thus extending the dataset. Those transformations include rotation, reflection, cropping, Color Jittering, and PCA Jittering. Traditional data augmentation techniques provide an effective way to help prevent supervised machine learning away from overfitting and it has become a
standard procedure before training the algorithm. However, tiny modifications done by some traditional augmentation techniques add little extra information to the training sample, therefore, is not that effective for improving the classification performance. Additionally, some of the augmented samples may appear different from the actual objects in real life, and it actually not does anything helpful when we try to employ the trained algorithms to other datasets.

In this paper, we demonstrate another approach for data augmentation via generating synthetic images by employing generative adversarial networks (GANs). Generally, GANs’ model consists of two different networks, which are trained alternately attempting to fit the underlying distribution of the input data from the training set. The discriminative model learns to determine whether the input data obey the generated distribution or the original distribution. The generative one learns from the original distribution and tries to produce fake samples that discriminative model cannot tell apart. Those two models achieve self-update through competition, with the result that it makes the generated distribution close to the original data distribution, and the produced samples are indistinguishable from the real samples.

The original intention of GANs is to avoid using Markov chain to cut down the amount of computation and create a mapping between the latent variable and underlying distribution. Among several years of developing, many variants of GANs have been presented aiming at various kinds of aspects. Some of them focus on image-to-image translation; some others try to generate high-resolution images; some are just committed to improving the performance of general GANs.

This paper is organized as follows: Section 2 describes the EL images data and their characteristics. The traditional data augmentation techniques applied on EL images also introduced in this section. The GAN architecture proposed is discussed in section 3. Section 4 presents several ways to use the generated images to improve the accuracy of classification, the corresponding results are also presented here. Section 5 discusses the application of the proposed method in the energy industry. At last, section 6 states the conclusion of this paper. The contribution of this work consists of two parts. First, we propose a GAN architecture called AC-PG GAN to learn the underlying distribution of EL images. Second, we present and compare several ways to use the generated images produced by the trained generator model in classification.

2. Dataset of EL images
The dataset we use in this paper contains 507 single channel EL images of monocrystalline solar cells including four kinds of defects, and all of them are collected from public EL images datasets.

![Figure 1](image1.png)

(a) Grid fingers   (b) Material defect   (c) Microcrack   (d) Deep cracks

Typically, the defects in solar modules can be divided into two main types [3], material-intrinsic defects and extrinsic defects. Defects like crystal grain boundaries and dislocations were caused by the material issue, in terms to affect the power efficiency of solar modules negatively. Extrinsic defects such as microcracks and breaks will reduce efficiency during the running time. Figure 1 shows examples of EL images with various kinds of defects in the dataset. Figure 1(a) and figure 1(b) show the grid fingers and material defect in EL images. Generally, finger interruptions are often treated as a kind of extrinsic defects, but the accurate relation between it and the drop of efficiency is still not explicit. Figure 1(c) and figure 1(d) show the microcrack and deep cracks which can absolutely influence efficiency. The dataset used in this paper contains 312 EL images of grid fingers, 30 of
material defect, 99 of microcrack and 66 of deep cracks. The original resolution of these EL images is 300 by 300 pixels and it is already a large scale in GAN application which brings great difficulties in the training procedure. The categories in the dataset are extremely unbalanced, and the amount is so small that it is not likely for a GAN model to train utterly. To make full of GAN model and its power of learning, we actually perform traditional data augmentation techniques on the original dataset beforehand to increase the amount. In order to maintain full characteristics of the original dataset and avoid adding too much irrelevant information, we only employing flip (horizontally, vertically), rotation (180 degrees, 360 degrees) and Color Jittering, augmenting data for around 11 times as the amount was.

3. AC-PG GAN

Due to the existence of high-resolution images in our dataset, we start with a GAN variant, the Progressive Growing of GANs [4] (PGGAN), to generate sharp images. Only a few variants of GAN can deal with high-resolution images (actually the resolution beyond 128 by 128 pixels is already tough for general GANs to learn) effectively, such as LAPGAN [5], BEGAN [6] and PGGAN, the one we refer to.

We follow PGGAN’s idea of training GAN model progressively, using different resolution samples to train the generator and discriminator progressively, starting with small resolution images and increase its size progressively. The advantage of this is that compared with training with high-resolution images in the first place, the strategy that start the training with low-resolution images and increase the size progressively will lead to much more stable results during the training. In this way, we can produce high-resolution samples while saving a bunch of video memory and computation.

![Figure 2. Artifacts in generated images](image)

![Figure 3. (a) ACGAN architecture. (b) CGAN architecture](image)

In the period between two progresses, that is after changing the size of the input, we have tried several kinds of interpolation method to mitigate the dramatic fluctuation caused by suddenly changing the input. We stick to the linear interpolation used in the original paper eventually, for the reason that other interpolations we have tried like quadratic interpolation will produce artifacts which
are visible to the naked eye. The examples of that are shown in figure 2. With unique progressive training procedure, GANs like PGGAN can generate more convincing fake samples, but like other GANs, PGGAN only trying to learn from the source data distribution in the dataset. Since the dataset contains more than one category, the data distribution mixed can be complicated, and learning of it can be difficult. Due to this, in order to mitigate mode collapse and enrich the diversity of the generated samples, we introduce the idea of Auxiliary Classifier GAN [7] (ACGAN) to our model to guide the generation of fake samples. The architecture of ACGAN is shown in figure 3a.

We refer to ACGAN and take its idea which is to add the categories label as additional information. This approach allows the model to be conditioned on external information and improve the image quality of the samples generated by GANs. Different from Conditional GAN [8] (CGAN), whose architecture is shown in figure 3b, the categories label information will not directly input into the discriminator of ACGAN, therefore it requires the discriminator not only have to judge the reliability of samples but also predict which class the samples belong. Adding this additional information make it easy to control the type of samples that we want to generate and it contributes to the classification later soon. If not, the generated samples will need to be marked manually. As demonstrated before, performing manual marking on the dataset containing EL images can be tough and also time-consuming as well as expertise-requiring.

![Generator architecture of AC-PG GAN](image)

**Figure 4.** Generator architecture of AC-PG GAN

Combining all we have expounded, the generator architecture of solar cell defects classification system called AC-PG GAN that we propose is shown in figure 4. This model is composed of several blocks, and each block consists of several convolution layers and upsampling layers. The main function of each block is to perform convolution on the input in order to get a feature map while adjusting the output to a specific size. The input is a combination of random noise (latent variable) and unique label information. In all progress, the input goes through the first block called Block_Pre with the primary purpose to narrow down the input size into 8 by 8 pixels as the basic unit for other blocks to proceed and perform upsampling during all progress.

The architecture of discriminator in the proposed model consists of several convolution layers and pooling layers. The output of the generator in one progress will be passed to the discriminator whose goal is to judge the reliability while outputting information about which class the input fake sample belongs.

By using the gradient penalty from WGAN-GP [9], the loss function of the discriminator is as follows:

\[
L(D) = -E_{r} [D(x, y)] + E_{r} [D(x, \hat{y})] - E_{r} [\|D_{r}(\hat{y}) \|_{1} + D_{z}(\hat{z})] + \sum_{i=1}^{N} y^{\ell(i)} \log \frac{1}{1 + \exp(-y^{\ell(i)})} + \frac{1}{2} \log(1 + \exp(\hat{y}^{\ell(i)}))
\]  

(1)
The first two parts in this equation are the Wasserstein loss first proposed in WGAN [10]. The next part is the gradient penalty proposed in WGAN-GP to replace the weight clipping which may constrain the weight of network to some certain values and damage the gradient in backpropagation. The last summation part is the cross-entropy loss, it measures the distance between the expecting label and the one which the discriminator outputs when input a real sample. The loss function of the generator is shown in equation (2).

$$L(D) = -E_{x \sim p} [D(x)] + \sum_{i=1}^{N} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$  \hspace{1cm} (2)$$

The cross-entropy loss in this one measures a similar distance when input a fake sample to the discriminator.

4. Image generation and classification

The training dataset is composed of EL images in the shape of 300 by 300 pixels. We produce samples in the shape of 256 by 256 pixels eventually for easy network designing. The input EL images are first resized to 256 by 256 pixels as input, and we apply stochastic gradient descent with the Adam optimizer and the learning rate we use is 0.0002. The training procedure continues until the improvement in the produced sample can barely be seen. The training procedure is processed on Pytorch 0.4 framework by an NVIDIA GeForce GTX 1080Ti graphics card.

The purpose of the whole GAN training is to augment the original dataset for better classification performance. We have tried several ways and examine the effectiveness of them by performing classification using several CNN models. The models chosen are AlexNet [11], ResNet [12] and SqueezeNet [13].

The dataset augmented by traditional data augmentation which we call D_A, contains 792 deep cracks, 1248 grid fingers, 360 material defect and 1344 microcrack EL images. The first thing we noticed is the imbalance in the deep cracks and material defect class. In order to balance the dataset for better CNN training, we generate 10000 images for each category and select some of them to fill the dataset which we call D_G. The examples of the generated EL images are shown in figure 5. To compare the two datasets and those performances in classification, three chosen models are trained on them respectively. The test dataset is the original dataset without any augmentation, and its images do not exist in both D_A and D_G, which is intentional, to avoid phenomena like improper training of GANs, and it can cause the generator to produce samples just like those in the dataset they are trained on. The samples produced under that circumstance often lead to much higher accuracy on the test dataset but are actually meaningless. The reason is that if the training dataset contains samples which already exist in the test set, the accuracy acquired on the test set, which is often higher, are in fact, not caused by the powerful classify ability but just the result of the excellent memory capability of CNN network, and it is exactly the researchers want to avoid in CNN model training.

![Figure 5. The generated EL images with defects](image)

(a) Grid fingers  (b) Material defect  (c) Microcrack  (d) Deep cracks

After selecting EL images from all the generated samples, we balance the material defect category and deep cracks category by adding the selected EL images. The dataset augmented (D_G1) contains 1262 deep cracks, 1248 grid fingers, 1359 material defect and 1344 microcrack EL images. Then we train three chosen models on it and compare the results with those from the models trained on the D_A. The results are shown in table 1 and table 2.
From the results in table 1 and table 2, we can see some improvements on those two augmented categories, but the decrease of accuracy of the microcrack is also noticeable. The reason may be the similar appearance between the microcrack in the original dataset and the deep cracks EL images that we produced. This problem can be solved by making the generator in the GAN model to generate finer images, but it requires more precise adjustments and attempts.

To further test this problem, on the base of the previous operation, we remove the added generated deep cracks EL images from D_G1 and get a new dataset (D_G2). The results of the classification are shown in table 3.

| Categories      | Test accuracy with D_A(%) | SqueezeNet | AlexNet | ResNet |
|-----------------|---------------------------|------------|---------|--------|
| Grid fingers    |                           | 96.47      | 95.80   | 99.04  |
| Microcrack      |                           | 54.55      | 65.66   | 91.92  |
| Material defect |                           | 40.00      | 53.30   | 96.67  |
| Deep cracks     |                           | 37.88      | 57.60   | 98.48  |

| Categories      | Test accuracy with D_G1(%) | SqueezeNet | AlexNet | ResNet |
|-----------------|---------------------------|------------|---------|--------|
| Grid fingers    |                           | 96.15      | 96.79   | 99.04  |
| Microcrack      |                           | 50.51      | 68.69   | 85.86  |
| Material defect |                           | 53.33      | 66.67   | 100.00 |
| Deep cracks     |                           | 42.42      | 66.67   | 98.48  |

| Categories      | Test accuracy with D_G2(%) | SqueezeNet | AlexNet | ResNet |
|-----------------|---------------------------|------------|---------|--------|
| Grid fingers    |                           | 96.15      | 94.87   | 98.72  |
| Microcrack      |                           | 53.54      | 70.71   | 94.95  |
| Material defect |                           | 53.53      | 66.67   | 100.00 |
| Deep cracks     |                           | 42.42      | 66.67   | 100.00 |

From the results in table 1 and table 2, it can be seen that after adding the generated material defect EL images in the dataset, all the classification performance are improved except a tiny small decrease in the grid fingers. The best improvement we get is in the material defect category with the maximum increase up to almost 14% achieved by AlexNet. The accuracy of the deep cracks category is also improved by nearly 10% to the maximum, and the microcrack category also gains some benefits with a few percent increases in accuracy. Compare with the results in table 2, we can make sure that the bad classification performance of microcrack is indeed caused by adding the imperfect generated deep crack EL images.

From all the results, the SqueezeNet performs the worst in those datasets and the ResNet achieves the best classification results in all categories with the highest accuracy. With the almost 100% correct results, the improvements in the results of ResNet are not that dramatical. The best improvement achieved by the AlexNet in view of its relatively good performance.

To better demonstrate the distribution of the generated images, we use the t-SNE visualization. The t-SNE algorithm for dimensionality reduction enables the embedding of high-dimensional data into a two-dimensional space. The results of them are shown in figure 6.
The results show that the generated samples follow the original distribution and fill the blanks in the distribution and make datasets more continuous.

5. Discussion
In the energy industry, especially when related to solar cells, the data samples like EL images tend to be difficult in acquiring. To obtain enough data samples for CNN or other deep learning networks can cost highly expensive equipment and a bunch of time. The method of data augmentation proposed in this paper provides a new way to avoid the troubles of acquiring experimental data. The AC-PC GAN model considering progressively training and label guidance is proposed to learn the underlying distribution of the EL images of solar cells and produce samples following the learned distribution. Addition to generating EL images, the model can also be adjusted to learn and produce solar cell images, current and voltage samples as well as I-V curve images.

But, there are still problems exist in its application. This model proposed requires the training of a GAN network which is known to be highly unstable during the training procedure. Besides, the training cost can be massive according to the scale of the required data, which limits the scope of its application. When this approach is used on a large scale in the energy industry, the training cost and the limitation should be fully considered for a suitable efficiency.

In the future, to make this approach more suitable for new energy industry application, the training cost needs to be cut down maybe with other suitable GAN structure to make this method for more different scales of data and the training procedure need to be further controlled by carefully adjusting the network to ensure the quality of generated data samples.

6. Conclusions
This paper proposed a new method of data augmentation to improve the performance of detecting defects in solar cells when using CNN models. Samples generated by the proposed AC-PC GAN model are valuable in the expanding of the dataset consisting of EL images.

In the experiments, all the three chosen CNN models achieve an increase of accuracy on the specific augmented categories in the test set, the maximum increase up to 14%. The results show that this augmentation method is effective and can be applied to the EL images classification of solar cells.

Acknowledgments
The work was supported by the National Natural Science Foundation of China (NSFC) (Nos. 61471124, 11404060), the Key Science and Technology Program of Fujian (No. 2016H0016), and the Natural Science Foundation of Fujian Province (No.2018J01667).

References
[1] Ben-Hur A, Horn D, Siegelmann H T et al 2000 A support vector clustering method Pro. 15th
Int. Conf. Pattern Recognition (ICPR) Barcelona 724-7

[2] Goodfellow I, Pouget-Abadie J, Mirza M et al 2014 Generative adversarial nets Advances in neural information processing systems (NIPS) Montreal Quebec Canada 2672-80

[3] Tsai D M, Wu S C, Chiu W Y 2013 Defect detection in solar modules using ICA basis images IEEE T. Industrial Informatics 9 122-31

[4] Karras T, Aila T, Laine S et al 2017 Progressive growing of gans for improved quality, stability, and variation (arXiv:1710.10196)

[5] Denton E L, Chintala S, Fergus R 2015 Deep generative image models using a laplacian pyramid of adversarial networks Advances in neural information processing systems (NIPS) Montreal Quebec Canada 1486-94

[6] Berthelot D, Schummi T, Metz L 2017 BEGAN: Boundary Equilibrium Generative Adversarial Networks (arXiv:1703.10717)

[7] Odena A, Olah C, Shlens J 2016 Conditional Image Synthesis With Auxiliary Classifier GANs (arXiv:1610.09585)

[8] Mirza M, Osindero S 2014 Conditional generative adversarial nets (arXiv:1411.1784)

[9] Gulrajani I, Ahmed F, Arjovsky M et al 2017 Improved training of wasserstein gans[C]. Advances in Neural Information Processing Systems (NIPS) Long Beach California USA 5767-77

[10] Arjovsky M, Chintala S, Bottou L 2017 Wasserstein gan (arXiv:1701.07875)

[11] Krizhevsky A, Sutskever I, Hinton G E 2012 ImageNet Classification with Deep Convolutional Neural Networks Advances in Neural Information Processing Systems (NIPS) Stateline Nevada USA 1097-105

[12] He K, Zhang X, Ren S et al 2016 Deep residual learning for image recognition Proceedings of the IEEE Conf. on computer vision and pattern recognition (CVPR) Las Vegas Nevada USA 770-8

[13] Iandola F N, Han S, Moskewicz M W et al 2016 SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size (arXiv:1602.07360)