



**ARTICLE**

## Deep Learning-Based Algorithm for Multi-Type Defects Detection in Solar Cells with Aerial EL Images for Photovoltaic Plants

Wuqin Tang, Qiang Yang and Wenjun Yan\*

College of Electrical Engineering, Zhejiang University, Hangzhou, 310027, China

\*Corresponding Author: Wenjun Yan. Email: yanwenjun@zju.edu.cn

Received: 15 July 2021 Accepted: 15 October 2021

### ABSTRACT

Defects detection with Electroluminescence (EL) image for photovoltaic (PV) module has become a standard test procedure during the process of production, installation, and operation of solar modules. There are some typical defects types, such as crack, finger interruption, that can be recognized with high accuracy. However, due to the complexity of EL images and the limitation of the dataset, it is hard to label all types of defects during the inspection process. The unknown or unlabeled create significant difficulties in the practical application of the automatic defects detection technique. To address the problem, we proposed an evolutionary algorithm combined with traditional image processing technology, deep learning, transfer learning, and deep clustering, which can recognize the unknown or unlabeled in the original dataset defects automatically along with the increasing of the dataset size. Specifically, we first propose a deep learning-based features extractor and defects classifier. Then, the unlabeled defects can be classified by the deep clustering algorithm and stored separately to update the original database without human intervention. When the number of unknown images reaches the preset values, transfer learning is introduced to train the classifier with the updated database. The fine-tuned model can detect new defects with high accuracy. Finally, numerical results confirm that the proposed solution can carry out efficient and accurate defect detection automatically using electroluminescence images.

### KEYWORDS

Electroluminescence images; deep clustering; automatic defect classification; transfer learning

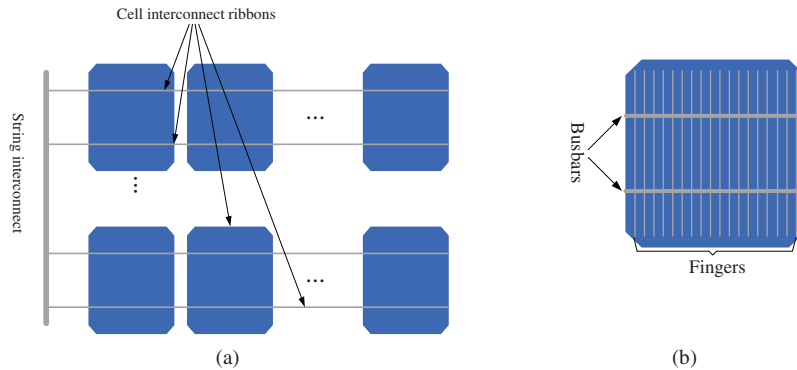
## 1 Introduction

In the past decades, the huge capacity of solar energy has been established around the world and the energy conversion efficiency of photovoltaic (PV) has achieved tremendous improvements year by year [1,2]. However, the conversion efficiencies can be impaired due to the long-time exposure under outdoor conditions that can cause long-term deterioration of PV module performance [3,4]. To detect the defects in the PV module, several physical algorithms are proposed. For example, current-voltage (IV) characteristics have been widely used to evaluate the status of the PV module [5–7]. However, IV curves could be scarcely influenced by some tiny defects that make it infeasible to accurately detect defects in PV modules [8]. In addition, infrared (IR) imaging is employed to detect the defects. The idea behind the IR imaging is that the temperature



in the defective area is higher than the normal area. However, such a technique cannot detect micro-defects due to the low resolution of IR images [9].

The inner structure of a silicon PV module is presented in Fig. 1. The current generated in a cell is collected and transferred through the busbars. The power generated in a PV module is the sum of all cells in the module. Therefore, the cell is a basic unit of a PV module and almost all of the defects in EL images are cell-level.



**Figure 1:** Inner structure of PV module and cell: (a) strings; (b) a cell

In recent years, EL is treated as an excellent technique to detect defects in PV modules. The idea behind it is that when a specified current is injected into the PV module, radiative recombination of carriers can create light emission and the light can be collected by the specified camera. Therefore, EL images can provide an inner perspective to evaluate the state of PV modules [10,11]. Given that the PV plants are always deployed in remote areas covering a large geographical area or some unsuitable places for human operation, the manual inspection with EL images is unfeasible, especially for large-scale PV plants. A set of studies for automatic defect detection using EL images has been carried out as follows. In [12], a series of image processing algorithms are proposed to extract cells from the PV module and then a well-trained CNN is used to recognize the crack. In [13], a public dataset of solar cells is provided that contains 2,624 solar cell images and two approaches are proposed to classify the EL images. In [14], a fusion model of Faster R-CNN and R-FCN is proposed to detect solar cell surface defects. In [15], an efficient method for defects inspection has been proposed that leverages the multi-attention network and the hybrid loss to improve the performance. In [16], a pipeline is developed to extract and classify the cell from the PV module. In [17], a deep learning-based defect detection of a photovoltaic module is proposed and GAN is introduced for the data augmentation. In [18], a novel complementary attention network is introduced that suppresses the noise feature in the background and enlarges the defects features simultaneously. In [19], a model is proposed to predict PV module electrical properties from EL image features using pixel intensity-based and machine learning-based classification algorithms. In [20], the detection of a crack in the PV module manufacturing system is presented and the proposed solution can identify the cells with cracks with high accuracy. In [21], the effect of crack distributions over a solar cell in terms of output power, short-circuit current density and open-circuit voltage was investigated. In [22], a new architecture that integrated fuzzy logic and convolution operations was proposed to suppress the subjectivity and fuzziness of defects recognition. In [23], an encoder-decoder network is proposed to perform semantic segmentation of EL images. In [24], a deep learning-based detection of

multi-type defects was proposed to implement inspection in the production line. In [25], a deep feature-based method was proposed to classify the defects. In this paper, the deep features were extracted by deep neural networks that were classified with machine learning methods.

The details of the previous work [12–25] are presented in Table 1. The limitations of these solutions can be summarized as follows: (1) Most images used in the previous studies are collected during the factory inspection and the resolution of the images captured during the factory inspection is generally much higher than those collected during the field inspection using the unmanned aerial vehicle (UAV). The solution used in the references could not be appropriate for the field inspection; (2) The dataset with sufficient images covering all defects can be hardly obtained, and the unknown or unlabeled defects may exist in the original dataset. This can degrade the defect detection performance; (3) Image annotation for the unknown or unlabeled defects can be time-consuming in practice. These limitations may significantly degrade the performance of automatic defect detection using EL images in terms of both efficiency and accuracy in large-scale photovoltaic plants.

**Table 1:** Defects detection of PV module with EL images

Reference/ Year	Model	Data in details		GPU		Detected defects types
		Train	Validation Test			
[12]/2018	LeNet	600	60	10	NVIDIA GTX1080	crack
[13]/2019	Adapted VGG	1968	–	656	–	material defects; finger interruption; micro-crack; degradation of cell inter-connection; electrically insulated cell parts
[14]/2020	Faster RCNN	861	–	600	NVIDIA GTX1080	broken cells; cracks; unsoldered areas;
[15]/2020	Proposed CNN	620	–	208	–	crack; finger interruption
[16]/2019	Proposed CNN	2840	–	710	–	crack; corrosion;
[17]/2020	Proposed CNN	6400	1600	600	NVIDIA TITAN	finger interruption; micro-crack; break;
[18]/2020	Proposed CNN	847	–	2782	–	finger interruption; crack; black core
[19]/2020	Proposed CNN	2841	711	889	GPU	busbar corrosion (5 levels)

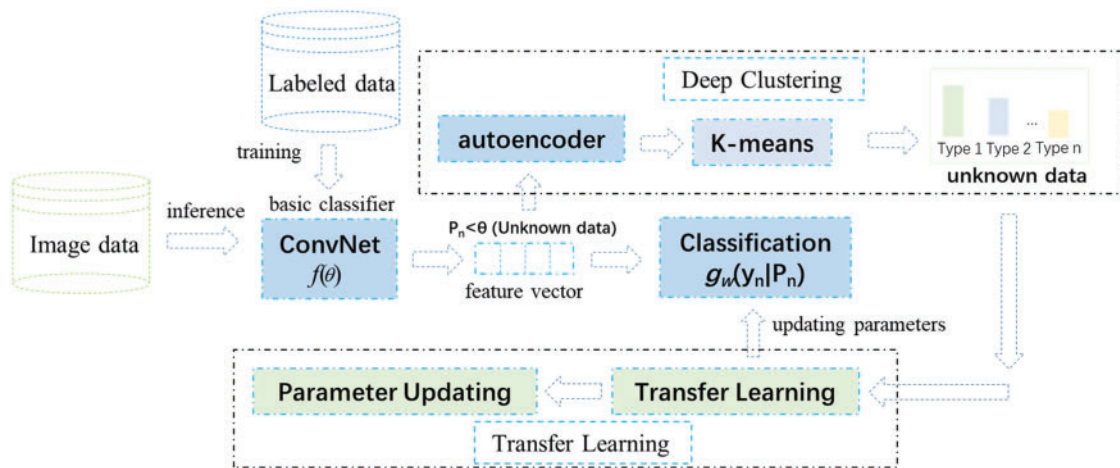
(Continued)

<b>Table 1 (Continued)</b>							
Reference/ Year	Model	Data in details				GPU	Detected defects types
		Train	Validation	Test			
[20]/2020	Traditional image process methods	–	–	–	–		crack
[21]/2021	manual	–	–	–	–		crack
[22]/2021	Proposed CNN	1679	420	525	NVIDIA TESLA P100		Defective; non-defective
[23]/2021	Proposed CNN	124	–	25	–		shunts; droplets
[24]/2021	Proposed CNN	3591	1196	1196	–		weak soldering; black area; scratch; finger interruption; crack; low cell; finger block; cell mix; disconnection; break; high cell; belt drop; bright mark; brightness saltation; foreign object; black edge; sucker mark
[25]/2021	Proposed CNN	2624			Nvidia GT-740 M		non-defected (0% defect rate), Possibly normal (33% defect rate), Possibly defected (66% defect rate) and Defected (100% defect rate)

To address the aforementioned challenges, a novel approach using the combination of deep learning, deep clustering, and transfer learning is introduced to establish our defects detection system of PV cells with EL image, which can recognize the labeled and unknown or unlabeled defects with high performance. Specifically, a deep learning-based features extractor and classifier are proposed to extract the deep features of images and classify the images. In addition, a deep clustering with unsupervised manner is designed to automatically classify the unlabeled or unknown defects. The structure of the CNN is adapted to improve computing efficiency. The framework of the proposed algorithmic solution is illustrated in Fig. 2.

The proposed defects detection system utilizes the phased design to address the outstanding technical challenges from both algorithm and system perspectives. The technical contributions made in this work are as follows:

- 1) A deep clustering algorithm is designed to cluster the different unknown or unlabeled defects based on the distance difference of the defects feature vector that is extracted by the well-trained CNN-based model. In addition, the deep clustering algorithm can be classified in an unsupervised manner;
- 2) A transfer learning algorithm is introduced to accurately detect the newly formed defects that are classified by the deep clustering algorithm. Data augmentation is applied to increase the performance of the CNN-based classification model;
- 3) To get the best trade-off between the performance and computing complexity, the effect of the model structure is considered and analyzed. The proposed model can obtain high accuracy with relatively simple computational complexity;
- 4) A real-world testbed with unmanned aerial vehicles (UAV) is built. To the best of the authors' knowledge, this work is the pioneering work to adopt the UAV for EL inspection in PV plants. In addition, a data set of EL images of PV modules with various defects is well established and maintained. The EL image can be shared when requested for non-commercial purposes.



**Figure 2:** Framework of the proposed defect detection algorithm

The remainder of the paper is organized as follows: [Section 2](#) presents the details of the proposed algorithm; In [Section 3](#), the performance of the proposed solution is evaluated through extensive experiments and the numerical results are presented and discussed; finally, the conclusive remarks and future work are given in [Section 4](#).

## 2 The Algorithm for Defects Detection

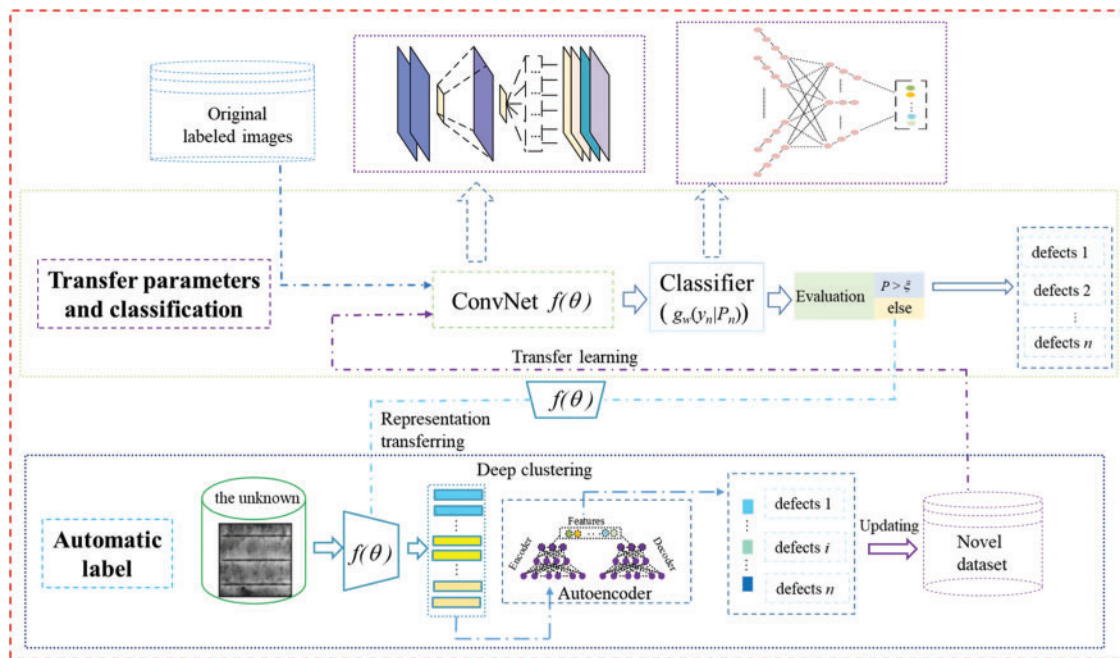
The framework contains several parts and steps that are presented in [Fig. 3](#) and the crucial parts are simply illustrated as follows:

**The feature extractor (ConvNet  $f(\theta)$ ):** The extractor is a refined CNN-based model to extract the feature vectors for defects classification or clustering. In the pre-training process, the classifier is firstly trained and obtains the original parameters with the original labeled images. To reduce the computational complexity, the parameters in ConvNet are frozen in the transfer learning period. The augmentation of the images is introduced to further improve the system performance.

**Classifier ( $g_w(y_n|P_n)$ ):** The classifier is a fully connected layer to classify the defects. In the pre-training period,  $g_w$  can get the initial parameter to distinguish the original labeled images and in the transfer learning process, the parameter in  $g_w$  is updated to classify the novel dataset. To increase the performance, the augmentation of the images is also introduced.

**Unknown defects detector:** Given that several defects in the collected dataset are unknown, we introduce a deep clustering algorithm to recognize the unknown defects without human intervention. A threshold  $\xi$  is used to distinguish the unknown defects. An autoencoder and  $k$ -means algorithms are designed to classify the unknown defects.

**Transfer learning and updating strategy:** To decrease the computing consumption, transfer learning is introduced to learn the feature of newly defects classified and the images with unknown defects can be classified automatically and transfer into the training set.



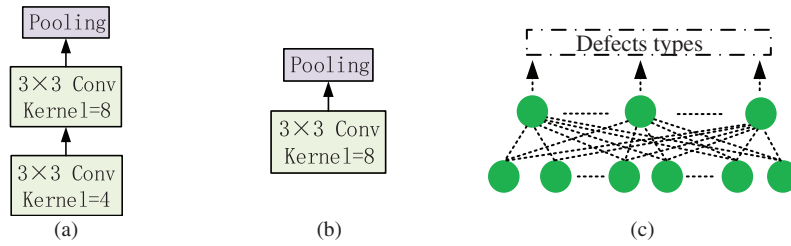
**Figure 3:** Details of the proposed defect detection algorithm

The details of the above algorithms are illustrated as follows.

### 2.1 The ConvNet and Defects Classifier

The aerial images of PV cells in the collected data set are represented as  $\{(I_1, p_1), (I_2, p_2), \dots, (I_n, p_n)\}$ , where  $I_m$  is the images collected at the mission point  $m$ . The images in the training set are denoted as  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , where  $x_m$  means the  $m$ th images in the training set and  $y_m$  is the class label of the  $m$ th images.

Motivated by [17], a novel structure of the deep learning-based model is proposed to extract the feature or representation of the image. The model for features extraction is denoted as ConvNet that contains two blocks: block A and block B and the classifier contains two fully connected layers. The structures of blocks A and B in ConvNet and the classifier are presented in Figs. 4a–c, respectively.



**Figure 4:** The structure of ConvNet and classifier: (a) block A in ConvNet; (b) block B ConvNet; (c) classifier  $g_w$

In fact, the deep learning-based model can be considered as a ConvNet mapping ( $f(\theta)$ ), where  $\theta$  is the parameters in the mapping. The vector obtained by applying the mapping is the representation or feature of the input image. To classify the image, a parametrized classifier  $g_w$  is built on the top of the ConvNet  $f(\theta)$ , where  $w$  is the parameter of the classifier. The label predicted by  $g_w$  is the image’s types to one of the  $k$  possible pre-defined classes. The purpose of the training process in the training set is to find the parameters  $\theta^*$  and  $w^*$  that the  $f(\theta^*)$  and  $g_{w^*}$  can get the good general features and recognize the defects with high accuracy. The parameters  $\theta$  and  $w$  are jointly updated by optimizing the following issue:

$$\min_{\theta, w} \left( \sum_j l(g_w(f_\theta(x_j)), y_j) \right) \tag{1}$$

where  $l$  is the cross-entropy loss function that is commonly applied to the classification task, as presented in (2):

$$E = - \sum_j t_k \log(y_k) \tag{2}$$

where  $y_k$  is the output of the deep learning-based model and  $t_k$  is the label of the image. The loss function is minimized by the mini-batch stochastic gradient descent. The output of a layer in the model can be computed as follows:

$$x^l = f(u^l), \text{ with } u^l = W^l \cdot x^{l-1} + b^l \tag{3}$$

where  $x^l$  is the output of the  $l$ th layer and  $W^l$  and  $b^l$  is the weight and bias in the  $l$ th layer. The updating of the weights can be calculated by (4) and (5)

$$W_{ij}^l = W_{ij}^{l-1} - \eta \frac{\partial E}{\partial W_{ij}^l} + v_{ij}^l \tag{4}$$

$$v_{ij}^l = \gamma v_{ij}^{l-1} - \lambda \eta w_{ij}^{l-1} \tag{5}$$

where  $\frac{\partial E}{\partial W_{ij}^l}$  is the gradient of the kernel weights and  $\eta$  is the learning rate,  $W_{ij}^l$  and  $v_{ij}^l$  denote the weight and momentum between the neuron  $i$  in  $l$ th layer and the neuron  $j$  in  $(l+1)$ th layer.

To improve the performance of the proposed model, data augmentation is widely applied to increase the size of the data. It has been proved in [17,25] that generative adversarial networks can create synthetic images and improve the performance of the deep learning-based method. However, due to the operation complexity of GANs and the size of the collected EL images

data, Gaussian blur and geometric transformation are performed in this paper to increase the number of training samples. The details of our data augmentation algorithm are presented in Algorithm 1.

---

**Algorithm 1**


---

**Input:** the original images  $I:(i_1, i_2, \dots, i_n)$

**Output:** the augmented images  $(i'_1, i'_2, \dots, i'_m)$

```

1  while  $i \in (i_1, i_2, \dots, i_n)$  do
    // Geometric transformation for the whole image in training data set
2  Initialization:  $\theta \in (-3^\circ \sim 3^\circ), (\Delta x, \Delta y) \in (-5\% \sim 5\%), k_i$ 
3  while pixel  $P_i \in I$  do
4      Geometric transformation // Eq. (6)
5      Get the new pixel  $P'$ 
6  end
    // Gaussian blur for the whole image in training data set
7  Initialize the standard deviation  $\delta$ 
8  while pixel  $P_i \in I'$  do
9      Gaussian blur //Eq. (7)
10     Get the new pixel  $P''$ 
11  end
12  Obtain the augmented image set
13 end

```

---

The grey value in the point  $(x, y)$  of image  $i$  is denoted as  $P_i(x, y)$  and the corresponding transformed grey value is denoted as  $P'_i(x, y)$ . The geometric transformation can be performed by (6)

$$\begin{cases} x' = k_{x1} \cdot x \cdot \cos \theta + k_{y1} \cdot y \cdot \sin \theta + \Delta x \\ y' = k_{y2} \cdot y \cdot \cos \theta - k_{x2} \cdot x \cdot \sin \theta + \Delta y \end{cases} \quad (6)$$

where  $\theta \in (-3^\circ \sim 3^\circ)$  is the rotation angle,  $k_{x1 \sim 2}, k_{y1 \sim 2}$  are the transformation coefficients and  $(\Delta x, \Delta y) \in (-5\% \sim 5\%)$  are the translation factors in axes  $x$  and  $y$ . The operation of Gaussian blur can be performed by (7)

$$\begin{aligned} P_i(x', y') &= P_i(x, y) \cdot G(x, y) \\ &= P_i(x, y) \cdot \frac{1}{2\pi\delta^2} \cdot e^{-(x^2+y^2)/2\delta^2} \end{aligned} \quad (7)$$

where  $G(x, y)$  is the Gaussian function and  $\delta$  is the standard deviation.

## 2.2 Unknown Defects Recognition and Training Set Upgrade Strategy

The images of PV cells captured by UAV are denoted as  $\chi: (x_1, x_2, \dots, x_n)$  and all  $\chi: (x_1, x_2, \dots, x_n)$  are transferred into the well-trained classifier that is described in Section 2.1. The well-trained classifier is represented by  $g_w(f(\theta))$  and the output of the classifier is the marginal probability distribution  $P$ . Therefore, the classification of a PV cell can be denoted as  $\{x_i, y_i, P_i\}$ , where  $x_i$  is an image of PV cell,  $P_i$  is the maximum probability of the predicted label  $y_i$ .



If the output of classifier  $P_i < \xi$ , where  $\xi$  is the preset threshold, it is reasonable to believe that  $x_i$  is an unknown defect. To classify the same type of unknown defects, a deep clustering algorithm (DC) is designed.

The proposed deep clustering algorithm contains two parts: (1) autoencoder; (2)  $k$ -means algorithms. The autoencoder is to extract the dominant features and consists of an encoder and decoder. The autoencoder is trained to minimize the reconstruction error, which is presented in (8)

$$l(z, z') = ||z - z'||^2 \quad (8)$$

---

**Algorithm 2**


---

**Input:** the original images  $I:(i_1, i_2, \dots, i_n)$

**Output:** the predicted label  $y_i$

```

1  while  $i \in (i_1, i_2, \dots, i_n)$  do
2      Put  $i_n$  into the well-trained model  $g_w(f(\theta))$  and get the probability  $P_n$  and the feature  $f(\theta)$ 
3      if  $P_i < \xi$  do
4          transmitting the image into a separate space  $\Omega$  and labeled unknown
5      end
        //the first process of predication is finished
6  end
7  Train the autoencoder by (8)
      while  $i' \in \Omega$  do
8      Put the feature  $f(\theta)$  into the  $k$ -means clustering by (9)
9      while  $s(k)$  is increasing do
10          $k=k+1$ 
11         computing the score of the clustering  $s(k)$  by (10)
12     end
13     get the max  $s(k)$  and  $k$  types of the unknown defects
14     separate the  $k$  types defects and store them into the database
15     updating the training set
16 end

```

---

Preliminary research [26] indicates the choice of the clustering algorithm is not crucial and in this paper, we introduce a standard clustering algorithm,  $k$ -means, to group the features extracted by ConvNet. In our cases, the features  $f_\theta(x_n)$  computed by ConvNet are clustered into  $k$  groups. More specifically,  $k$ -means for the features is to minimize the within-cluster  $k$  square as presented in (9)

$$\min \sum_{j=1}^k \sum_{z \in S_i} ||z - u_i|| \quad (9)$$

where  $u_i$  is the arithmetical mean of points in the nearest or same center set  $S_i$ . The pseudocode of defect detector and training set updating strategy is presented in Algorithm 2.

$s(k)$  is introduced to calculate  $k$ , as given in (9). More specifically, in (10), a smaller  $W_k$  and larger  $B_k$  indicate higher  $s(k)$ , and hence a better clustering performance. Therefore, when max

$s(k)$  is obtained,  $k$  can be calculated.

$$s(k) = \frac{\text{tr}(B_k)}{\text{tr}(W_k)} \cdot \frac{m-k}{k-1} \quad (10)$$

where  $B_k$  is the trace of the between group dispersion matrix and  $W_k$  is the covariance matrix of data in the same type, and  $\text{tr}$  is the trace of the matrix.

### 2.3 Detection of the Unknown Defects Using Transfer Learning

To decrease the computing complexity, transfer learning is introduced to train the classification model. In this paper, the labeled training set is considered as the source domain  $D_s = \{\chi, P(X)\}$ , where  $\chi$  is the feature space of the labeled PV cell images and  $P(x)$  is marginal probability. Similarly, the target domain can be denoted as  $T = (x, f(\cdot))$ , where  $x$  is the feature space in the target domain and  $f(\cdot)$  is the target predictive function. The purpose of our design is to improve the learning process of predictive function  $f(\cdot)$  using the knowledge in  $D_s$ .

The earlier layers in the deep learning-based model are to extract the generic features from images, e.g., colors, edges and shapes, while the deeper layers are more likely to learn the more abstract features for classification [27]. Depending on the defects features in the EL images, it is optional to fine-tune the last full connected (FC) layer in the classifier and freeze the earlier layer in the ConvNet. ConvNet is used as the feature extractor for the image representation that is pre-trained on the given training set.

In the training process, the Adam optimizer is used for backpropagation and the categorical cross-entropy is used as the loss function that is presented in Eq. (2). The details of the process are described in Algorithm 3. In this paper, the classifier is presented in Fig. 4c.

---

#### Algorithm 3

---

**Input:** the pre-trained model

the clustered unknown images  $x_i, y_i, P_i$

**Output:** the predicted label  $y_i$

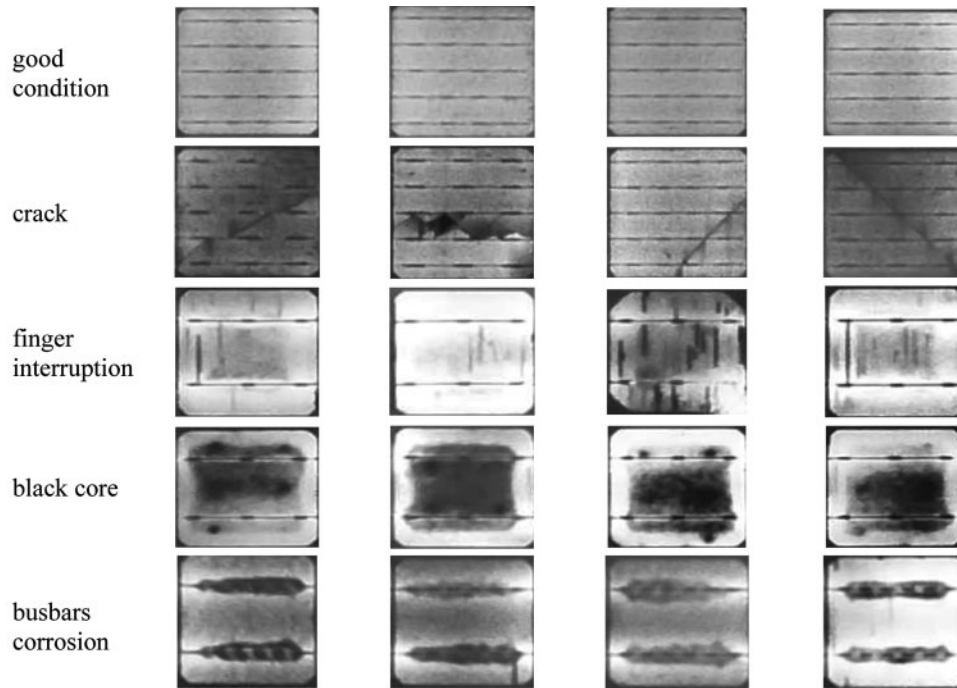
- 1 **while**  $x \in (x_1, x_2, \dots, x_n)$  **do**
  - 2   freeze the parameters in the well-trained ConvNet
  - 3   Calculate the gradient and update the weight and threshold by Eq. (2)~Eq. (5)
  - 4 **end**
  - 5 Save the parameter in the classifier
- 

### 3 Experimental Assessment and Numerical Results

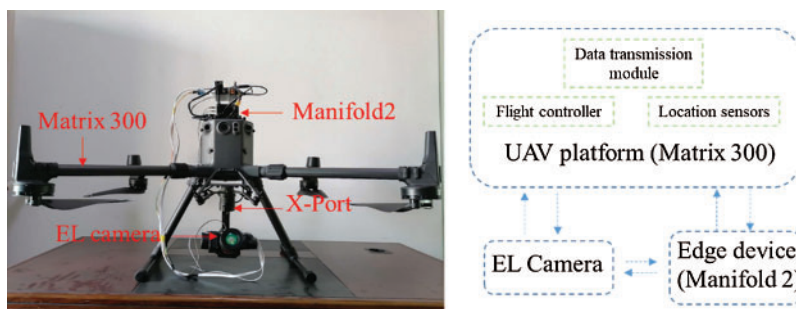
In this section, numerical experiments have been performed to evaluate the performance of the proposed algorithms. In this paper, the collected paper contains EL images with several types of cell conditions: good condition, crack, finger interruption, black core, busbars corrosion, which is presented in Fig. 5.

The data set is collected with a UAV-based platform in the real PV plants in Gonghe, Qinghai Province, China (100 MW). The platform is presented in Fig. 6. In the platform, four-rotor aircraft, DJI M300 is employed and some others accessories to establish the platform contain EL camera, X-port and Manfold2. In addition, a high-performance computation platform (with 4

NVIDIA TITAN GPUs) is used to decrease the computing time during the training process. The installed software for implementing the computing are Python and Pytorch.



**Figure 5:** Sample images of PV cells in different conditions

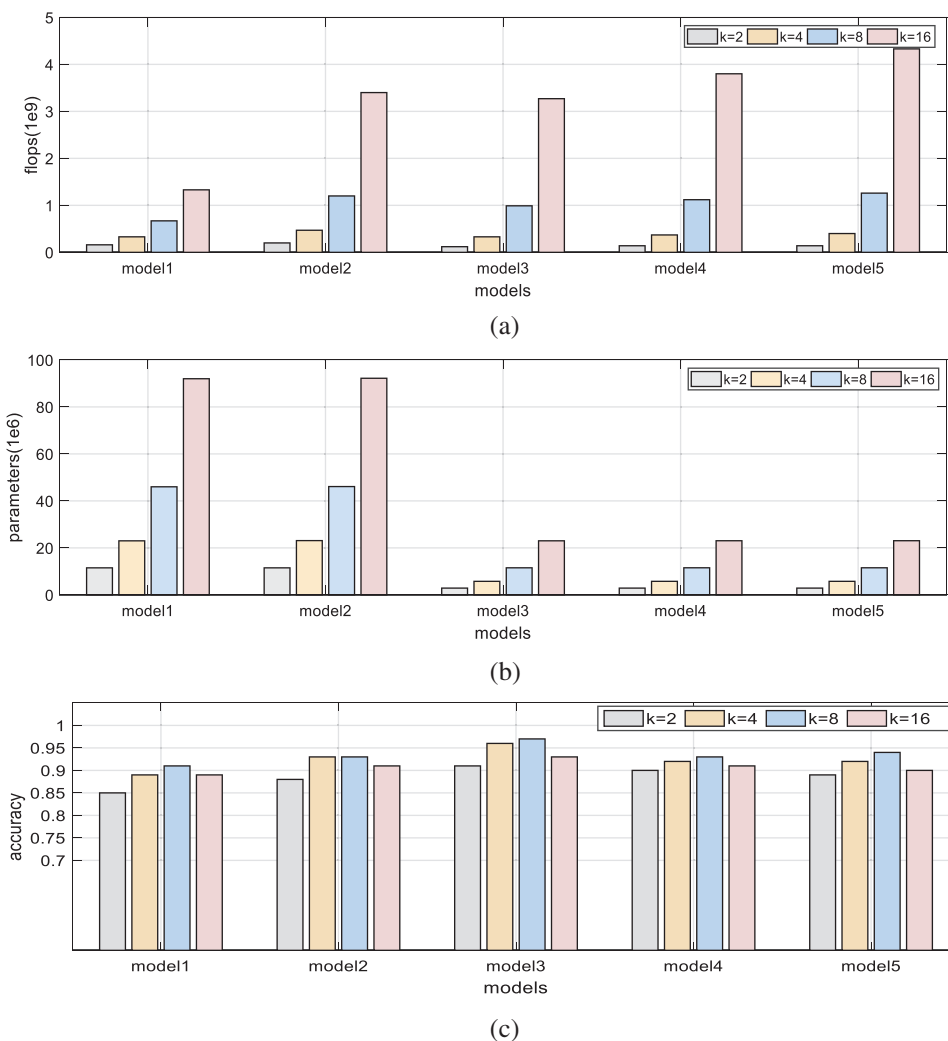


**Figure 6:** The UAV-based platform

### 3.1 Effects of the Models Using Different Structures

In this section, we evaluate the effects of the model with different structures. As mentioned previously, the features extracted from earlier layers in the deep learning-based model could include more information about color, edge or texture and the features extracted by the deeper layer can be more abstract. The experiments are carried out to evaluate the performance of these models with different structures and parameters, and the numerical results are presented in Fig. 7. Here, in total five models, i.e.,  $(c1p1)$ ,  $(c1c2p1)$ ,  $(c1c2p1c3p2)$ ,  $(c1c2p1c3c4p2)$  and  $(c1c2p1c3c4c5p2)$ , are extensively compared and analyzed. These models are represented by

Model 1~Model 5, where  $(ci)$  and  $(pi)$  are the  $i$ th convolution layers and the  $i$ th pooling layer, respectively. At the end of the model, two fully connected layers are adopted to classify the defects. In addition, given that the feature mappings extracted by different kernels can be different, the impact of the kernel ( $k=2, 4, 8, 16$ ) in the convolution layer is also analyzed.



**Figure 7:** Experimental evaluation of CNN model with different parameter settings: (a) flops of different models; (b) parameters of different models; (c) mean accuracy of defect detection

It can be inferred from Figs. 7a and b that with the increasing of the kernel number, the parameters and flops grow rapidly. It means that the computing process has become more complicated along with the increase of kernel number. When the number of kernels is doubled, the parameters and flops are about doubled. The results presented in Fig. 7c illustrate that the growth of the number of kernels from 2 to 8 can increase the accuracy because more kernels can provide more deep features. However, we also noticed that when the number of kernels is beyond 8, the performance could be damaged. It is indicated that the convolution layer with 4 or 8 kernels is sufficient to extract the defects features in EL images. In addition, an additional number of

kernels could increase the computing complexity greatly. Therefore, model 3 is selected to classify the defects.

### 3.2 Performance of Classification Accuracy and Implementation Details

In this section, we trained the proposed deep learning-based model for 160 epochs. Once the training process is completed, the parameters with the lowest validation loss are loaded to the model for test. To evaluate the performance of the proposed solution, a 10-fold cross-validation scheme is introduced. The original dataset, which contains 5,000 EL images (1,000 images for each type of condition), is equally divided into ten parts and each part contains 100 images for each type of condition. In the experiment, seven parts, two parts and one part are selected randomly as the training, validation and test set. The mean accuracy and standard deviation for each type are presented in Table 2.

**Table 2:** The mean accuracy and standard deviation in the experiments

	Good condition	Finger interruption	Crack	Busbar corrosion	Black core
Mean accuracy	0.98	0.93	0.94	0.96	0.96
Standard deviation	0.01	0.017	0.017	0.011	0.009

To show the performance of the proposed methods, the confusion matrix is presented in Fig. 8.

		Prediction label				
		good condition	finger interruption	crack	busbar corrosion	black core
True label	good condition	98	2	0	0	0
	finger interruption	1	93	3	2	1
	crack	1	3	94	1	1
	busbar corrosion	0	1	1	96	2
	black core	0	1	2	1	96

**Figure 8:** Classification confusion matrix

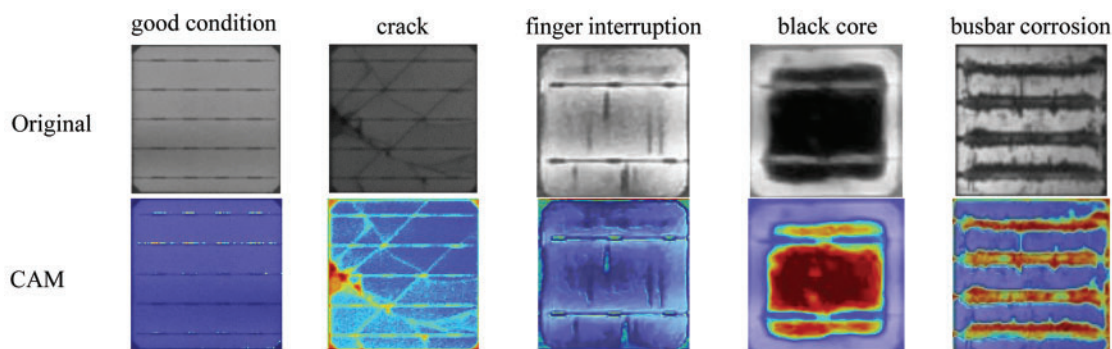
The classification accuracy of different defects is presented in Table 2. The classification accuracy of the proposed model is all beyond 0.9 and the mean accuracy of defects classification in the 10-fold cross-validation scheme is 0.95. The accuracy of finger interruption is the lowest (0.93). This is due to that some features of finger interruption could be covered by the background.

To analyze the features extraction ability of the proposed model, class activation maps (CAM) are introduced to visualize the learned features [28]. For easy explanation, the feature map of the top convolution layer is used as the input to create the heat map. Fig. 9 shows the visualization of the extracted features map. It can be concluded from Fig. 9 that the proposed model can extract the defects feature with high performance.

### 3.3 Performance of Unknown Defects Detector

In this part, the proposed deep clustering method is evaluated with the existing methods. In the experiments, crack and finger interruption are treated as the known defects as well as the cells in good condition, while busbars corrosion and black core are considered as the unknown defects. 2,400 images (about 800 images in each labeled type) are used to train the ConvNet and 200 images of the cells with busbars corrosion and black core are used to test the deep clustering.

The pre-trained ConvNet is used to extract the 256-D features from the input images. Then,  $k$ -means is employed to separate the features extracted from the unknown images and  $k$  can be determined by Eq. (10). All the hyperparameters in the models are set randomly and the learning rate is 0.0001. In Table 2, scores of the proposed deep clustering (DC) with different  $k$  values are obtained. It can be observed that the highest score is calculated when  $k$  is 2, which matches the experiment setup.



**Figure 9:** Visualization of the feature mappings

To evaluate the performance of DC, we introduce another two algorithms to implement the same task. One directly used  $k$ -means to cluster the images and another is called the combination of the principal component analysis with  $k$ -means (PCA+ $k$ -means) to cluster the original images. All the relevant settings in the experiments for the three methods are the same. Here, three commonly used metrics, i.e., V-measure, adjusted mutual information (AMI) and adjusted rand index (ARI), is introduced. The larger value of the metric indicates better performance. The results are all presented in Table 3 and it can be concluded that the proposed DC outperforms the other two benchmark methods in the three metrics most of the time. This is due to the fact that the direct  $k$ -means treat all the information as the same during the processing and the defects features could be covered by numerical noises, which can greatly impact the clustering results and although PCA+ $k$ -means can reduce the dimensionality of the features, PCA is not applicable to extract the features in EL image. In addition, the performance of the methods with the different number of clusters ( $k = 2, 3, 4$ ) is presented in Table 3 and it can be observed that when  $k = 2$ , the best results are obtained, which matches the experimental setup.

**Table 3:** Comparison with different clustering methods and  $k$  values

	$s(k)$	V-measure	AMI	ARI
DC, $k = 2$	125.8	0.82	0.89	0.78
$k$ -means, $k = 2$	67.3	0.53	0.51	0.47
PCA+ $k$ -means, $k = 2$	70.4	0.49	0.52	0.44

(Continued)

**Table 3 (Continued)**

	$s(k)$	V-measure	AMI	ARI
DC, $k = 3$	83.4	0.72	0.69	0.67
$k$ -means, $k = 3$	51.7	0.51	0.41	0.43
PCA+ $k$ -means, $k = 3$	50.5	0.53	0.42	0.41
DC, $k = 4$	78.5	0.73	0.54	0.61
$k$ -means, $k = 4$	46.4	0.49	0.32	0.47
PCA+ $k$ -means, $k = 4$	44.7	0.51	0.34	0.42

### 3.4 Comparison with the Existing Methods

In this section, we compared the proposed method with the state-of-art approaches, namely, VGG16 [29], MobileNet [30], ResNet50 [31] and InceptionV3 [32]. To make these models more appropriate to the dataset, the last two layers in these models are replaced by two connected layers ( $1 \times 1 \times 256$ ,  $1 \times 1 \times 5$ ). We do not change the other parameters in these models and the parameters are presented in [29–32].

In this work, accuracy, and trainable parameters in different models are considered and analyzed, which is presented in detail in Table 3. Those methods can extract the defects features with the combination of some convolution and fully connected layers. As presented in Table 4, the well-trained InceptionV3 can obtain a better accuracy of 92%, but the trainable parameters in the model are larger than the proposed method. Overall, the proposed method outperforms the above-mentioned models in the defect classification of EL images and provides the best trade-off between computational complexity and classification performance.

**Table 4:** Performance comparison with other models

	Proposed	VGG16	MobileNet	ResNet50	InceptionV3
Trainable Params	11 M	23 M	20 M	57 M	55 M
Accuracy	0.96	0.87	0.9	0.89	0.92

## 4 Conclusions and Future Work

This paper proposed a framework for the application of deep learning to address the problems of PV cells defects detection with EL images. A well-trained feature extractor (ConvNet) and classifier are obtained and given that there are some unknown defects during the inspection process, a deep clustering technique is proposed to distinguish the unknown or unlabeled defects without human intervention. To relieve the limitation of insufficient data, image augmentation is used. In addition, transfer learning is adopted to transfer the defects feature map to the target domain. The proposed algorithmic solution is evaluated extensively under different operational scenarios. The experiment results prove the accuracy and efficiency of the model.

For future work, the algorithms can be deployed in the edge devices for online defect detection and more types of defects types need to be considered and investigated. In addition, topological optimization can be introduced to obtain the best trade-off between computing accuracy and complexity.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

1. Green, M. A., Dunlop, E. D., Hohl-Ebinger, J., Yoshita, M., Kopidakis, N. et al. (2020). Solar cell efficiency tables (version 56). *Progress in Photovoltaics Research & Applications*, 28(7), 629–638. DOI 10.1002/pip.3303.
2. Polly, S. J., Dann, M., Fedorenko, A., Hubbard, S., Landi, B. et al. (2018). Development of a nano-enabled space power system. *IEEE 7th World Conference on Photovoltaic Energy Conversion*, pp. 3389–3391, Waikoloa.
3. Sharma, V., Chandel, S. S. (2013). Performance and degradation analysis for long term reliability of solar photovoltaic systems: A review. *Renewable & Sustainable Energy Reviews*, 27, 753–767. DOI 10.1016/j.rser.2013.07.046.
4. Schmidt, M., Braunger, D., Schaffler, R., Schock, H. W., Rau, U. (2000). Influence of damp heat on the electrical properties of Cu(in,Ga) Se<sub>2</sub> solar cells. *Thin Solid Films*, 361, 283–287. DOI 10.1016/S0040-6090(99)00820-2.
5. Chegaar, M., Azzouzi, G., Mialhe, P. (2006). Simple parameter extraction method for illuminated solar cells. *Solid-State Electronics*, 50, 1234–1237. DOI 10.1016/j.sse.2006.05.020.
6. Mohamed, S., Halim, B., Elgorashi, T., Elmirghani, J. (2020). Fog-assisted caching employing solar renewable energy and energy storage devices for video on demand services. *IEEE Access*, 8, 115754–115766. DOI 10.1109/Access.6287639.
7. Bouzidi, K., Chegaar, M., Bouhemadou, A. (2017). Solar cells parameters evaluation considering the series and shunt resistance. *Solar Energy Materials and Solar Cells*, 91(18), 1647–1651. DOI 10.1016/j.solmat.2007.05.019.
8. Osawa, S., Nakano, T., Matsumoto, S., Matsumoto, S., Katayama, N. (2016). Fault diagnosis of photovoltaic modules using AC impedance spectroscopy. *IEEE International Conference on Renewable Energy Research and Applications*, pp. 210–215, Birmingham.
9. Paggi, M., Berardone, I., Infuso, A. (2014). Fatigue degradation and electric recovery in silicon solar cells embedded in photovoltaic modules. *Scientific Reports*, 4, 4506. DOI 10.1038/srep04506.
10. Köntges, M., Kurtz, S., Packard, C. E., Jahn, U., Berger, K. A. et al. (2014). Review of failures of photovoltaic modules. *Technical Report*. <http://repository.supsi.ch/id/eprint/9645>.
11. Breitenstein, O., Bauer, J., Bothe, K. (2011). Can luminescence imaging replace lock-in thermography on solar cells? *IEEE Journal of Photovoltaics*, 1(2), 159–167. DOI 10.1109/JPHOTOV.5503869.
12. Banda, P., Barnard, L. (2018). A deep learning approach to photovoltaic cell defect classification. *Annual Conference of the South African Institute of Computer Scientists and Information Technologists: Technology for Chang*, pp. 215–221, Port Elizabeth, South Africa.
13. Deitsch, S., Christlein, V. (2019). Automatic classification of defective photovoltaic module cells in electroluminescence images. *Solar Energy*, 185, 455–468. DOI 10.1016/j.solener.2019.02.067.
14. Zhang, X., Hao, Y., Wang, A. (2020). Detection of surface defects on solar cells by fusing multi-channel convolution neural networks. *Infrared Physics & Technology*, 108, 10334–10351. DOI 10.1016/j.infrared.2020.103334.
15. Rahman, M., Chen, H. (2020). Defects inspection in polycrystalline solar cells electroluminescence images using deep learning. *IEEE Access*, 8, 40547–40558. DOI 10.1109/Access.6287639.
16. Karimi, A. M., Fada, J. S., Hossaine, M. A., Yang, S. Y., Peshek, T. J. et al. (2019). Automated pipeline for photovoltaic module electroluminescence image processing and degradation feature classification. *IEEE Journal of Photovoltaics*, 9(5), 1324–1335. DOI 10.1109/JPHOTOV.5503869.



17. Tang, W., Yang, Q., Xiong, K. X., Yan, W. J. (2020). Deep learning based automatic defect identification of photovoltaic module using electroluminescence images. *Solar Energy*, 201, 453–460. DOI 10.1016/j.solener.2020.03.049.
18. Su, B., Chen, H., Chen, P. (2020). Deep learning-based solar-cell manufacturing defect detection with complementary attention network. *IEEE Transactions on Industrial Informatics*, 17(6), 4084–4095. DOI 10.1109/TII.9424.
19. Karimi, A. M., Fada, J. S., Parrilla, N. A., Pierce, B. G., Koyuturk, M. et al. (2020). Generalized and mechanistic PV module performance prediction from computer vision and machine learning on electroluminescence images. *IEEE Journal of Photovoltaics*, 10(3), 878–887. DOI 10.1109/JPHOTOV.5503869.
20. Dhimish, M., Mather, P. (2020). Ultrafast high-resolution solar cell cracks detection process. *IEEE Transactions on Industrial Informatics*, 16(7), 4769–4777. DOI 10.1109/TII.9424.
21. Dhimish, M., Alessandro, V., Daliento, S. (2021). Investigating the impact of cracks on solar cells performance: Analysis based on nonuniform and uniform crack distributions. *IEEE Transactions on Industrial Informatics*, 99, 1–9. DOI 10.1109/TII.9424.
22. Ge, C., Liu, Z., Fang, L. (2021). A hybrid fuzzy convolutional neural network based mechanism for photovoltaic cell defect detection with electroluminescence images. *IEEE Transactions on Parallel and Distributed Systems*, 32(7), 1653–1664. DOI 10.1109/TPDS.2020.3046018.
23. Sovetkin, E., Achterberg, E. J., Weber, T. (2021). Encoder–decoder semantic segmentation models for electroluminescence images of thin-film photovoltaic modules. *IEEE Journal of Photovoltaics*, 11(2), 444–452. DOI 10.1109/JPHOTOV.5503869.
24. Zhao, Y., Zhan, K., Wang, Z. (2021). Deep learning-based automatic detection of multitype defects in photovoltaic modules and application in real production line. *Progress in Photovoltaics Research and Applications*, 29(4), 471–484. DOI 10.1002/pip.3395.
25. Demirci, M. Y., Beli, N., Gümüü, A. (2021). Efficient deep feature extraction and classification for identifying defective photovoltaic module cells in Electroluminescence images. *Expert Systems with Applications*, 175(2), 114810. DOI 10.1016/j.eswa.2021.114810.
26. Caron, M., Bojanowski, P., Joulin, A. (2018). Deep clustering for unsupervised learning of visual features. *European Conference on Computer Vision*, pp. 139–156, Munich, Germany.
27. Srinivas, S., Sarvadevabhatla, R. K. (2016). A taxonomy of deep convolutional neural nets for computer vision. *Frontiers in Robotics and AI*, 2, 36. DOI 10.3389/frobt.2015.00036.
28. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. et al. (2017). Grad-cAM: visual explanations from deep networks via gradient-based localization. *IEEE International Conference on Computer Vision*, pp. 618–626, Venice, Italy.
29. Simonyan, K., Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, San Diego.
30. Howard, A., Zhu, M., Chen, B. (2017). Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.
31. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas.
32. Szegedy, C., Liu, W., Jia, Y. (2015). Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston.