ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20250320002496
Original scientific paper

Solar Cell Anomaly Detection by ResNet Work with SE Attention Module

Tianyi REN, Yatong ZHOU*

Abstract: Aiming at the limitations of low detection accuracy and low detection efficiency of current anomaly detection methods for solar cells, especially the insufficient detection of subtle defects such as cracks and finger failures, an anomaly detection model SE-PatchCore is proposed for anomaly detection of solar cells in electroluminescence (EL) images by adding attention module in ResNet. Firstly, the SE (Squeeze and Excitation) attention mechanism is introduced into ResNet, and the network is applied to the advanced local anomaly detection model PatchCore. Finally, the Fair k-Centers method is used for coreset subsampling. Through testing on ELPV dataset, the score of SE-PatchCore in image-level AUROC is up to 98.7%, and the score of pixel-level AUROC is up to 98.5%, which has an improvement of 0.2% and 0.6% compared with PatchCore. And it has higher improvement over other methods such as SPADE and PaDiM. A higher AUROC score means the model can more accurately distinguish defective solar cells from normal ones and reduce false positives and false negatives. This helps to cut down on unnecessary rework, lower production costs, and ensure that more high-quality solar cells reach the market, meeting the increasing demand for reliable renewable energy products. The above results show that the introduction of SE attention module and the Fair k-Centers method can effectively improve the accuracy of anomaly detection of solar cells in EL images, which provides strong technical support for solving problems in this field.

Keywords: anomaly detection; attention mechanism; deep learning

1 INTRODUCTION

In today's world, new energy technologies are developing rapidly, with new energy products such as solar cells expanding their application fields and permeating various aspects of production and life. It is inevitable that a certain proportion of defects will occur in the production process of solar cells, such as cracks, dark areas and finger failures. Implementing defect detection through technical means to identify surface defects is an important procedure to ensure the quality of solar cell production and meet customers' requirements. Among various defect detection techniques, analyzing electroluminescence (EL) images is an effective method for identifying defects such as cracks and finger failures. Generally, the detection of defects involves visually inspecting the EL images, which are inefficient, costly, and have a detection accuracy that is significantly influenced by human factors, making it difficult to meet the requirements of large-scale and high-precision quality inspection. Therefore, the automation of visual inspection through the use of image recognition technology is anticipated to mitigate these challenges. Although numerous automated methods [1-7] have been developed for the defect detection of solar cells, existing methods inadequately detect subtle defects like cracks or finger failures due to insufficient attention to local features, limiting both detection accuracy and efficiency.

Recently, deep learning has enabled notable breakthroughs in the area of image recognition. The use of deep learning in anomaly detection has led to models with superior performance, solving various complex pattern recognition problems that were previously difficult to address. Not only does this hold incredible significant research value, but it also shows a broad application prospect, gradually becoming one of the important directions for both foundational research and practical application in the field of smart manufacturing. Bartler et al. [8] were the first to apply Convolutional Neural Networks (CNN) to the detection of solar cells in EL images. Hussain et al. [9] proposed Gradient Guided Filter Tuning (GGFT) and Filter Fused Data Scaling (FFDS) mechanisms, which are integrated with a CNN for the detection of surface cracks in solar cells. Korovin et al. [10] introduced a deep learning-driven automated detection framework named SeMaCNN, which is designed for the classification and anomaly detection of EL images to assess solar cell quality. Hassan et al. [11] introduced a deep learning network-based architecture (DSMP-CNN) for defect segmentation in polycrystalline solar cells. By incorporating a dual spin pooling mechanism, DSMP-CNN effectively unites the positive attributes of mean pooling and max pooling, which improved accuracy in crack detection. Beyond CNN-based methods for solar cell inspection, Generative Adversarial Network (GAN) [12, 13], YOLO series models [14, 15], unsupervised adversarial training framework [16] and Large-Scale Visual Language Model [17] are used to detect solar cell anomality.

Notably, the human visual system is characterized by its ability to focus on parts of a scene without trying to process the entire image at one time. Instead, humans use a sequence of partial glances and concentrate on key regions in a selective manner to enhance the understanding of visual structure [18]. Therefore, employing attention mechanisms in image recognition serves as an effective method to select or emphasize weak features of objects while suppressing noise information in the image. In the context of solar cell EL images, with their complex and non-uniform backgrounds, attention mechanisms offer great potential for improving anomaly detection. Several studies have integrated attention mechanisms into solar cell inspection models [19-21], but there is still room for enhancing their ability to handle subtle defects and improve overall detection performance.

To address these gaps, we propose an EL image inspection method for solar cells called SE-PatchCore, which integrates the Squeeze-and-Excitation (SE) attention module [22] into the ResNet architecture. Simultaneously, this network is incorporated into the detection pipeline of PatchCore [23], employing the approach of constructing a coreset by extracting features from defect-free image patches to achieve efficient anomaly detection in products. Furthermore, we use a more effective coreset subsampling method, the Fair k-Centers as suggested in [24], to replace the original iterative greedy approximation [25]. Our

research aims to develop a more accurate and efficient model for detecting cracks, dark areas, and finger failures in solar cell EL images, thereby making a significant contribution to the field of solar cell quality inspection. The contributions of this article are as follows.

1) The SE attention module is integrated into PatchCore [23], leveraging global information to selectively emphasize informative features and suppress less useful information.
2) SE-PatchCore is capable of computing relationships among features without associating training processes, which are typically found in traditional self-attention models.
3) The more efficient Fair k-Centers method was used to replace the original iterative greedy approximation for solving the coreset subsampling problem.

2 RELATED WORKS

2.1 Reconstruction-Based Method

The reconstruction-based methods for deep learning anomaly detection involve training models, such as autoencoders [26] and GANs, to learn the feature representations of normal data. These learned representations are then used to reconstruct new data. If the new data is anomalous, it will not be accurately reconstructed by the model, and thus anomalies can be identified through reconstruction error. In the scenario utilizing autoencoders, Chen et al. [27] proposed an Autoencoder-based network anomaly detection method that leverages the ability of Autoencoders to capture nonlinear correlations between features, thereby enhancing detection accuracy. Zhou et al. [28] introduced a new deep autoencoder that not only retains the capability of discovering high-quality nonlinear features but also removes outliers and noise without requiring normal training data. In the case based on GAN, Tang et al. [29] developed a neural network for anomaly detection called Dual Autoencoder Generative Adversarial Network (DAGAN), which addresses the issue of sample imbalance. Song et al. [30] proposed a new anomaly segmentation network (AnoSeg) that can generate accurate anomaly maps directly through self-supervised learning.

2.2 Representation-Based Method

The representation-based anomaly detection involves training deep learning models on normal data to capture the intrinsic characteristics of such data within a learned feature space. Once this representation is established, the system can assess new incoming data points by measuring how well they fit into the established pattern of normalcy. If a data point significantly deviates from what is considered normal based on its position in the feature space, it is flagged as an anomaly. This approach leverages various deep learning architectures, including but not limited to autoencoders for reconstructing input data [31], GANs for generating and distinguishing between real and fake samples [32, 33], and manifold learning algorithms for uncovering the underlying structure of the data [34, 35].

Recently, there has been significant encouragement towards using pre-trained models on large-scale external datasets for anomaly detection. Deep feature extraction [36] involves leveraging pre-trained deep neural networks to extract high-level feature representations of the data. However, while these methods have shown promise, they come with their own set of limitations. For instance, CS-Flow [37] and FastFlow [38] employ normalizing flows as a representation technique for the task of anomaly detection. From multiple levels of the network, SPADE [39] uses feature maps to implement detailed anomaly detection and visualization of anomaly locations based on the k-Nearest Neighbors (k-NN) method. A notable limitation of SPADE is its relatively low sensitivity to fine-grained anomalies. The reliance on k-NN for anomaly scoring might lead to overlooking subtle changes in data patterns, as it treats all features equally without considering their discriminative power. As a result, it may fail to detect anomalies that are characterized by minute variations in local regions of the data. PaDiM [40] implements this method on a per-patch basis across multiple level feature maps. Some compositions within them bear similarities to those in PatchCore [23]. However, PaDiM struggles with emphasizing subtle local anomalies effectively. Due to the limited feature weighting mechanism, it tends to average out the importance of different features within a patch. Consequently, anomalies that manifest minor but critical deviations in specific features can be easily missed, which reduces the effectiveness in detecting fine defects in products or intricate data irregularities.

As one of the state-of-the-art anomaly detection methods today, PatchCore [23] uses a WideResNet50 [41], which has been pre-trained on ImageNet, as its feature extraction component. It uses average pooling to consolidate the feature maps derived from the intermediate layers of the WideResNet50, thereby computing features for each patch. During the training phase, features computed from normal sample data are stored in a memory bank. In the testing phase, the k-NN method is used to obtain features from the test data that have smaller distances to those in the memory bank within the feature space. This distance serves as the anomaly score, and the highest one constitutes the anomaly score of the image. PatchCore [23] reduces the loss associated with normal samples and anomalous information by considering adjacent pixels of patches and employs an iterative greedy approximation to build the core set, which lowers computational expenses. Although its strong performance on the MVTec AD benchmark, PatchCore is less sensitive to minute defects in products. Its averaging-based feature aggregation and reliance on global similarity metrics may smooth out local anomalies and fail to capture the fine details that are crucial for detecting small-scale anomalies.

We propose SE-PatchCore to address these issues by introducing attention mechanisms to the extracted features. Unlike SPADE, which treats all features equally, SE-PatchCore uses the attention mechanism to emphasize anomaly features, enabling it to capture fine-grained anomalies more effectively. By selectively highlighting the most discriminative features, it can identify subtle changes in data patterns that SPADE might overlook. Compared to PaDiM, SE-PatchCore provides a more refined feature weighting, ensuring that minor but critical anomalies within patches are not overshadowed by other less relevant features. For PatchCore's limitation in detecting small-scale defects, SE-PatchCore suppresses less useful information through the attention mechanism, enhancing its sensitivity to small-scale anomalies. This allows the

model to focus on the local details that are indicative of anomalies, making it more suitable for applications where detecting minute defects in products is essential.

2.3 SE Attention Module

The SE attention module enhances the network's representational power by adaptively adjusting the importance of channel features, and it consists of two primary components: the Squeeze stage and the Excitation stage. In the Squeeze stage, the module obtains a global feature descriptor for each channel by applying global average pooling across the spatial dimensions of the input feature maps. This global statistical feature is termed the Squeeze feature. The Squeeze feature is then processed through a fully connected layer to reduce the number of channels, followed by a non-linear transformation via a ReLU activation function. In the Excitation stage, the module passes the reduced features through another fully connected layer to restore the original dimensionality and outputs them, followed by a sigmoid normalization to learn the weights for each channel. These weights are used to scale each channel of the input feature maps, thereby adjusting the importance of channel-wise features. Finally, the adjusted channel features are multiplied pixel-wise with the input features, resulting in feature maps with adaptive weights that reflect the refined importance of each channel.

The SE attention module consists of three steps: Squeeze, Excitation and Scale. Firstly, in the Squeeze stage, global pooling is used to compress a feature map of size $H \times W \times C$ into a $1 \times 1 \times C$ feature map along the channel dimension. According to the calculation method described in Eq. (1), each two-dimensional feature map is transformed into a scala Z_c, which represents a global receptive field across the entire spatial extent of the input feature map.

$$Z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{1}$$

In Eq. (1), H and W are the height and the weight of the feature map, u_c is the feature value at position (i, j).The Squeeze operation transforms high-dimensional spatial features into a compact and channel-aware representation. This lays the foundation for the SE module to dynamically reweight channels and focus on discriminative features.

The function $F_{ex}(, W)$represents the Excitation step in the SE module, where W refers to the learnable parameters of the fully connected layers. This operation models the inter-channel relationships by processing the squeezed feature vector through two fully connected layers, ultimately producing a set of scaling coefficients that will be applied to each channel of the input feature map. The equation for the computation is as follows:

$$s = F_{ex}(z, W) = \sigma\left(W_2 \delta\left(W_1 z\right)\right) \tag{2}$$

In Eq. (2), W_1 and W_2 are the parameters learned of the two fully connected layers, where δ denotes the ReLu function, σ denotes the Sigmoid function, and z is the $1 \times 1 \times C$ feature vector obtained from the previous step.

The final stage is the scaling of weights, where s represents the importance of different channels. $F_{scale}(\cdot, \cdot)$ denotes the operation that scales the original feature maps using the weights generated from the Excitation step. Throug $F_{scale}(\cdot, \cdot)$, these weights are applied to the previously obtained feature maps by multiplying each feature map in u with its corresponding weight. This results in the final output of the SE module. The equation for this computation is as follows:

$$F_{scale}(s_c, u_c) = s_c u_c \tag{3}$$

In Eq. (3), s_c is the scaling coefficient for the C-th channel, and u_c is the feature map of the$C$-th channel. The structure of the SE module is shown in Fig. 1.

The SE attention module boosts the representational capabilities of a CNN without adding to the model's complexity. It allows the network to adaptively reweight the feature responses on a per-channel basis, thereby enhancing its comprehension of the input data. Furthermore, by permitting the model to concentrate on features that are particularly significant for certain tasks, the SE module can emphasize critical aspects of the target objects, leading to improved accuracy in the detection of anomalies of solar cells in EL images.

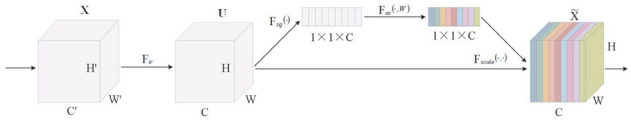


Figure 1 Structure of SE model

3 RESEARCH PROPOSED MODEL

Our proposed model builds upon PatchCore [23] and introduces the SE attention module, which is called SE-PatchCore. SE-PatchCore maintains the superior anomaly detection capability in anomaly detection, especially for localized anomalies, as seen in PatchCore. By introducing the SE attention module, SE-PatchCore has better anomaly detection performance in EL images of solar cells. In coreset subsampling, we introduce Fair k-Centers method, a k-center method that incorporates fairness constraints to ensure equitable representation in the selection of centers. Fig. 2 illustrates the constructure of the SE-PatchCore model.

3.1 Feature Extraction

We use X_N to denote the set of all nominal images ($x \in X_N, y_x = 0$), where $y \in \{0, 1\}$ indicates whether image x is a nominal image ($y = 0$) or an anomaly image($y = 1$). According to this rule, we use X_T to denote the test sample dataset $(\forall x \in X_T, y_x \in \{0, 1\})$. SE-PatchCore uses a WideResNet50 network pre-trained on ImageNet to extract features from the input images. We use $\varphi_{i,j} = \varphi_j(x_j)$ to denote the features extracted by the pre-trained network φ, where x_i represents an image from the dataset, and j indicates the layer level within the network architecture.

Each layer's final output within the convolutional network is utilized to generate feature maps for anomaly detection. Since shallow features of solar cells in EL images are more beneficial for effective defect detection, SE-PatchCore uses the feature maps from Layer1 and Layer2 of WideResNet50. We extract features by performing feature aggregation over local neighborhoods. It also extends the features from the pre-trained network φ to φ_i, $_f(H, W)$ to account for uneven patch sizes p.The extended feature vectors are expressed as follows:

$$N_p^{(H,W)} = \left\{ \begin{matrix} (a,b) \mid a \in \left[H - \lfloor p/2 \rfloor, ..., H + \lfloor p/2 \rfloor \right], \\ b \in \left[W - \lfloor p/2 \rfloor, ..., W + \lfloor p/2 \rfloor \right] \end{matrix} \right\} \quad (4)$$

Compared to Layer 2, Layer 1 contains richer local feature representations. Therefore, the feature aggregation method of PatchCore is used to Layer 1 for the detection of local anomalies. Let $\varphi_1(H, W, C)$ be the feature of Layer1, where H denotes the height, W denotes the channel width, and C denotes the number of channels. The patch-level features that aggregate local features in the neighborhood are expressed as follows:

$$P_1 = f_{\text{agg}}(\varphi_1) \quad (5)$$

In Eq. (5), f_{agg} denotes the aggregation function that operates on the neighborhood. In SE-PatchCore, average pooling is applied with a kernel size of 3, a stride of 1, and padding of 1.

Because Layer 2 offers a more global representation of features compared to Layer1, the features from Layer 2 are used as input to the SE attention module. Let the feature map of Layer2 be denoted as $\varphi_2(H, W, C)$ and let f_{SE} represent the process of passing the input features through the SE attention module. The features that account for relationships derived from Layer 2 can be represented as follows:

$$P_2 = f_{\text{SE}}(\varphi_2) \quad (6)$$

P_1 focuses on local features to detect local anomalies, while P_2 recalibrates global features through the attention mechanism, extracting features that are more useful for anomaly detection. The SE module recalibrates global features from Layer2 via channel attention, enhancing the extraction of discriminative features for anomaly regions. After extracting the output features from Layer 1 and Layer 2, bilinear interpolation is applied to the features from Layer 2 to match the output dimensions of Layer 1.

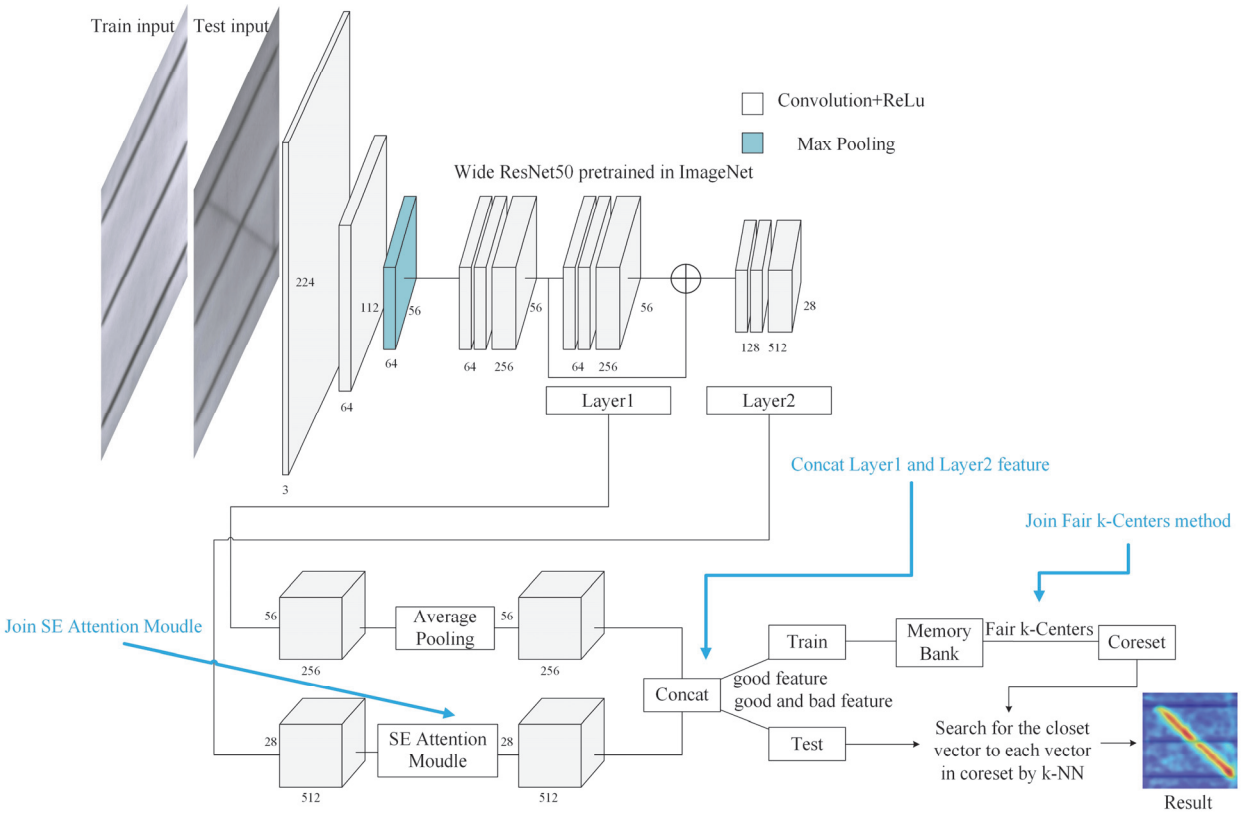


Figure 2 Anomaly detection process for solar cells in EL images

3.2 Core Set Subsampling

With an increase in the number of training input images, there is a corresponding growth in the memory bank's required size, which can result in substantially longer inference times. To reduce the time required for anomaly detection and improve detection efficiency, it is necessary to decrease the number of parameters in the memory bank while preserving the features stored within it as much as possible. Consequently, we use coreset subsampling to identify a subset $S \in A$, where the solution for sample S most closely approximates the solution for the entire sample set A. The corset M_c selected from the memory bank M is designed where by the representational

span of M_c mirrors that of the original memory bank M [25, 42]. We use the Fair k-Centers method [24], which satisfies fairness constraints, to replace the iterative greedy approximation method for coreset subsampling used in PatchCore [23]. The process for Fair k-Centers method is shown in Fig. 3.To satisfy the fairness constraints and find an optimal solution, this method leverages maximum matching techniques to adjust the initially selected center positions. This ensures that the final selection of k centers adheres to the predefined proportion requirements for each group and better minimizes the distance from the farthest point to its nearest center. The coreset subsampling procedure is in Tab. 1.

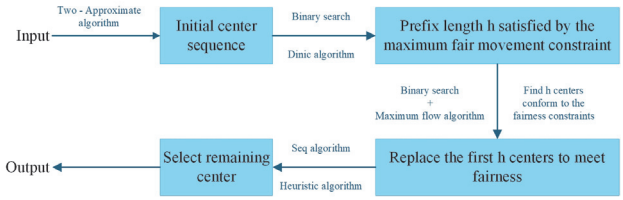


Figure 3 Process for Fair k-Centers method

3.3 Anomaly Detection

During the testing phase, when a test image is input, feature extraction is performed in the same manner as during the training phase. Using the coreset subsampled by training, we use m^{test} to denote the feature of test data and m^* to denote the nearest neighbor feature of m^{test}. The method then evaluates the anomaly score s for the test image X^{test} based on the distance between m^{test} and its nearest neighbor m^*. The anomaly score of the image s is determined by taking the highest anomaly score from all patches obtained from the test image. Through calculating the distance between test features and the nearest normal features in the memory bank, we take the maximum distance from all patches as the initial anomaly score to highlight local abnormal regions.

$$m^{\text{test}}, m^* = \underset{m^{\text{test}} \in P\left(X^{\text{test}}\right)}{\operatorname{argmax}} \underset{m \in M}{\operatorname{argmin}} \left\| m^{\text{test}} - m \right\|_2 \tag{7}$$

$$s^* = \left\| m^{\text{test}} - m^* \right\|_2 \tag{8}$$

Table 1 The coreset subsampling procedure

Method: Coreset subsampling
Input: Pretrained network φ, nominal data X_N Output: Patch-level memory bank M Method: for $x_i \in X_N$ $M = P_1 + P_2$ end Apply Fair k-Centers method $M_c \leftarrow \{\}$ For $i \in [0, \ldots, L-1]$ using Fair k-Centers method to find m_i m_i makes up M_c end $M \leftarrow M_c$

To derive the anomaly score s, adjustments are applied to the raw anomaly scores. In SE-PatchCore, this involves reweighting s^* based on the behavior of the nearest neighbor features to better distinguish anomalies. Specifically, the anomaly score is calculated using a formula that considers the distance between test features and their nearest normal features in the memory bank, as well as the distribution of surrounding neighbor features. This approach enhances the model's ability to discriminate anomalies effectively.

$$s = \left(1 - \frac{\exp \left\| m^{\text{test}} - m^* \right\|_2}{\sum_{m \in N_b\left(m^*\right)} \exp \left\| m^{\text{test}} - m^* \right\|_2} \right) \tag{9}$$

Where $N_b(m^*)$ denotes the b nearest patch-level features in the memory bank M to the test feature m^*. This reweighting of anomaly detection scores is more precise and robust compared to simply calculating the distance between features. Finally, the module's anomaly scores are recalculated according to their respective spatial positions. To match the resolution of the original input, the results are upscaled using bilinear interpolation. Additionally, the method applies a Gaussian kernel with width $\sigma = 4$ to smooth the results, optimizing the final anomaly segmentation map to be smoother and more natural, reducing false positives or false negatives that might arise from minor fluctuations during the anomaly detection process.

4 EXPERIMENTS

4.1 Solar Cells in Electroluminescence Image Anomaly Detection Dataset

We experiment using SE-PatchCore to detect anomalies in one public dataset for EL images of photovoltaic cells (ELPV) [43-45]. In the ELPV dataset, this paper selects monocrystalline solar cell images for anomaly detection and defines five types of defects: dark area, crack, finger failure on two types of solar cells (finger failure1 and finger failure2), and a mixed anomaly of crack and finger failure on one type of solar cell (mixture). For this experiment, a total of 382 solar cells in EL images are selected, with 180 normal images used for training and constructing the core subset, and 202 images, including both normal and defective samples, used for testing.

4.2 Experimental Condition

The experimental environment is based on the Windows 10 operating system, with programming conducted in the Python language. The CPU model is Intel Core i7-1065G7, and the system has 16GB of RAM. The batch size is set to 1, and the Coreset Subsampling sampling rates are 10% and 1%. SPADE [39], PaDiM [40], and PatchCore [23] are used as comparison methods.

4.3 Evaluation Metrics

For evaluating our test images, we use both AUROC and pixel-wise AUROC metrics. A positive finding is indicative of solar cells anomalies, while a negative result points to normal status. AUROC quantifies the model's anomaly detection effectiveness by mapping the true positive rate against the false positive rate and evaluating

the area under the curve. Meanwhile, the pixel-wise AUROC assesses anomaly segmentation accuracy by mapping the changes in precision relative to recall across the test data, again using the area under the curve as a measure.

The values for both AUROC and pixel-wise AUROC span from 0 to 1, with higher scores reflecting a greater capacity to differentiate. As these metrics get closer to 1, they demonstrate an increased proficiency of the model in discerning between anomalous and normal solar cells in EL images.

4.4 Detection Results

The results for AUROC and pixel-wise AUROC on ELPV dataset are shown in Tab. 2. For SE-PatchCore and PatchCore we list two levels of core set subsampling (10% and 1%). The results are averages of the five defect types.

Table 2 AUROC and pixel-wise AUROC on ELPV dataset

Method	SPADE	PaDiM	PatchCore-1%	PatchCore-10%	SE-PatchCore-1%	SE-PatchCore-10%
AUROC	86.4	98.3	98.3	98.5	98.6	98.7
pwAUROC	78.7	98.2	97.6	97.9	98.3	98.5

The experimental results in Tab. 2 demonstrate that SE-PatchCore achieves higher anomaly detection and segmentation performance across different scales of coreset subsampling. Even with a coreset subsampling rate of 1%, SE-PatchCore delivers high performance. Compared to SPADE, SE-PatchCore realizes a significant performance improvement in anomaly detection on solar cells in EL images. Notably, when compared to PaDiM, which also exhibits high detection performance, SE-PatchCore-10% reduces the image-level error from 1.7% to 1.4%, representing a 24% reduction in error. And SE-PatchCore-1% reduces the image-level error from 1.7% to 1.3%, representing an 18% reduction in error. Although the reduction in pixel-level error is modest, the score of SE-PatchCore-10% in pixel-level AUROC is up to 98.5% and the score of SE-PatchCore-1% in pixel-level is up to 98.3%, which still shows the excellent anomaly segmentation performance in SE-PatchCore.

The results for AUROC and pixel-wise AUROC on ELPV dataset are shown in Tab. 3 and Tab. 4. In this section, we detail the AUROC and pixel-wise AUROC for each type of solar cells in EL images, comparing SE-PatchCore and PatchCore with coreset subsampling rates of 10% and 1%. The aim is to investigate the impact of the SE attention module on image anomaly detection and segmentation performance under the same sampling ratio. Additionally, we conduct anomaly detection using SE-PatchCore without the Fair k-Centers method (denoted as SE-PatchCore*) to explore the effect of employing the Fair k-Centers method for coreset subsampling.

Table 3 AUROC on ELPV dataset (compared with PatchCore)

Types	Crack	Dark area	Finger failure1	Finger failure2	Mixture	Average
SE-PatchCore-10%	99.7	97.0	99.4	98.9	98.4	98.7
SE-PatchCore-1%	99.7	96.9	99.2	98.9	98.1	98.6
SE-PatchCore*-10%	99.2	96.7	99.6	98.7	98.6	98.5
SE-PatchCore*-1%	99.2	96.5	99.6	98.4	98.2	98.4
PatchCore-10%	99.0	99.1	98.4	98.1	98.1	98.5
PatchCore-1%	98.7	98.9	98.2	97.8	97.9	98.3

Table 4 Pixel-wise AUROC on ELPV dataset (compared with PatchCore)

Types	Crack	Dark area	Finger failure1	Finger failure2	Mixture	Average
SE-PatchCore-10%	97.8	97.9	99.7	98.9	97.9	98.5
SE-PatchCore-1%	97.4	97.8	99.7	98.7	97.8	98.3
SE-PatchCore*-10%	97.3	97.8	99.4	98.9	98.0	98.3
SE-PatchCore*-1%	97.2	97.5	99.4	98.9	97.8	98.2
PatchCore-10%	95.2	98.7	99.0	98.6	97.8	97.9
PatchCore-1%	94.6	98.6	98.6	98.4	97.7	97.6

The experimental results in Tab. 3 and Tab. 4 indicate that, under the same coreset subsampling ratio, SE-PatchCore generally outperforms PatchCore. Although the AUROC and pixel-wise AUROC for dark area decreased by 2.1% and 0.9% at 10% sampling, and by 2.0% and 0.8% at 1% sampling, there were improvements across the other four types of anomalies. Specifically, the average AUROC and pixel-wise AUROC at 10% coreset subsampling increased by 0.2% and 0.6%, respectively, while at 1% coreset subsampling, they increased by 0.3% and 0.7%, respectively. Notably, for crack detection, the pixel-wise AUROC of SE-PatchCore improved by 2.6% at 10% sampling and by 2.8% at 1% sampling compared to PatchCore. These results highlight that for fine defects such as cracks, SE-PatchCore exhibits superior anomaly segmentation performance. This underscores the advantage of the SE attention module in enhancing primary features while suppressing less important ones, thereby improving the detection of subtle anomalies.

Additionally, Tab. 3 and Tab. 4 compare the model using only the SE attention module without employing the Fair k-Centers method for coreset subsampling against SE-PatchCore. Using the Fair k-Centers method for coreset subsampling can provide an increase of 0.2% in AUROC and 0.1% in pixel-level AUROC for SE-PatchCore-1%. For SE-PatchCore-10%, it can provide a 0.2% increase in both AUROC and pixel-level AUROC. The results show that using the Fair k-Centers method yields better anomaly detection and segmentation performance compared to using the iterative greedy approximation method for coreset subsampling. This highlights the effectiveness and superiority of the Fair k-Centers method in constructing the coreset. These numerical improvements carry significant practical implications for quality control in solar cell

manufacturing. A 0.2% increase in AUROC indicates a more refined ability to distinguish between normal and anomalous solar cells, which can translate into a notable reduction in false negatives.

However, it is also important to note that the type of dark area defects have experienced performance decreases. One possible reason for this could be the limited representative data available for these specific defect types during the training phase. The other possible reason is that SE-Patchcore focuses on detecting small-scale defects and has certain limitations for defects of the dark area type.

Tab. 5 and Tab. 6 provide a comparison of anomaly localization results for the crack type and finger failure type anomalies, respectively, on SE-PatchCore, SE-PatchCore* (without the Fair k-Centers method), PatchCore and PaDiM, with coreset subsampling at 10% and 1%.

Table 5 Localization of anomaly areas onCrack

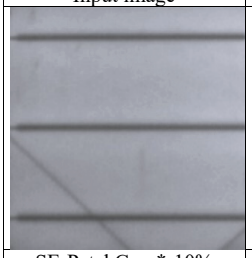
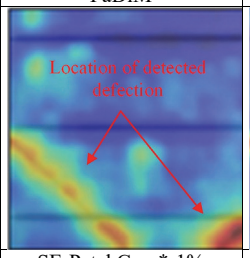
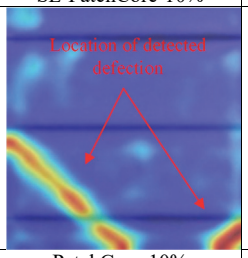
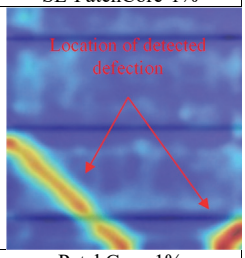
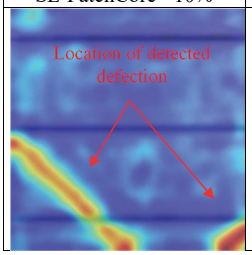
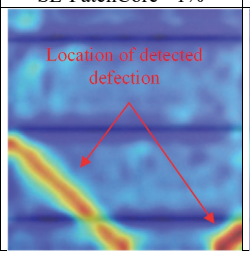
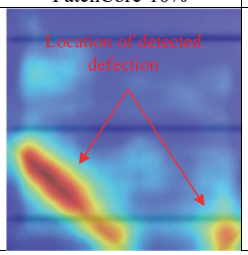
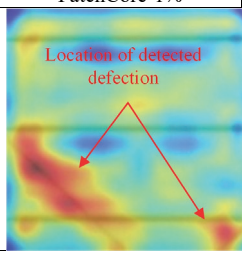
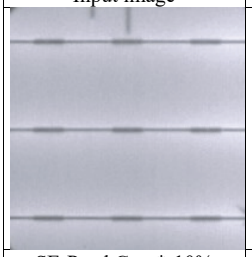
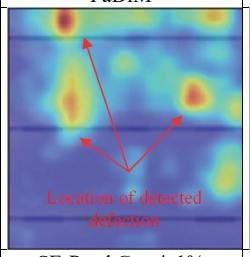
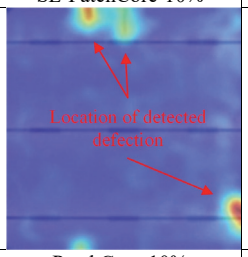
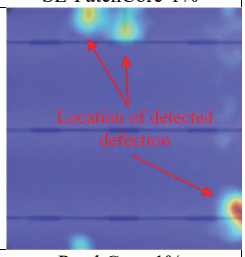
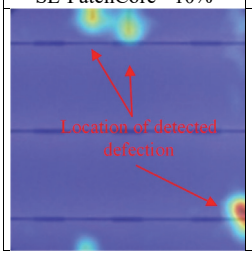
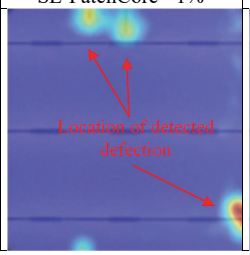
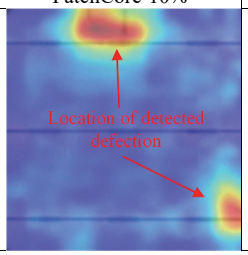
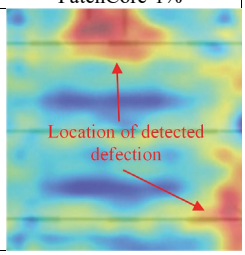
Input image	PaDiM	SE-PatchCore-10%	SE-PatchCore-1%
			
SE-PatchCore*-10%	SE-PatchCore*-1%	PatchCore-10%	PatchCore-1%
			

Table 6 Localization of anomaly areas on Finger failure

Input image	PaDiM	SE-PatchCore-10%	SE-PatchCore-1%
			
SE-PatchCore*-10%	SE-PatchCore*-1%	PatchCore-10%	PatchCore-1%
			

Tab. 5 and Tab. 6 demonstrate the effectiveness and superiority of SE-PatchCore in anomaly localization, showing more precise positioning of defect locations. The PaDiM and Patchcore divide some normal areas into defect areas, and the positioning of the defect areas is not precise. Moreover, for some small-scale defects in finger failure, there are some missed detections in PaDiM. When using coreset subsampling at merely 1%, SE-PatchCore can precisely localize anomaly positions. SE-PatchCore with 10% coreset subsampling not only achieves precise localization of anomalies but also provides superior prediction scores for defect locations compared to SE-PatchCore-1%. When the Fair k-Centers method is used for coreset subsampling, the anomaly localization of anomalous areas is notably more pronounced, and the prediction scores at the locations of anomalies are also higher according to the heatmap. This indicates that, compared to the iterative greedy approximation method, the coreset subsampled using the Fair k-Centers method more closely approximates the memory bank formed by the training set.

5 CONCLUSIONS

We proposed a SE-PatchCore method which extends the PatchCoreto detect anomalies in solar cells in ELimage by introducing SE attention module. The experimental results validated on the ELPV dataset show that

SE-PatchCore achieves performances of 98.7% in AUROC and 98.5% in pixel-wise AUROC, representing improvements of 0.2% and 0.6%, respectively, over PatchCore. Although its performance on cracks is slightly lower than that of PatchCore, SE-PatchCore demonstrates a clear advantage in the other four types of defect test sets. This highlights the effectiveness of the SE attention module in emphasizing key features while suppressing irrelevant ones. Moreover, by comparing SE-PatchCore using the Fair k-Centers method with the original iterative greedy approximation method for coreset subsampling, we found that employing the Fair k-Centers method to construct the core subset resulted in a 0.2% improvement in both AUROC and pixel-wise AUROC. This indicates that the Fair k-Centers method has ability to select a more representative subset of normal patterns while maintaining fairness across feature distributions, which contributes to enhancing anomaly detection and segmentation performance. Our work establishes a robust benchmark for solar cell defect detection in EL images, demonstrating the synergistic benefits of attention-guided feature extraction and optimized coreset selection.

Future research directions could explore how to improve the anomaly detection performance for "dark area" types, which can join targeted data augmentation, multi-scale feature fusion or hybrid approaches combining pixel-level and region-based analysis. Moreover, based on further optimizing the attention mechanism or using other attention modules, we will try to reduce detection time through methods such as simplifying the neural network architecture and decreasing computational complexity. This would aim to achieve precise and efficient detection of anomalies in solar cells in EL images.

Acknowledgments

This project is financially supported by Science and Technology Cooperation Special Project of Shijiazhuang (SJZZXC24001).

6 REFERENCES

[1] Zhu, J. & Pang, G. (2024). Toward generalist anomaly detection via in-context residual learning with few-shot sample prompts. *2024 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 17826-17836. https://doi.org/10.1109/CVPR52733.2024.01688

[2] Tsai, D.-M., Wu, S.-C., & Li, W.-C. (2012). Defect detection of solar cells in electroluminescence images using Fourier image reconstruction. *Solar Energy Materials and Solar Cells*, *99*, 250-262. https://doi.org/10.1016/j.solmat.2011.12.007

[3] Bin, L., Xianghao, H., & Shuai, F. (2011). Automatic inspection of Surface Crack in solar cell images. *2011 Chinese Control and Decision Conference (CCDC)*, 993-998. https://doi.org/10.1109/CCDC.2011.5968329

[4] Rakotoniaina, J. P., Breitenstein, O., Rifai, M., & Franke, D., & Schneider, A. (2004). Detection of cracks in silicon wafers and solar cells by lock-in ultrasound thermography. *Proceedings of PV Solar Conference*, 640-643.

[5] Belyaev, A., Polupan, O., Ostapenko, S., Hess, D., & Kalejs, J. P. (2006). Resonance ultrasonic vibration diagnostics of elastic stress in full-size silicon wafers. *Semiconductor Science and Technology*, *21*(3), 254-260. https://doi.org/10.1088/0268-1242/21/3/007

[6] Hilmersson, C., Hess, D. P., Dallas, W., & Ostapenko, S. (2008). Crack detection in single-crystalline silicon wafers using impact testing. *Applied Acoustics*, *69*(8), 755-760. https://doi.org/10.1016/j.apacoust.2007.03.002

[7] Chakrapani, S. K., Padiyar, M. J., & Balasubramaniam, K. (2011). Crack detection in full size CZ-silicon wafers using lamb wave air coupled ultrasonic testing (LAC-UT). *Journal of Nondestructive Evaluation*, *31*(1), 46-55. https://doi.org/10.1007/s10921-011-0119-3

[8] Bartler, A., Mauch, L., Yang, B., Reuter, M., & Stoicescu, L. (2018). Automated detection of solar cell defects with deep learning. *2018 26th European Signal Processing Conference (EUSIPCO)*, 2035-2039. https://doi.org/10.23919/EUSIPCO.2018.8553025

[9] Hussain, M., Chen, T., Titrenko, S., Su, P., & Mahmud, M. (2022). A gradient guided architecture coupled with filter fused representations for micro-crack detection in photovoltaic cell surfaces. *IEEE Access*, *10*, 58950-58964. https://doi.org/10.1109/ACCESS.2022.3178588

[10] Korovin, A., Vasilev, A., Egorov, F., Saykin, D., Terukov, E., Shakhray, I., Zhukov, L., & Budennyy, S. (2023). Anomaly detection in electroluminescence images of heterojunction solar cells. *Solar Energy*, *259*, 130-136. https://doi.org/10.1016/j.solener.2023.04.059

[11] Hassan, S. & Dhimish, M. (2023). Dual spin max pooling Convolutional Neural Network for solar cell crack detection. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-38177-8

[12] Balzategui, J., Eciolaza, L., & Maestro-Watson, D. (2021). Anomaly detection and automatic labeling for solar cell quality inspection based on generative Adversarial Network. *Sensors*, *21*(13), 4361. https://doi.org/10.3390/s21134361

[13] Labaca-Castro, R. (2023). Generative adversarial nets. *Machine Learning under Malware Attack,* 73-76. https://doi.org/10.1007/978-3-658-40442-0_9

[14] Zhang, M. & Yin, L. (2022). Solar cell surface defect detection based on improved Yolo V5. *IEEE Access*, *10*, 80804-80815. https://doi.org/10.1109/ACCESS.2022.3195901

[15] Aktouf, L., Shivanna, Y., & Dhimish, M. (2024a). High-precision defect detection in solar cells using yolov10 deep learning model. *Solar*, *4*(4), 639-659. https://doi.org/10.3390/solar4040030

[16] Zhu, N., Wang, J., Zhang, Y., Wang, H., & Han, T. (2024). An Adversarial Training Framework Based on Unsupervised Feature Reconstruction Constraints for Crystalline Silicon Solar Cells Anomaly Detection. *IEEE Transactions on Instrumentation and Measurement*, *73*, 1-13. https://doi.org/10.1109/TIM.2024.3462989

[17] Wang, H., Li, C., & Li, Y. F. (2024). Large-Scale Visual Language Model Boosted by Contrast Domain Adaptation for Intelligent Industrial Visual Monitoring. *IEEE Transactions on Industrial Informatics*, *20*(12), 14114-14123. https://doi.org/10.1109/TII.2024.3441638

[18] Larochelle, H. & Hinton, G. E. (2010). Learning to combine foveal glimpses with a third-order Boltzmann machine. *Advances in neural information processing systems*, *23*, 1243-1251.

[19] Su, B., Chen, H., Chen, P., Bian, G., Liu, K., & Liu, W. (2021). Deep learning-based solar-cell manufacturing defect detection with complementary attention network. *IEEE Transactions on Industrial Informatics*, *17*(6), 4084-4095. https://doi.org/10.1109/TII.2020.3008021

[20] Wang, C., Chen, H., & Zhao, S. (2024). RERN: Rich edge features refinement detection network for polycrystalline solar cell defect segmentation. *IEEE Transactions on Industrial Informatics*, *20*(2), 1408-1419. https://doi.org/10.1109/TII.2023.3275705

[21] Chen, H., Song, M., Zhang, Z., & Liu, K. (2022). Detection of surface defects in solar cells by bidirectional-path feature

pyramid group-wise attention detector. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1-9. https://doi.org/10.1109/TIM.2022.3218111

[22] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745

[23] Roth, K., Pemula, L., Zepeda, J., Scholkopf, B., Brox, T., & Gehler, P. (2022). Towards total recall in industrial anomaly detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14298-14308. https://doi.org/10.1109/CVPR52688.2022.01392

[24] Jones, M., Nguyen, H., & Nguyen, T. (2020). Fair k-centers via maximum matching. *International conference on machine learning*, *119*, 4940-4949.

[25] Sener, O. & Savarese, S. (2017). Active learning for convolutional neural networks: A core-set approach. *arxiv preprint arxiv:1708.00489*.

[26] Hinton, G. E. & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with Neural Networks. *Science*, *313*(5786), 504-507. https://doi.org/10.1126/science.1127647

[27] Chen, Z., Yeo, C. K., Lee, B. S., & Lau, C. T. (2018). Autoencoder-based network anomaly detection. *2018 Wireless Telecommunications Symposium (WTS)*, 1-5. https://doi.org/10.1109/WTS.2018.8363930

[28] Zhou, C. & Paffenroth, R. C. (2017). Anomaly detection with robust deep autoencoders. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 665-674. https://doi.org/10.1145/3097983.3098052

[29] Tang, T. W., Kuo, W. H., Lan, J. H., Ding, C. F., Hsu, H., & Young, H. T. (2020). Anomaly detection neural network with dual auto-encoders gan and its industrial inspection applications. *Sensors*, *20*(12), 3336. https://doi.org/10.3390/s20123336

[30] Song, J., Kong, K., Park, Y. I., Kim, S. G., & Kang, S. J. (2021). AnoSeg: Anomaly segmentation network using self-supervised learning. *arxiv preprint arxiv:2110.03396*.

[31] Abati, D., Porrello, A., Calderara, S., & Cucchiara, R. (2019). Latent space autoregression for novelty detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 481-490. https://doi.org/10.1109/CVPR.2019.00057

[32] Zaigham Zaheer, M., Lee, J.-H., Astrid, M., & Lee, S.-I. (2020). Old is gold: Redefining the adversarially learned one-class classifier training paradigm. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14171-14181. https://doi.org/10.1109/CVPR42600.2020.01419

[33] Zhou, X., Xiong, J., Zhang, X., Liu, X., & Wei, J. (2021). A radio anomaly detection algorithm based on modified generative Adversarial Network. *IEEE Wireless Communications Letters*, 10(7), 1552-1556. https://doi.org/10.1109/LWC.2021.3074135

[34] Olson, C. C., Judd, K. P., & Nichols, J. M. (2018). Manifold learning techniques for unsupervised anomaly detection. *Expert Systems with Applications*, *91*, 374-385. https://doi.org/10.1016/j.eswa.2017.08.005

[35] Yousefpour, A., Shishehbor, M., Zanjani Foumani, Z., & Bostanabad, R. (2024). Unsupervised anomaly detection via nonlinear manifold learning. *Journal of Computing and Information Science in Engineering*, *24*(11). https://doi.org/10.1115/1.4063642

[36] Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on Convolutional Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(10), 6232-6251. https://doi.org/10.1109/TGRS.2016.2584107

[37] Rudolph, M., Wehrbein, T., Rosenhahn, B., & Wandt, B. (2022). Fully convolutional cross-scale-flows for image-based defect detection. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1829-1838. https://doi.org/10.1109/WACV51458.2022.00189

[38] Yu, J., Zheng, Y., Wang, X., Li, W., Wu, Y., Zhao, R., & Wu, L. (2021). Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arxiv preprint arxiv:2111.07677*.

[39] Cohen, N. & Hoshen, Y. (2020). Sub-image anomaly detection with deep pyramid correspondences. *arxiv preprint arxiv:2005.02357*.

[40] Defard, T., Setkov, A., Loesch, A., & Audigier, R. (2021). Padim: A Patch Distribution Modeling Framework for ANOMALY DETECTION and localization. *International conference on pattern recognition*, 475-489. https://doi.org/10.1007/978-3-030-68799-1_35

[41] Zagoruyko, S. & Komodakis, N. (2016). Wide residual networks. *Procedings of the British Machine Vision Conference 2016*. https://doi.org/10.5244/C.30.87

[42] Sinha, S., Zhang, H., Goyal, A., Bengio, Y., Larochelle, H., & Odena, A. (2020). Small-gan: Speeding up gan training using core-sets. *International Conference on Machine Learning*, 9005-9015.

[43] Buerhop, C., Deitsch, S., Maier, A., Gallwitz, F., Berger, S., Doll, B., Hauch, J., Camus, C., & Brabec, C. (2018). A Benchmark for Visual Identification of Defective Solar Cells in Electroluminescence Imagery. *35th European PV Solar Energy Conference and Exhibition*, 1287-1289.

[44] Deitsch, S., Buerhop-Lutz, C., Sovetkin, E., Steland, A., Maier, A., Gallwitz, F., & Riess, C. (2021). Segmentation of photovoltaic module cells in uncalibrated electroluminescence images. *Machine Vision and Applications*, *32*(4). https://doi.org/10.1007/s00138-021-01191-9

[45] Deitsch, S., Christlein, V., Berger, S., Buerhop-Lutz, C., Maier, A., Gallwitz, F., & Riess, C. (2019). Automatic classification of Defective Photovoltaic module cells in electroluminescence images. *Solar Energy*, *185*, 455-468. https://doi.org/10.1016/j.solener.2019.02.067

Contact information:

Tianyi REN
1) School of Electronics and Information Engineering,
Hebei University of Technology, Tianjin 300401, China
2) Innovation and Research Institute of Hebei University of Technology in Shijiazhuang, Shijiazhuang 050299, Hebei, China
E-mail: 202431903003@stu.hebut.edu.cn

Yatong ZHOU
(Corresponding author)
1) School of Electronics and Information Engineering,
Hebei University of Technology, Tianjin 300401, China
2) Innovation and Research Institute of Hebei University of Technology in Shijiazhuang, Shijiazhuang 050299, Hebei, China
E-mail: zyt@hebut.edu.cn