IET Renewable Power Generation

Research Article

Photovoltaic cell defect classification using convolutional neural network and support vector machine

ISSN 1752-1416
Received on 26th November 2019
Revised 29th June 2020
Accepted on 23rd July 2020
E-First on 2nd October 2020
doi: 10.1049/iet-rpg.2019.1342
www.ietdl.org

Ashfaq Ahmad[1,2], Yi Jin[1] ✉, Changan Zhu[1], Iqra Javed[3], Asim Maqsood[2], Muhammad Waqar Akram[1]

[1]Department of Precision Machinery & Instrumentation, University of Science and Technology of China, Hefei, People's Republic of China
[2]Department of Electronics & Electrical Systems, The University of Lahore, Lahore, Pakistan
[3]Department of Informatics and Systems, University of Management and Technology, Lahore, Pakistan
✉ *E-mail: jinyi08@ustc.edu.cn*

Abstract: Automatic defect classification in photovoltaic (PV) modules is gaining significant attention due to the limited application of manual/visual inspection. However, the automatic classification of defects in crystalline silicon solar cells is a challenging task due to the inhomogeneous intensity of cell cracks and complex background. The present study is carried out for automatic defects classification of PV cells in electroluminescence images. Two machine learning approaches, features extraction-based support vector machine (SVM) and convolutional neural network (CNN) are used for the solar cell defect classifications. Suitable hyperparameters, algorithm optimisers, and loss functions are used to achieve the best performance. Solar cell defects are divided into seven classes such as one non-defective and six defective classes. Feature extraction algorithms such as histograms of oriented gradients (HOG), KAZE, Scale-Invariant Feature Transform (SIFT) and speeded-up-robust features (SURF) are used to train the SVM classifier. Finally, the performance results are compared. It is concluded that CNN's accuracy for solar cell defect classification is 91.58% which outperforms the state-of-the-art methods. With features extraction-based SVM, accuracies of 69.95, 71.04, 68.90, and 72.74% are obtained for HOG, KAZE, SIFT, and SURF, respectively. The present study may contribute to making a PV system more efficient for classifying defects to improve the power system efficiency.

1 Introduction

Photovoltaic (PV) is one of the most important sources of renewable energy. It has undergone a great advancement in the last few years. PV modules are normally protected by an aluminium body and laminated-glass [1]. However, these protection schemes cannot constantly avoid thermal and mechanical damages during the manufacturing process, transportation, and handling the PV module during installation [2]. Even after installation, the modules are exposed to natural hazards such as raining, lightning, snow falling, and wind during operation in the field. It leads to the appearance of cracks and other cell defects [3]. Such defects affect the module quality, increase power loss, decrease conversion efficiency, and also reduce the overall system performance [3]. The proper classification and assessment of defects can help to increase the PV system performance, quality, and reliability [4].

The defect classification in PV cells has a key role in controlling the quality and output power of PV cells. The fast and accurate determination of the defect locations in PV module and cell is very important [5]. The PV defects can be classified using infrared (IR) imaging [6], electroluminescence (EL), large-area laser beam induced current, current–voltage characteristics [6] etc. Recent advancements in EL imaging have made it possible to get the defects information hidden within the PV cell [7]. The EL imaging technique is a powerful and quick characterisation technique that provides information regarding the defects in PV cells [8, 9].

The visual inspection of the PV module has a significant demand. However, there are some difficulties in the use of visual inspection, the principal ones are time-consuming, costly, and required high salaries of trained personnel. Moreover, the visual classification of solar cell defects may involve human error. In this context, an automatic inspection system has become increasingly important to meet the growing demand for high-quality PV power generation.

To solve the challenging issue of automatic defect classification in PV cells, a few researchers have conducted studies. In [10], the authors conducted a study based on the Fourier image reconstruction to identify the defects inside the EL images. However, the limitation of this study is to detect the type of crack. Tseng *et al*. [11] considered a monocrystalline solar cell for automatic detection of finger interruptions. However, the method does not have sufficient indications for future power loss predictions. The authors of [12] presented an automatic defect detection approach to classifying the saw-mark defects in multi-crystalline solar wafer images. In their study, they utilised CNN model for classification. Chen *et al.* [13] proposed a multi-spectral deep convolutional neural network (CNN) for visual solar cell defect detection. In [14], Deitsch *et al.* presented a study based on CNN and support vector machine (SVM) for the identification of defects using 2624 samples of EL images and achieved an accuracy of 88.24%. However, the model accuracy still needs to be improved. Chiou *et al.* [15] developed a model for extracting crack defects in solar cell images using a regional growth detection algorithm. The authors of [16] used the machine vision approach for solar cells cracks detection. However, this approach can only detect the edge defect of the solar cell. Anwar and Abdullah [17] proposed an algorithm for micro-cracks identification in polycrystalline PV cells using improved anisotropic diffusion filter and image segmentation. However, this technique is only applicable to micro-crack cell detection and not applicable to multiple defects classification. In [18], the authors applied deep neural networks for cracks and missing corners detection in solar cells. However, the dataset used in this method is small. In another research [19], the author employs a deep belief network for defect detection in PV cells.

In [20], the authors developed a model for PV cell crack detection using a pattern recognition approach and SVM is trained with local descriptors extracted from images data. However, the authors utilised photoluminescence (PL) and IR images in their research. The authors [21] presented a study using the Haar-like feature extraction algorithm for unsupervised defect detection. However, the techniques are only useful on a small dataset. Therefore, defect classification methods for the larger dataset are

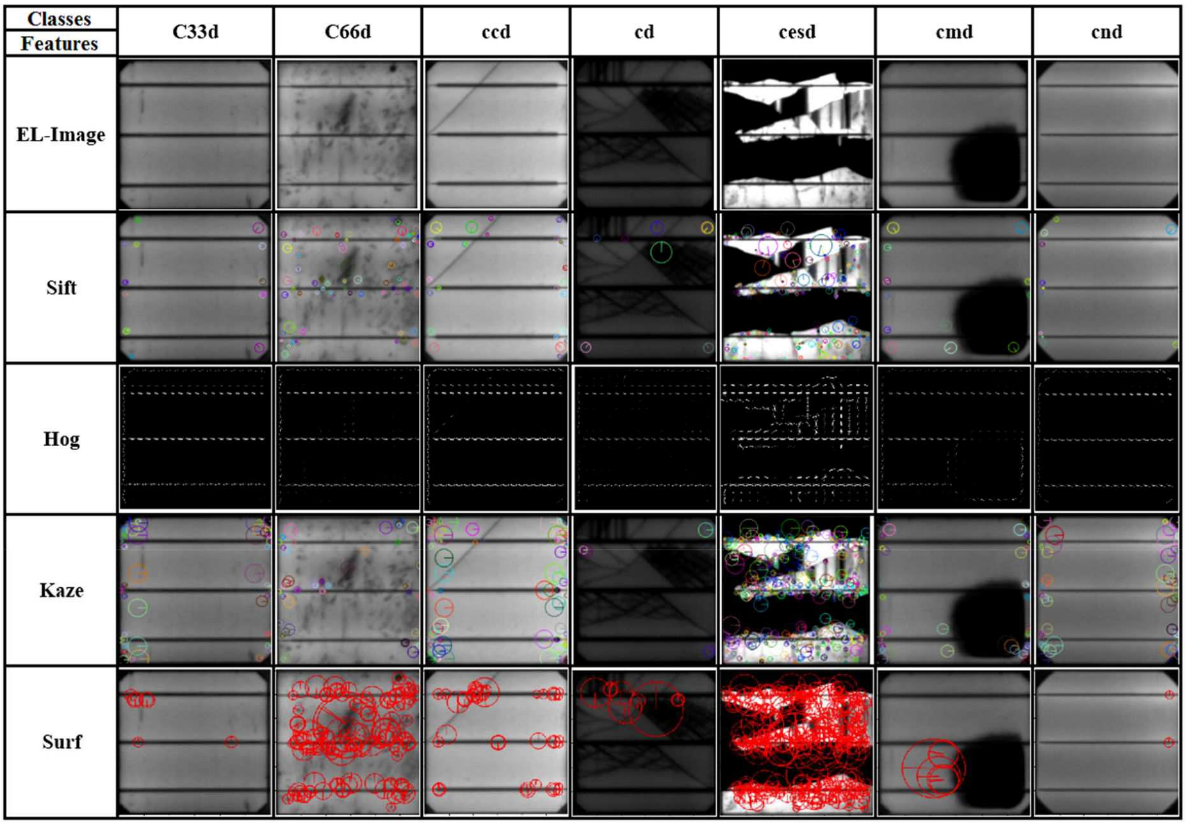

Fig. 1 *Key points detectors and descriptors for all seven classes*

required. In another research [22], the author described that in CNN architectures, convolutional layers are specifically designed for image data. Convolutional layers utilise kernel to filter the image for local features extraction in initial layers and normally followed by a pooling layer to reduce the number of extracted features. Extracted features are collected in final convolution layers to detect higher-level features.

The above studies wholly affirm the excellent performance of deep learning and feature extraction approaches in defect classification. However, the drawbacks of existing studies are that they classify only a few defects. The performance in solar cells' automatic defect classification needs to be further improved. We presume that it is possible to achieve efficient automatic defect classification in PV cells using CNN architecture and other feature extraction techniques such as histograms of oriented gradients (HOG), KAZE, SIFT, and speeded-up-robust features (SURF).

In line with the above motivation, the objectives of the present study are to classify defects in EL images of PV cells using CNN and SVM (radial basis kernel). In this study, the classification of defective cell images is important, especially when distinguishing between normal and defective cells. Therefore, the EL images dataset is divided into seven classes. The proposed classes are one normal class named as a non-defective cell (cnd) and six defective classes (33% defective cell (c33d), 66% defective cell (c66d), crack defective cell (ccd), defective cells (cd), electrically separated defective cells (cesd), and material defective cell (cmd). For SVM-based classification, features are extracted using HOG, KAZE, SIFT, and SURF and fed to the classifier. Therein, the principal component analysis (PCA) algorithm is utilised for dimensionality reduction. Finally, the defect classification results of CNN- and SVM-based approaches are presented in terms of classification accuracy. The perfect defect classification of solar cells can help to enhance the PV system performance, quality, and reliability.

The paper is structured as follows: the basic theory of solar module defects and machine learning approaches is presented in Section 2, methodology of the research is illustrated in Section 3, results and discussion are represented in Section 4, and finally, conclusions are drawn at the end.

2 Solar cell defects and identification using EL imaging

This section introduces a few solar cell defects and different images based techniques that can be used to identify the defects. This section will also discuss Machine learning approaches.

2.1 Solar cell defects

In a very large-scale power production of solar cells, the uncertainty in the output power at the generating station of the PV system due to the defect is a serious issue. To achieve reliable performance and high efficiency of the PV production system, there is a strong need for timely and accurate assessments of PV cells. To this end, it is most important to detect and classify defects.

The performance of any type of PV cell is affected by its defects [23]. In general, solar cell defects are classified into two major types such as intrinsic and extrinsic. A few cell defects are considered in this research for classification as shown in Fig. 1 (see the first row). hese are non-defective cells, defective cells, 33% defective cells, 66% defective cells, crack defective cells, electrically separated defective cells, and material defective cells. Imaging-based PV cell defects can be identified using IR imaging, PL, EL etc. [24, 25]. In this research, we used EL images dataset for classification purposes.

2.2 Identification of defects using EL imaging

EL imaging is non-destructive technology that is utilised for defect detection in PV cells. Suitable hardware configuration is required for capturing the EL images. Generally, the EL imaging [26] system hardware arrangement includes direct current (DC) power supply, solar cells, probe wires, and charge-coupled device (CCD) camera; and the imaging is carried out in a dark room as shown in Fig. 2.

In EL imaging, the current is applied to PV modules by using a DC power supply, which can provide enough voltage to achieve short circuit current. Meanwhile, the solar cell is excited and IR signals are emitted that can be imaged by a CCD camera in a dark environment [27]. The high-resolution EL imaging enables

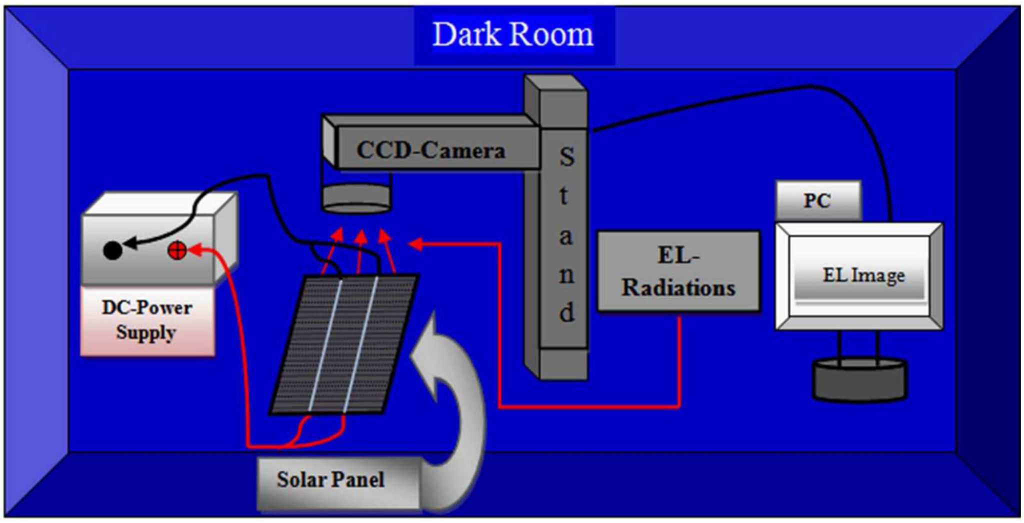

Fig. 2 *EL imaging setup*

detection of the finger interruptions, microcracks, line cracks etc. The visual inspection of EL images is not only time consuming and expensive but also needs trained experts. In the present research, these limitations are overcome by proposing an automatic defect classification scheme for EL images of PV cells.

The EL images are greyscale images. The regions of low EL intensity can be seen from these images that may indicate defects. The EL intensity is proportional to the excess number of minority carriers ($n_{P(o)}$) at the area of p–n junction, which is controlled by the forward voltage $V_{\rm f}$ (equal to the difference of quasi-Fermi rate and minority carrier) in accordance with (1)

$$n_{P(o)} = \ n_P \exp(qV_{\rm f}/KT) \tag{1}$$

where n_P is the density of minority equilibrium carrier at the layer-p, q is the charge of the electron, k is a constant of Boltzmann, and T is the temperature [28].

Thus, the EL dependence on forward voltage could be expressed, when the forward current is varied, and then the EL intensity $I_{\rm L}$ is demonstrated as a function of $V_{\rm f}$ can be analysed by (2)

$$I_{\rm L} = A\exp(qV_{\rm f}/KT) \tag{2}$$

or

$$\ln I_{\rm L} = A' + (q/KT)V_{\rm f} \tag{3}$$

where A and A' are constants [28].

3 Support vector machine

This section introduces SVM classification for automatic defect identification in solar cells.

In the present research, we propose SVM to address the PV cells defect classification. It is an intelligent supervised machine learning algorithm. In SVM, a high-dimensional feature space is used to map the input vectors non-linearly into a very concise form [29]. Therein, feature space optimal hyperplane is resolved to maximise the simplification capability of the classifier. The main characteristics of SVM are the possibility to determine an error acceptable for validation; the dimension of the space does not depend on the upper limit of generalisation errors, and the error limit is reduced by increasing the margin γ. A SVM is a very powerful, versatile machine learning algorithm; based on the principle of SVM, it can perform linear or non-linear classification, regression problems. It is one of the most popular algorithms in machine learning; SVMs are particularly well suited for classification of complex but small- or medium-sized datasets [29].

3.1 SVM for defect classification

The training set in support vector classification is $= \{X_{1,}X_{2,}X_{3,}\dots\dots\dots, X_n\}, \ \ X_i \in R^M$, where, M is the feature of each training sample (X_i) that defines a specific identification and corresponds to each of the two categories $Y = \{Y_1, Y_2, Y_3, \dots, Y_n\}, \ \ Y_i \in \{+\,1, \ -1\}$. A vector quantity $\boldsymbol{W} \in R^N$ and a scalar quantity $b \in R$ exists, when data is linearly divisible in such a way that $Y_i(\ \boldsymbol{W} \cdot X_i + b)$ is greater than or equal to one for all the models in the training information set $(\ i = 1, \ 2, \ 3, 4, \dots, l)$. As a result, canonical hyperplane $\boldsymbol{W} \cdot X + b$ is equal to +1 for nearest points on the one side, and $\boldsymbol{W} \cdot X + b$ is equal to $-\ 1$ for nearest points on the other side [30]. The optimal hyperplane segregates the points lying on different categories, yielding the highest margin of separation. A segregating hyperplane can be derived as

$$\text{Minimise } \frac{1}{2} \parallel W^2 \parallel \ + \ C\left(\sum_{i=1}^{l} \varepsilon_i\right) \text{subject to } y_i(\boldsymbol{W} \cdot X_i + b) \geq \ 1 - \\ (\varepsilon_i, \ \varepsilon_i) \geq \ 0 \ \forall \tag{4}$$

A Lagrangian is constructed to solve the optimisation problem as

$$\lambda(w, b, \alpha) = \ \frac{1}{2} \parallel w^2 \parallel \ - \sum_{i=1}^{l} \alpha_i(y_i(w \cdot x_i \ \ b) - 1) \tag{5}$$

The fundamental variables w and b are initially minimal, and they are increased with the double variable α_i. The conditions of Karush–Kuhn–Tucker direct to obtain the resultant vector with respect to the training model, $w = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x}_i$ for a few values of $\alpha_i \geq 0$. A subset of the training model α_i is not equal to zero and is positioned on the boundary; these are called the support vectors. In such a case, with SVM the input vector, $\boldsymbol{x}_{\rm i}$ can map to feature space of higher dimension where a linear decision margin is unsuitable. A function of kernel $k(. , .)$ can be represented under this condition that $k\left(x_i, x_j\right) = \ x_i \,.\, x_j$. To build input space SVM using convolution of scalar product, the non-linear decision function is as follows:

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i k(\boldsymbol{x}, \ \boldsymbol{x}_i) \ + \ b\right) \tag{6}$$

where $\boldsymbol{x}$ is the test vector. The primal constraint is used to find b and is calculated by expression $\alpha(y_i(w \cdot \boldsymbol{x}_i + b) - \ 1) = 0$, where

$i = 1,\ 2,\ 3, \ldots, l$ such that $\alpha_i \neq 0$ and signal function is represented by sgn.

4 Convolutional neural networks

This section introduces CNN classification for automatic defect identification in solar cells.

In the present research, we propose CNN architecture to address the PV cells defect classification. The CNNs are a particular category of artificial neural networks. CNN can automatically learn a feature hierarchy from the input image matrices, which is better than other complex feature extraction algorithms [31]. CNNs have the most influential innovations in the field of computer vision and are generally used in various applications such as image classification, image segmentation, object tracking, and so on [32]. CNN is developed by researching the function of the visual cortex in the brain. In fact, several neurons have a small local receptive field in the visual cortex. Simply neural network is a directed structure connecting an input layer to an output layer. All operations are usually differentiable and the overall vector function can easily be written as

$$\bar{Y} = f(\bar{X})$$

where $\bar{X} = \ (X_1,\ X_2,\ X_3, \ldots,\ X_n)$ and $\bar{Y} = \ (Y_1,\ Y_2,\ Y_3, \ldots,\ Y_m)$.

4.1 CNN architecture for defect classification

A standard CNN structural design consists of several nested layers of convolutional and pooling followed at the end by fully connected (FC) layers. A simplified arrangement of CNN can be described as input, convolutional layer, pooling layer, and FC. Input: when a computer machine sees an image, it will see an array of pixel values. Generally, CNN input is one-grey channel image or three-colour channel image matrix that contains the intensity values at each position.

4.1.1 Convolutional layer: It is the most important part of a CNN configuration. The convolutional layer takes ‘I’ feature maps as the input and has ‘k’ feature maps as output. It is the primary layer that extracts features from an input image. Convolutional layer applies a set of filters or neurons that are connected to specific regions (referred to as a receptive field) in the last layer (or input image). These filters have learnable parameters.

In convolution operation, the parameter sharing strategy is applied in which one filter is convolved throughout the spatial dimensions over the entire input image to extract one feature. The initial layers extract local or low-level features of images such as straight edges, simple colours, and curves. The following (later) layers extract higher-level image features such as shape, design etc. The mathematical convolutional procedure can be expressed in (7) [33]

$$y_{i^{l+1},j^{l+1},d} = b_d + \sum_{i=0}^{H} \sum_{j=0}^{W} \sum_{d^l=0}^{D^l} f_{i,j,d^l,d} \times x^l_{i^{l+1}+i,j^{l+1}+j,d^l} \tag{7}$$

where $y_{i^{l+1},j^{l+1},d}$ is the output feature map of the lth layers, H is the height of the spatial location, w is the width of the spatial location, D^l is the preceding layer feature map, $f_{i,j,d^l,d}$ is a convolution kernels of the lth layers at the ith row and the jth column in D^l feature map, $x^l_{i^{l+1}+i,j^{l+1}+j,d^l}$ is the output of the preceding layer neuron at the i^{l+1}th row and the i, j^{l+1}th column in j, d^l feature map and b_d is the bias term usually added to $y_{i^{l+1},j^{l+1},d}$.

The updating of mathematical weights can be expressed by

$$W = W_i - \eta \frac{\mathrm{d}L}{\mathrm{d}W} \tag{8}$$

where L is the loss, W is the weight, W_i is the initial weight, and η is the learning rate.

4.1.2 Pooling layers: This layer is generally used to minimise the spatial size of the output and is used immediately after the convolution layer. The purpose of this layer is to reduce the computation cost and network parameters. Less number of parameters avoid overfitting. Commonly, the pooling layer is placed between two consecutive convolutional layers. Max pooling is the most common form of pooling.

4.1.3 FC layer: FC layer is the last CNN architecture layer. FC layer is attached to the end of the network to identify high-level features obtained from convolutional layers. Finally, Softmax classifier is used to classify the outputs of the network as per the proposed classes.

5 Methodology

This section describes the proposed intelligent algorithms for solar cell defect classification. The model developed in this research is based on the CNN architecture model and SVM model (with multi-feature extraction techniques). The developed models are based on solar cells dataset of EL images with a resolution of 300×300 pixels. For better defects classification, data augmentation techniques are applied to enlarge the training dataset. Furthermore, evaluation criteria and flowchart of CNN and SVM optimisation models are discussed in this section.

5.1 Implementation of CNNs

This sub-section introduces the proposed framework and selection process of CNNs.

5.1.1 Data augmentation: A large dataset is important for the performance of a deep learning model to avoid overfitting. However, the performance of the model can be enhanced by increasing the data size. The data augmentation techniques of cropping, rotating, noise addition, and flipping are generally used [34].

In our study, we have limited training data, so our dataset needs to be expanded. To enhance the training dataset, we applied the offline data augmentation. In solar cell defect classification, it is not important whether the position and direction of the defect are at a right angle or not; therefore, data augmentation techniques can be successfully applied for EL image's dataset.

We select data augmentation techniques such as rotating, flipping, and cropping in current research. Furthermore, few of the other data augmentation techniques suggested by the authors [35] have been found to be successful; and based on that, we also select contrast and Gaussian blur to increase the training data size. The selected rotation ranges are $\pm 3°$, $90°$, $180°$, and $270°$. Additionally, random flipping along the x-axis and y-axis is adopted. The cropping method is applied in such a way that important information is preserved. To improve the generalisation capability, we randomly blur the images through a Gaussian filter with radius three. Similarly, we also used contrast to clearly show the location of various defects. To check the difference from the original images, the new images obtained are carefully checked. After implementing these augmentation operations, the accuracy is improved.

5.1.2 Batch size: It is one of the CNN model hyperparameters. That specifies the number of samples to be processed before updating the internal model parameters. Different batch size values such as 32, 64, 96 etc. are used in the CNN model to increase the training speed or to get higher accuracy. Therefore, in present research, the suitable batch size value is 64.

5.1.3 Epoch: This is another model hyperparameter. It determines the number of times that the learning algorithm operates throughout the training dataset. Therefore, in the present research, 150 epochs are selected for the model training.

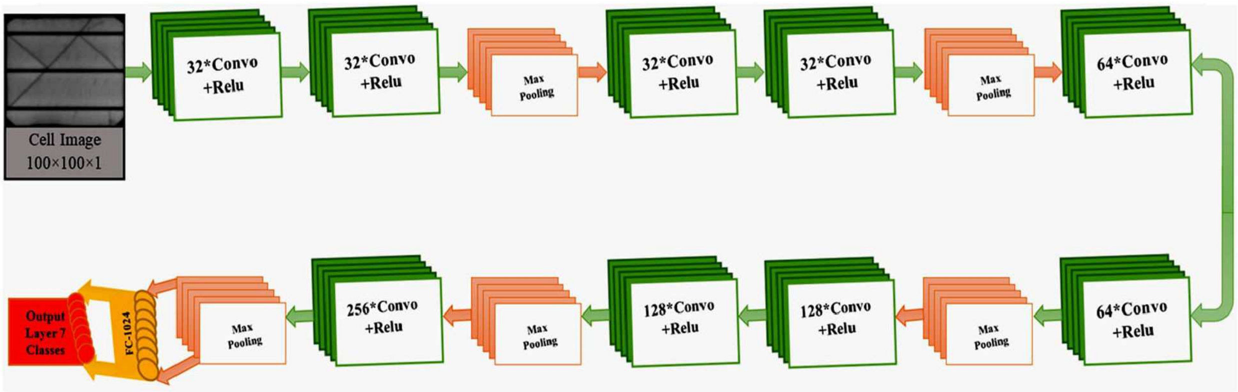

Fig. 3 *Proposed CNN architecture*

5.1.4 Dropout: It is a regularisation technique, which prevents overfitting by dropping out units both hidden and visible in a CNN model. In this research, the selected dropout value is 0.5 for the best performance results.

5.1.5 Softmax: It is an activation function in CNN and normally used in the last layer of the CNN model instead of other activation functions such as rectified linear unit (ReLU), tanh, and sigmoid. The reason is that it converts the output of the last layer of the neural network into what is basically a distribution of probability.

5.1.6 CNN architecture: The step-by-step process of selecting CNN architecture is discussed in this sub-section. CNN has been used in the present work to extract features and classify cell defects. The input images are greyscale EL images with a size of 100 × 100. In Section 5, it is mentioned that the original size of the dataset is 300 × 300. However, the input image size is before passing to CNN layer is 100 × 100. Mean that we have down sampled the input image size into 100 × 100. The reason is that it reduces the computational expense or the feature set size for the training of neural network. Therefore, it would reduce the time required to train the network. Based on experimentation; we select nine-layered (convolutional layers) network architecture. The details of layers are given below.

The selection of layers, kernels, and other related details are as follows: the first four convolution layers have 32 kernels/neurons, the fifth and sixth convolution layers use 64 kernels, the seventh and eighth convolutional layers contain 128 kernels. Finally, 256 kernels are used in the ninth convolution layer. All the kernels in these nine layers are of size 3 ×3 with a stride of 1. The architecture is illustrated in Fig. 3.

Additionally, there are five sub-sampling or pooling layers. The pooling is carried out by max-pooling layers that are used to reduce the image size and it also helps to mitigate the risk of over fitting. Moreover, max-pooling itself does not learn, it simply takes the $K \times K$ region as an input and gives a single maximum value of that region as an output. In this process, pooling use max function to reduce $N \times N$ size input into $(N/K) \times (N/K)$ output because max function reduces each $K \times K$ block to a single value. In the above selected CNN architecture, there are five blocks and the pooling layer is used in each block. The selection of filter size for pooling is 2 ×2 with a stride of 2.

5.1.7 Batch normalisation: Batch normalisation is a technique to improve the neural network's speed, efficiency, and stability. By reducing internal covariate shift, it can accelerate deep network training. Each layer in the network adapts to the constant changes in its input distribution, a phenomenon known as the internal covariate shift. Batch normalisation is achieved through a normalisation step, which fixes each input layer's means and variances. A network is known to train faster with input data have zero mean and unit variance and normalising each input feature separately, rather than jointly [36]. The normalisation of each input layer is complicated. Zeroing the mean of the input layer will ignore the learned bias of the previous layer, and the problem with the unit variance becomes even worse. The parameters of the previous layer's may increase infinitely when the inputs layers are normalised [36].

Therefore, a batch normalisation layer re-learns the scale and bias of the input layer after normalisation. The batch normalisation relationship is expressed as

$$y^K = \gamma^K \text{normalised}\left(x^K\right) + \beta^K \tag{9}$$

where y^K is the batch normalisation, x^K is the value after normalisation, γ^K is a scaling parameter, and β^K is the shifting parameter. Batch normalisation has the following advantages: it reduces the impact of bad initialisations; it improves the network accuracy by a margin, it speeds up the training and regularises the model and reduces the model overfitting.

A FC layer with 1024 kernels and ReLU activation is the last layer of CNN architecture. Finally, the FC connected layer produces the output of the whole network with K neurons that are connected directly to the last output layer with the soft-max classifier. The ReLU activation function is used in each layer of the convolution. In addition, the proposed CNN architecture also uses L2 weight regularisation, dropout of 0.5, Adam optimiser, and cross-entropy loss functions.

5.1.8 Optimisation: Optimisation is a large part of machine learning. Almost every machine learning algorithm has an optimisation algorithm at its core. Optimisation algorithms are used to train the CNN network by optimising the loss function or cost function. These functions depend on the learnable parameters of the network such as weights (W) and bias (b). It is necessary to find the right weights during the training process to reduce the loss function. For the auto-adjustment of neural network weights, optimisation is used to achieve fast convergence with higher efficiency. Basically, in response to the loss function output, optimisers joint the loss functions and model parameters together by updating the model weights. Generally, the cost function is defined as

$$C(W, b) = \sum_{i=1}^{m} L\left(\hat{y}^i, \; y^i\right) \tag{10}$$

where C is the cost function, L is the loss, y is the actual value, $\hat{y}$ is the predicted value, W is the weight, and b is the bias. Error is calculated as the difference between actual and predicted output in machine learning networks, i.e. output $= y - \hat{y}$. The function used to calculate the error is called the loss function. To deal with different types of regression and classification, different types of loss functions are used. In the present research, we utilise the cross-entropy loss and Adam optimiser. Cross-entropy is the most common loss function used in neural networks, which is defined as

$$H(P, q) = -\sum_{x}^{M} P(x) \log q(x) \tag{11}$$

where H is the cross-entropy loss, x is the class label, M is number of classes, $p(x)$ is the correct label probability, and $q(x)$ is the predicted probability. Cross-entropy can be directly used for multiclass classification [37].

Adam calculates adaptive learning rates for each parameter during the weight update process. Learning rate affects the training speed and network learning. A very low learning rate makes model learning difficult and train it slower; whereas, very high learning rate increases the sensitivity of outer values, and makes the model train too fast, but may get stuck in local minima. When training loss does not improve anymore for several iterations, the learning rate can be decreased by a factor: $\lambda = \lambda \times 0.1$. It helps to learn the fine-grained network differences in the data. Training parameters used in the present study are given in Table 1.

5.2 Defect classification using SVM

This sub-section introduces various features extraction techniques combined with training SVM for PV cells EL images classification.

5.2.1 SVM classification: The classification of PV cell defects is accomplished by training a model with EL images using a radial-based kernel SVM. To train the SVM model, features are first extracted from EL images of the cell using feature extraction techniques and then fed to the SVM classifier. Typically, the features are extracted at significant points that are also known as key points. The extracted features are classified into the proposed seven classes under study. However, the accuracy of the classifier generally depends on the training dataset size and quality of the features. The training data set of adequate size and classes of distinct nature can improve the performance.

5.2.2 Feature extraction: In the present study, PV cell EL image features are first extracted and then fed to train the SVM. For feature extraction, four feature descriptors such as HOG, KAZE, SIFT, and SURF are used and compared. These descriptors compare the average value of neighbourhood pixels with the centre pixel to find the location of the local key points. On the other side, key point detectors depend on the image's texture hence the number of high-frequency components, i.e. corners and edges are proportional to the number of key points. Feature detection and description algorithms usually work in scale-spaces to detect and describe features at different scale levels. We utilise different combinations of key point detectors and feature descriptors for the feature extraction task. The HOG has the only descriptor whereas, SIFT, SURF, and KAZE contains both detector and descriptor.

HOG is a gradient-based feature descriptor and computes the features by splitting the image window into small areas of space. Furthermore, accumulates a local 1-D gradient orientation histogram or cell pixels edge orientations for each region and combines all histogram entries for analysis [38]. Solar cell EL images HOG descriptors are shown in Fig. 1 (see the third row). SIFT is a multi-scale feature detection algorithm [39]. SIFT algorithm transforms the input image vector in a wide variety of local features that are invariant with respect to image rotation, translation, scaling; and partially invariant to changes in illumination. The stage filtering approach is utilised for feature detection that identified a stable point in scale space. The solar cell EL images SIFT features are shown in Fig. 1 (see the second row). KAZE is an algorithm for the automatic detection and description of multi-scale 2D features in non-linear scale spaces [40]. KAZE algorithm utilises additive operator splitting and variable conductivity diffusion techniques for the creation of non-linear scale-spaces. Both the scale and rotation key points are invariant to image resolution. Key point features of KAZE for cells are illustrated in Fig. 1 (see the fourth row).

SURF is a key points detector and a local feature descriptor. SURF descriptor describes how intensity content is distributed within the neighbourhood's interest point [41]. SURF works great for tasks of classification, object detection, and recognition. SURF key points features are presented in Fig. 1 (see the fifth row).

Table 1 Selected training parameters

Training parameters	Values
optimiser	Adam
learning rate	0.001
drop out	0.5
batch size	32
loss function	cross-entropy
epochs	150

5.2.3 Principle component analysis: It is an unsupervised machine learning algorithm that learns representation of data. The PCA algorithm makes it possible to compress input data. PCA learns an orthogonal, linear transformation of the data to transform the correlation of original variables linearly and non-linearly into a subspace where variables are not correlated. PCA is a simple and effective dimensionality reduction algorithm that preserves as much information as possible in the data. It is widely used for the extraction of features and data compression.

Dimensionality reduction: There are two main types of dimensionality reduction to decrease the complexity and avoid overfitting of a model, i.e. feature extraction and feature selection. In the feature extraction method, the feature set is used to derive the information, and a new feature subspace is constructed, whereas, in feature selection, original features are selected as a subset.

PCA algorithm is highly sensitive to data scaling and the image features need to be standardised before it [42]. The standardised dataset is used to create the covariance matrix and then decompose the covariant matrix into eigenvectors for dimensionality reduction. The appropriate size of the eigenvectors is $(n \times n)$ for change into PCA space, where the number of pixels per image is n. The top eigenvalues will help to change the matrix. The matrix of transformation is calculated by using the following equation:

$$K = \text{number of pixels}/2 \tag{12}$$

PCA space transformation: The new matrix $\boldsymbol{N}$ is created from the eigenvectors that are used as a matrix of transformation to change pixels of images into PCA space. This is achieved with the replacement of elements in the following equation:

$$\boldsymbol{Y} = N(\boldsymbol{P} - m_x)^t \tag{13}$$

where $\boldsymbol{Y}$ is the PCA-converted image vector, $\boldsymbol{P}$ is the image vector, and m_x is the mean value of each pixel location for all the training images dataset [43]. This is known as PCA transform. For the test set images, the same transformation process is applied.

5.2.4 SVM training: The extracted features from the PCA are passed to the SVM classifier to classify the all seven types of PV defect classes. We trained SVM with the radial-based function (RBF) kernel. The kernel function and kernel parameters such as gamma (γ) and capacity parameter (C) affect the SVM model's precision. To study the effectiveness of the classifier, a suitable selection of hyperparameters is therefore important. As mentioned, different combinations of C and γ parameters are selected for optimisation of the classification model. A grid search method is applied to select the best parameters. The details of these parameters are as follows: $C \in \{10^i,\ i = -2,\dots,4\}$ and $\gamma \in \{10^{-5},\ 10^{-2}\}$.

5.3 Evaluation criteria

There are many criteria in the classification field that can be used to evaluate the model. We should select an appropriate criterion based on the actual problem. We considered accuracy to illustrate the performance of SVM and CNN cell defects classification models. In addition, for the CNN model, recall rate, precision, and F1 score of the model are also considered. Under imbalance data conditions, these measuring factors perform better. Based on the

© The Institution of Engineering and Technology 2020

Table 2 Division of classes according to the module type

Module-type	Seven classes						
	c33d	c66d	ccd	cd	cesd	cmd	cnd
monocrystalline	106	52	112	127	41	82	554
polycrystalline	166	49	141	175	67	71	881
total	272	101	253	302	108	153	1435

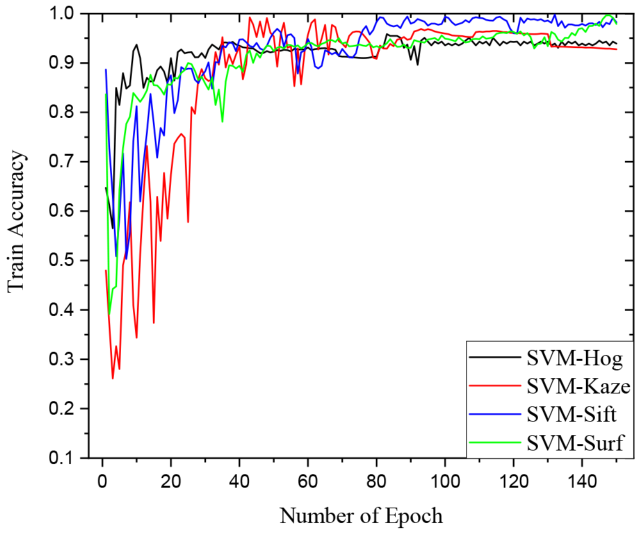

Fig. 4 *Train accuracies comparison for four SVM models*

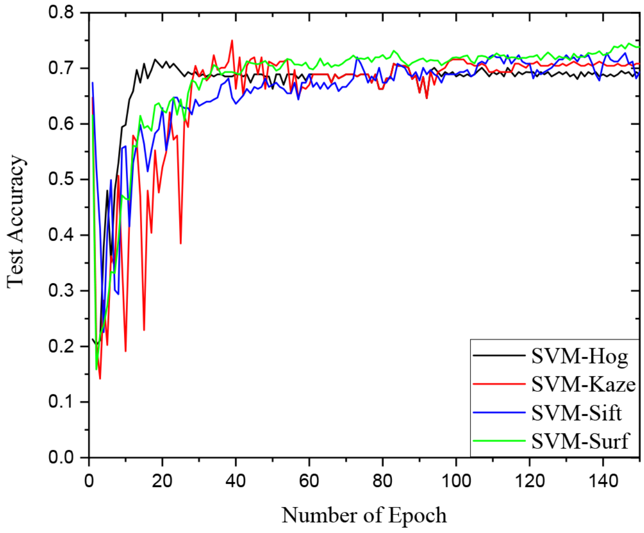

Fig. 5 *Test accuracies comparison for four SVM models*

correct and incorrect classification, the sample can be divided into positive, negative true values, and positive, negative false values. The accuracy of the model is the ratio between the number of accurate predictions and the total number of input samples and is given as

$$\text{Accuracy} = \frac{\text{true positive } (T_P) + \text{ true negatives } (T_N)}{\text{total number of samples}} \tag{14}$$

Precision, recall rate, and F1 score are calculated by using the (15), (16), and (17). The confusion matrix table is also designed to present the performance of multi-class model that describes the difference between classes and their relationship

$$\text{Precision} = \frac{\text{true positive } (T_P)}{\text{true positive } (T_P) + \text{ false positive } (F_P)}$$
$$\text{Precision} = \frac{\text{true positive } (T_P)}{\text{total predicted positive}} \tag{15}$$

$$\text{Recall rate} = \frac{\text{true positive } (T_P)}{\text{true positive } (T_P) + \text{ false negative } (F_N)}$$
$$\text{Recall rate} = \frac{\text{true positive } (T_P)}{\text{total actual positive}} \tag{16}$$

$$\text{F1 score} = 2 \times \frac{\text{precision} \times \text{recall rate}}{\text{precision} + \text{ recall rate}} \tag{17}$$

6 Results and discussion

This section describes the dataset and the details of experiment results, which are conducted with the proposed intelligent algorithms such as SVM with feature extraction techniques and CNN architecture.

The proposed algorithms are implemented with OpenCV, NumPy, PIL, TensorFlow, TFLearn, and SVC libraries in Python (Anaconda-Spyder). The experiments are carried out on a CPU of Intel Xeon E5–2680 v4, 2.4 GHz, and RAM of 128 GB and GPU of Nvidia GeForce GTX 1080 Ti.

6.1 Dataset

A public solar cell EL images dataset is used in our study [44]. This dataset is the first PV cells EL images dataset that is publicly available. This dataset comprises 2624 images and the image resolution is 300 × 300 pixels. This solar cell dataset is based on 44 different types of solar modules, consisting of 18 modules of monocrystalline material type and 26 polycrystalline modules.

In the present research, the dataset of 2624 EL images is divided into seven classes and labelled as non-defective cell (cnd), 33% defective cell (c33d), 66%defective cell (c66d), crack defective cell (ccd), defective cells (cd), electrically separated defective cells (cesd), and material defective cell (cmd). The details of these classes according to material type are given in Table 2.

6.2 Classification performance results

This section reports the EL images cell defect classification accuracy result of SVM and CNNs.

6.2.1 SVM classification accuracy: This sub-section deals with the defect classification accuracy results of PV cell EL images using feature extraction techniques such as HOG, KAZE, SIFT, and SURF. The SVM models for each of these feature extraction techniques are trained to obtain the best accuracy results. The input data for these models are EL images with a resolution of 100 × 100 pixels. The train and test defect classification accuracy results of the four features extraction algorithm-based models for SVM classifier versus number of epochs are shown in Figs. 4 and 5, respectively. The best parameters are set for SVM-RBF training. The models are trained for 150 epochs. The test accuracy results for HOG-SVM, KAZE-SVM, SIFT-SVM, and SURF-SVM are 69.95, 71.04, 68.90, and 72.74%, respectively. The findings of the classification test accuracies are shown in Table 3. It can be seen that the classification accuracy of the SURF-SVM model is better than the other three SVM models.

6.2.2 CNN classification accuracy: This sub-section deals with the CNN architecture-based defects classification accuracy results of PV cell EL images. In addition, this sub-section discusses other performance results.

Cross-validation (CV): CV is a statistical process that is often used in machine learning methods to validate the model's performance. The method is often called *k*-fold CV or rotation

Table 3 Test accuracies for feature extraction-based SVM models

Svm Features	SVM-HOG	SVM-KAZE	SVM-SIFT	SVM-SURF
accuracy	69.95 %	71.04 %	68.90 %	72.74 %

Table 4 Test accuracies of CNN model for 10-folds

No. of folds	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eighth	Nineth	Tenth	Average
accuracy, %	91.0	89.90	90.81	91.10	91.58	91.25	91.21	91.24	91.26	90.73	91.41

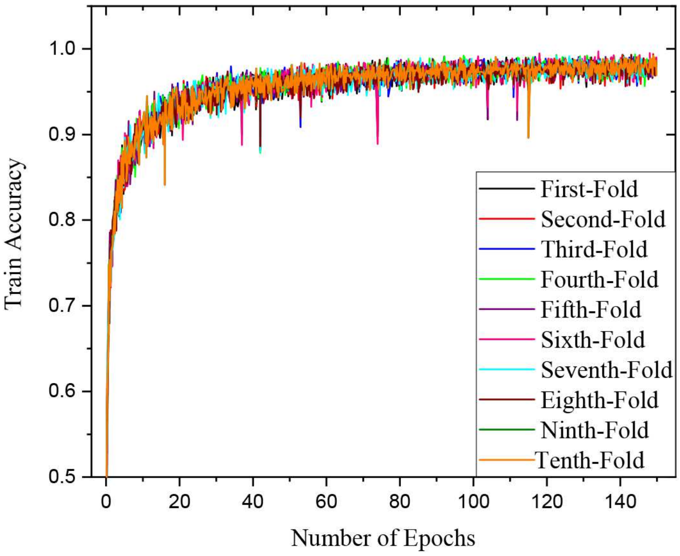

Fig. 6 *Train accuracies comparison for CNN models*

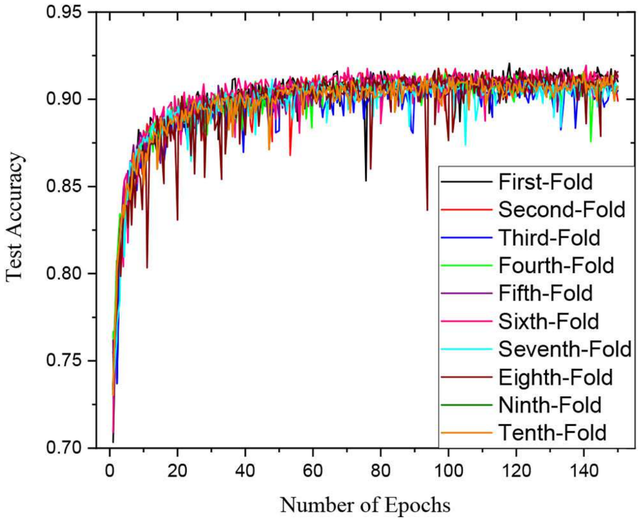

Fig. 7 *Test accuracies comparison for CNN models*

estimation where *k* is the number of folds in the dataset. In this research, prediction models are trained and evaluated for ten-fold CV, i.e. data are divided into ten folds and ten accurate results are obtained as explained below: During each turn, nine of the folds are used for training and remaining one fold is used for testing purpose. This is repeated ten times in a way that each fold gets tested once and finally we get ten results. The obtained accuracy results are 91.00, 89.90, 90.81, 91.10, 91.58, 91.25, 91.21, 91.24, 91.26 and 90.73%, respectively (see Table 4).

The overall average classification test accuracy for tenfold CV is 91.41%. The comparison of train and test defect classification accuracy results of the CNN model for ten-fold CV with respect to the number of epochs are shown in Figs. 6 and 7, respectively.

Confusion matrix and other performance assessment: We also constructed a confusion matrix to describe the relation and difference between classes. Recall, precision values, and F1 score are also reported. To evaluate the performance of the model according to these measures, train and test data are randomly split into 90 and 10%, respectively, and the model is trained for 150 epochs. The confusion matrix of the model is shown in Table 5.

It can be observed from the confusion matrix table that 2.3% of the crack cell defect EL images (ccd) are misclassified into other classes. This is the least misclassified class. On the other hand, cell defect class (cd) is the most misclassified class, i.e. 13.45% of cd images are misclassified into the other six classes. In addition, the electrically separable cell defect (cesd) class is also highly misclassified, i.e. 12.3% of cesd images are misclassified.

The precision, recall, and F1-score are shown in Table 6. The proposed CNN architecture achieves average precision of 91.57%, recall of 91.85%, and F1-score of 91.57%.

6.2.3 Prediction results: In this section, a few correct classified and misclassified prediction results of CNN architecture for EL cell images are described. The prediction results of the CNN-based

Table 5 Confusion matrix for seven defects classes

Classes	c33d	c66d	ccd	cd	cesd	cmd	cnd
c33d	**93.7**	0.89	0	1.68	1.68	0.22	1.8
c66d	1.6	**93.1**	0.11	0.89	1.34	0.56	2.46
ccd	0	0.21	**97.7**	0.95	1.05	0.11	0
cd	3.13	1.14	0.28	**86.6**	5.49	0.85	2.56
cesd	3.8	0.89	0.33	3.90	**87.7**	1.0	2.34
cmd	1.72	0.54	0	1.72	0.43	**92.6**	3.02
cnd	3.9	1.33	0	1.33	2.1	0.95	**90.4**

The bold values are representing the correct classification percentage of defective or non defective cell classes.

Table 6 Model accuracy, precision, recall, and F1 score

Classes	c33d	c66d	ccd	cd	cesd	cmd	cnd
precision	0.86	0.94	0.99	0.90	0.87	0.96	0.89
recall	0.94	0.93	0.98	0.87	0.88	0.93	0.90
F1-score	0.90	0.94	0.98	0.88	0.87	0.94	0.90

Overall model accuracy at the 150th epoch is 91.58%.

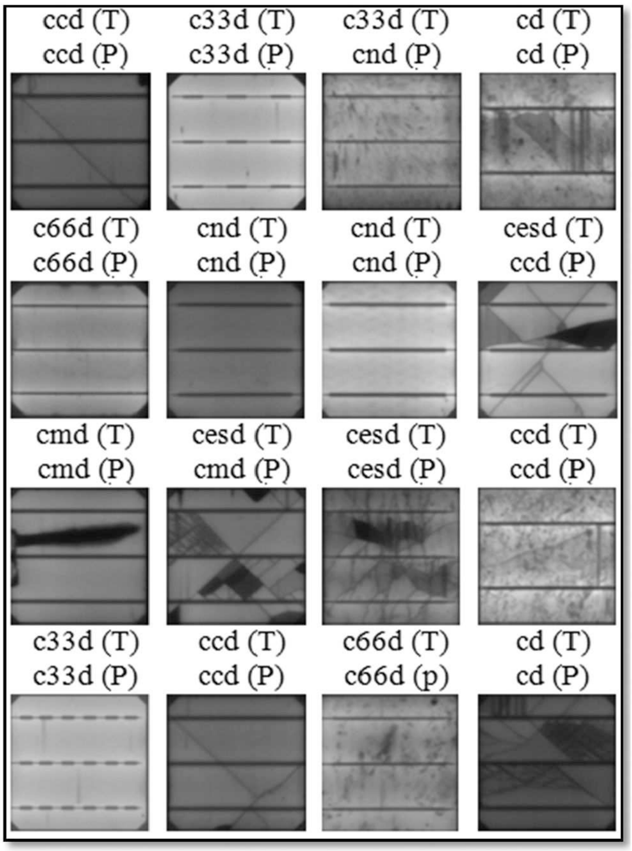

Fig. 8 *CNN model prediction results for correct classification and misclassification (T and P Indicate true and predicted labels, Respectively)*

models for solar cell defects classification are shown in Fig. 8. The prediction of the model in the first row and column three is incorrect; originally the cell is 33% defected but the model prediction for this cell is not defective. Furthermore, in the second row and column four, the cell is electrically separable defective, but the prediction is cell crack defective. Moreover, in the third row and column two, the cell is electrically separable defective, but the prediction is cell material defective. The predicted and true labels for classification are on top of the images indicated by P and T, respectively. Some of the cell's images have a high probability to be misclassified. The reason for misclassification is that there is more than one defect inside a single solar cell, and another reason can be the material related.

Table 7 Present study comparisons with the existing studies

Studies	Dataset	Prediction-method	Accuracy, %
[14]	PV-cell public dataset	CNN-transfer learning method	88.42
		linear-SVM with KAZE/VGG features	82.44
our study	PV-cell public dataset (same in the existing study)	Hog- SVM	69.95
		KAZE-SVM	71.04
		SIFT-SVM	68.90
		SURF-SVM	72.74
		CNN	91.58

6.2.4 Comparison: This section includes the comparison of our study result with the result obtained by existing studies. The existing study [14] obtained results in their research by using the public PV cell EL images dataset. In their study, they utilised SVM and CNN algorithms. They obtained the best average results of 82.44% with linear SVM using KAZE/VGG features and of 88.42% with CNN using the transfer learning approach. The comparison results are shown in Table 7. In our research, we conducted a series of experiments with RBF SVM using features extraction algorithms such as HOG, KAZE, SIFT, and SURF, but with CNN, we achieved better performance than the existing study. We get state-of-the-art results of 91.58% for seven classes of solar cell defects.

7 Conclusions and future work

In this research, features extraction-based SVM and CNN methods are presented for the classification of solar cell defects. The successful classification of defects in a polycrystalline silicon PV cell is a challenging task due to its background texture. To classify the seven types of cell defects, the proposed machine learning approaches are applied to the public dataset of solar cell EL images. Furthermore, suitable hyperparameters, algorithm optimisers, and loss functions are considered in the training process to achieve the best performance. Moreover, data augmentation techniques such as rotation, flipping, cropping, contrast, and Gaussian blurring are also chosen to obtain the best accuracy results.

The performance of the proposed features extraction-based RBF-SVM methods (HOG, KAZE, SIFT, and SURF) is compared with CNN architecture; and it is concluded that the CNN architecture for classification of solar cell defects is more accurate than SVM. CNN's accuracy for solar cell defect classification is 91.58%, which outperforms the state-of-the-art methods. With

SVM, we obtain accuracies of 69.95, 71.04, 68.90 and 72.74% for HOG, KAZE, SIFT, and SURF, respectively. The proposed CNN model can correctly classify the images into the correct class. The proposed methods can be useful for auto inspection of PV cell defects during manufacturing and operation phases to improve the quality and reliability.

In the future, a large labelled EL images cell dataset can be collected with a large number of images for each class. In addition, the hybrid algorithm techniques can also be used.

8 References

[1] Su, B., Chen, H., Zhu, Y., *et al.*: 'Classification of manufacturing defects in multicrystalline solar cells with novel feature descriptor', *IEEE Trans. Instrum. Meas.*, 2019, **68**, pp. 1–14, doi: 10.1109/tim.2019.2900961

[2] Nasr Esfahani, S., Asghari, S., Rashid-Nadimi, S.: 'A numerical model for soldering process in silicon solar cells', *Sol. Energy*, 2017, **148**, pp. 49–56, doi: 10.1016/j.solener.2017.03.065

[3] Li, G., Akram, M.W., Jin, Y., *et al.*: 'Thermo-mechanical behavior assessment of smart wire connected and busbar PV modules during production, transportation, and subsequent field loading stages', *Energy*, 2019, **168**, pp. 931–945, doi: 10.1016/j.energy.2018.12.002

[4] Crozier, J.L.: 'Characterization of cell mismatch in photovoltaic modules using electroluminescence and associated electro-optic techniques magister scientiae', *Doctoral dissertation, Nelson Mandela Metropolitan University*, 2012, **40**, pp. 1–87

[5] Tománek, P., Škarvada, P., MacKů, R., *et al.*: 'Detection and localization of defects in monocrystalline silicon solar cell', *Adv. Opt. Technol.*, 2010, **2010**, pp. 1–5, doi: 10.1155/2010/805325

[6] Waqar Akram, M., Li, G., Jin, Y., *et al.*: 'Improved outdoor thermography and processing of infrared images for defect detection in PV modules', *Sol. Energy*, 2019, **190**, pp. 549–560, doi: 10.1016/j.solener.2019.08.061

[7] Fuyuki, T., Kondo, H., Yamazaki, T., *et al.*: 'Photographic surveying of minority carrier diffusion length in polycrystalline silicon solar cells by electroluminescence', *Appl. Phys. Lett.*, 2005, **86**, pp. 1–3, doi: 10.1063/1.1978979

[8] Buerhop, C., Wirsching, S., Bemm, A., *et al.*: ''Evolution of cell cracks in PV-modules under field and laboratory conditions', *Prog. Photovoltaics Res. Appl.*, 2018, **26**, pp. 261–272, doi: 10.1002/pip.2975

[9] Breitenstein, O., Bauer, J., Bothe, K., *et al.*: 'Luminescence imaging versus lock-in thermography on solar cells and wafers'. Proc. 26th European Photovoltaic Solar Energy Conf. and Exhibition, USA, vol. 1, 2011, pp. 1031–1038. http://www.mpi-halle.de/mpi/publi/pdf/10436_11.pdf

[10] Tsai, D.M., Wu, S.C., Li, W.C.: 'Defect detection of solar cells in electroluminescence images using Fourier image reconstruction', *Sol. Energy Mater. Sol. Cells*, 2012, **99**, pp. 250–262, doi: 10.1016/j.solmat.2011.12.007

[11] Tseng, D.C., Liu, Y.S., Chou, C.M.: 'Automatic finger interruption detection in electroluminescence images of multicrystalline solar cells', *Math. Probl. Eng.*, 2015, **2015**, pp. 1–13, doi: 10.1155/2015/879675

[12] Tsai, D.M., Fan, M.S.K., Huang, Y.Q., *et al.*: 'Saw-mark defect detection in heterogeneous solar wafer images using GAN-based training samples generation and CNN classification'. VISIGRAPP 2019 – Proc. 14th Int. Jt. Conf. on Computer Vision, Imaging and Computer Graph Theory and Applications, China, vol. 5, 2019, pp. 234–240. doi: 10.5220/0007306602340240

[13] Chen, H., Pang, Y., Hu, Q., *et al.*: 'Solar cell surface defect inspection based on multispectral convolutional neural network', *J. Intell. Manuf.*, 2018, **31**, pp. 1–14, doi: 10.1007/s10845-018-1458-z

[14] Deitsch, S., Christlein, V., Berger, S., *et al.*: 'Automatic classification of defective photovoltaic module cells in electroluminescence images', *Sol. Energy*, 2019, **185**, pp. 455–468, doi: 10.1016/j.solener.2019.02.067

[15] Chiou, Y.C., Liu, J.Z., Liang, Y.T.: 'Micro crack detection of multi-crystalline silicon solar wafer using machine vision techniques', *Sens. Rev.*, 2011, **31**, pp. 154–165, doi: 10.1108/02602281111110013

[16] Fu, Z., Zhao, Y., Liu, Y., *et al.*: 'Solar cell crack inspection by image processing'. Proc. 2004 Int. Conf. on Business of Electronic Product Reliability and Liability 200030, Shanghai, China, 2004 pp. 77–80. doi: 10.1109/beprl.2004.1308153

[17] Anwar, S.A., Abdullah, M.Z.: 'Micro-crack detection of multicrystalline solar cells featuring an improved anisotropic diffusion filter and image segmentation technique', *Eurasip J. Image Video Process.*, 2014, **2014**, pp. 1–17, doi: 10.1186/1687-5281-2014-15

[18] Wang, X.B., Li, J., Yao, M.H., *et al.*: 'Solar cells surface defects detection based on deep learning', *Moshi Shibie Yu Rengong Zhineng/Pattern Recognit. Artif. Intell.*, 2014, **27**, pp. 517–523

[19] Qian, X., Zhang, H., Zhang, H., *et al.*: 'Solar cell surface defects detection based on computer vision', *Int. J. Performability Eng.*, 2017, **13**, pp. 1048–1056, doi: 10.23940/ijp.17.07.p6.10481056

[20] Demant, M., Welschehold, T., Oswald, M., *et al.*: 'Microcracks in silicon wafers I: inline detection and implications of crack morphology on wafer strength', *IEEE J. Photovoltaics*, 2016, **6**, pp. 126–135, doi: 10.1109/JPHOTOV.2015.2494692

[21] Tsai, D.M., Li, G.N., Li, W.C., *et al.*: 'Defect detection in multi-crystal solar cells using clustering with uniformity measures', *Adv. Eng. Inf.*, 2015, **29**, pp. 419–430, doi: 10.1016/j.aei.2015.01.014

[22] Lecun, Y., Bottou, L., Bengio, Y., *et al.*: 'Gradient-based learning applied to document recognition', *Biochem. Biophys. Res. Commun.*, 2005, **330**, pp. 1299–1305, doi: 10.1016/j.bbrc.2005.03.111

[23] Banda, P., Barnard, L.: 'A deep learning approach to photovoltaic cell defect classification', *Association for Computing Machinery*, n.d., **18**, pp. 215–221

[24] Jahn, U., Herz, M., Köntges, M., *et al.*: 'Review on Infrared and Electroluminescence Imaging for PV Field Applications Report IEA-PVPS T13- 10', 2018

[25] Bedrich, K., Bokalic, M., Bliss, M., *et al.*: 'Electroluminescence imaging of PV devices: advanced vignetting calibration', *IEEE J. Photovoltaics*, 2018, **8**, pp. 1297–1304, doi: 10.1109/JPHOTOV.2018.2848722

[26] Fraz??o, M., Silva, J.A., Lobato, K., *et al.*: 'Electroluminescence of silicon solar cells using a consumer grade digital camera', *Measurement*, 2017, **99**, pp. 7–12, doi: 10.1016/j.measurement.2016.12.017

[27] Köntges, M., Kurtz, S., Packard, C., *et al.*: 'Review of Failures of Photovoltaic Modules-Report IEA-PVPS T13-01:2014', 2014. doi: 978-3-906042-16-9

[28] Fuyuki, T., Kitiyanan, A.: 'Photographic diagnosis of crystalline silicon solar cells utilizing electroluminescence', *Appl. Phys. A, Mater. Sci. Process.*, 2009, **96**, pp. 189–196, doi: 10.1007/s00339-008-4986-0

[29] Brownlee, J.: 'Master machine learning algorithms: discover how they work and implement them from scratch' in '*Machine Learning Mastery with Python*' (Australia, 2016), **1**, p. 163. Available at http://machinelearningmastery.com

[30] Bhavsar, H., Ganatra, A.: 'A comparative study of training algorithms for supervised machine learning', *Int. J. Soft. Comput. Eng.*, 2012, **2**, pp. 74–81, doi: 10.1.1.492.6088

[31] Ren, S., He, K., Girshick, R., *et al.*: 'Faster R-CNN: towards real-time object detection with region proposal networks', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, **39**, pp. 1137–1149, doi: 10.1109/TPAMI.2016.2577031

[32] Redmon, J., Divvala, S., Girshick, R., *et al.*: 'You only Look once: unified, real-time object detection', CVPR, Las Vegas, NV, USA, 2016, doi: 10.1109/CVPR.2016.91

[33] Wu, J.: 'Introduction to convolutional neural networks'. National Key Laboratory Novel Software Technology, 2017, **1**, pp. 1–31, doi: 10.1007/978-3-642-28661-2-5

[34] Goodfellow, I., Bengio, Y., Courville, A.: 'Deep learning', *Data Sci.*, 2019, **1**, pp. 307–342, doi: 10.1016/b978-0-12-814761-0.00010-1

[35] Géron, A.: '*Hands-On machine learning with scikit-learn and TensorFlow*' (O'Reilly Media, CA, USA, 2017)

[36] Bourez, C.: '*Deep learning with theano build the artificial brain of the future*' (Today, Birmingham, UK, 2017)

[37] Chen, S., Wang, Y.: 'Convolutional neural network and convex optimization', Tech. Rep., 2014, **14**, pp. 1–11

[38] Dalal, N., Triggs, B., Europe, D.: 'Histograms of oriented gradients for human detection', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, 2005

[39] Lowe, D.G.: 'Object recognition from local scale-invariant features', *Proceedings of IEEE international conference on computer vision*, 2016, **2**, p. 150

[40] Alcantarilla, P.F., Bartoli, A., Davison, A.J.: 'KAZE features'. ECCV'2012 (LNCS, **7577**), Berlin, Heidelberg, 2012, pp. 214–227. doi: 10.1007/978-3-642-33783-3_16

[41] Bay, H., Ess, A., Tuytelaars, T., *et al.*: 'Speeded-up robust features (SURF)', *Comput. Vis. Image Underst.*, 2008, **110**, pp. 346–359, doi: 10.1016/j.cviu.2007.09.014

[42] Tsai, D.M., Chen, M.C., Li, W.C., *et al.*: 'A fast regularity measure for surface defect detection', *Mach. Vis. Appl.*, 2012, **23**, pp. 869–886, doi: 10.1007/s00138-011-0403-3

[43] Chaouch, H., Najeh, T., Nabli, L.: 'Multi-variable process data compression and defect isolation using wavelet PCA and genetic algorithm', *Int. J. Adv. Manuf. Technol.*, 2017, **91**, pp. 869–878, doi: 10.1007/s00170-016-9774-y

[44] Buerhop-Lutz, C., Brabec, C.J., Camus, C., *et al.*: 'A benchmark for visual identification of defective solar cells in electroluminescence imagery'. 35th European Photovoltaic Solar Energy Conf. and Exhibition, Brussels, Belgium, 2018, pp. 1287–1289. doi: 10.4229/35THEUPVSEC20182018-5CV.3.15