



# Efficient deep feature extraction and classification for identifying defective photovoltaic module cells in Electroluminescence images

Mustafa Yusuf Demirci <sup>a,\*</sup>, Nurettin Beşli <sup>a</sup>, Abdulkadir Gümüşçü <sup>a</sup>

<sup>a</sup> Department of Electrical and Electronics Engineering, Engineering Faculty, Harran University, Şanlıurfa, Turkey

## ARTICLE INFO

**Keywords:**

Electroluminescence imaging  
Defect detection  
Feature extraction  
Feature selection  
Deep features  
Deep learning

## ABSTRACT

Electroluminescence (EL) imaging has become the standard test procedure for defect detection throughout the production, installation and operation stages of solar modules. Using this test, defects such as micro cracks, broken cells, and finger interruptions on photovoltaic modules could be easily detected and potential power loss issues could be effectively addressed. Although EL test is a very powerful inspection method, interpreting the EL images could be quite challenging due to the inhomogeneous background and complex defect patterns. Therefore, evaluating the damaged cells and determining the defect severity require expertise, and could be time consuming to apply these processes manually for each cell. Hence, the automatic visual inspection of photovoltaic cells is very important. In this study, a novel automatic defect detection and classification framework for solar cells is proposed. In the proposed Deep Feature-Based (DFB) method, the image features extracted through deep neural networks are classified with machine learning methods such as support vector machines, K-Nearest Neighborhood, Decision Tree, Random Forest and Naive Bayes. Thus, classical machine learning and deep learning techniques are used together. In order to combine the features taken from different deep network architectures in various combinations, the minimum Redundancy Maximum Relevance (mRMR) algorithm is employed for the feature selection. In this way, the dimensions of the feature vectors are reduced and the classification performance is increased with fewer features. With the determination of the best features extracted from different layers of deep neural networks, state-of-the-art results were obtained for both 4-class and 2-class datasets. Moreover, a Lightweight Convolutional Neural Network (L-CNN) architecture has been proposed and trained from scratch, and the results are compared with previous works. As a result, the highest scores are obtained using DFB method with Support Vector Machines (SVM) and classification scores of 90.57% and 94.52% were obtained for the dataset with 4 - class and 2 - class, respectively. The proposed DFB-SVM models outperformed other studies using the same dataset. The results showed that the proposed framework can detect PV cell defects with high accuracy.

## 1. Introduction

Solar energy systems have gained a broader attention in recent years as renewable energy source. Over the last decade, huge solar power plants have been built all over the world, which led to the activation of the large-scale production facilities that produce solar energy components. In 2019, the global annual PV production is estimated to be around 115 GW, up 12%, and the total PV capacity is 627 GW (REN21, 2020). One of the main components of solar energy systems is the PhotoVoltaic (PV) module. In order for solar energy systems to work efficiently, solar modules must also operate at high efficiency. It is vital to determine the module performance quickly and take necessary

interventions to avoid wasting resources, especially in medium-large scale solar power plants. For manufacturing plants, quality control is an obligation. This usually involves a detailed inspection of each solar cell before use. Although there are different types of PV modules in the market, the dominant technology is Crystalline Silicon (c-Si) due to its well-known production method and lower cost per power. Monocrystalline and polycrystalline modules with this technology cover up to 97% of the market (Tang, Yang, Xiong, & Yan, 2020). These PV modules could be exposed to different defects and malfunctions not only in production but also in assembly, transportation and operation stages in the field. Defects like Potential Induced Degradation (PID) may occur within time. However, defects like cracks and fractures can be caused by

\* Corresponding author.

E-mail address: [mfecidpc@gmail.com](mailto:mfecidpc@gmail.com) (M.Y. Demirci).

environmental conditions such as snow, storm or hail (Jahn, Herz, Kötges, Parlevliet, Paggi, Tsanakas, & Tanahashi, 2018). Some defects such as finger interruptions, micro cracks and dislocated areas may occur during manufacturing and can be detected in the production phase due to the non-uniformity of silicon wafer. Defects like cell disconnections, soldering malfunctions and lamination flaws may occur during the assembly stage. All these defects adversely affect the operation of a PV module in different manner. Malfunctions such as PID, cracked cells, cell disconnections and fractures could seriously reduce the power output of the entire module (Kötges, Kurtz, Packard, Jahn, Berger, Kato, & Van Iseghem, 2014).

### 1.1. Background

The main tests used to detect the defects mentioned above for solar panels are I-V curve measurement, thermal - infrared imaging (IR) and Electroluminescence (EL) imaging. I-V curve method is based on measuring the output voltage and current of PV modules under certain radiation and visualizing it on a graph. Although it is possible to determine overall condition of an entire module with the I-V curve, it is not possible to determine which cells are defective and the exact location of defects. IR imaging is another common method for visualization of the solar modules and cells conditions. While unexpected high current causes heating in a region, an open circuit may cause that area to be at low temperature. Therefore, IR imaging can detect defect such as short circuits, hot spots, and inactive cells. However, it is not possible to detect defects such as micro cracks effectively with this method due to relatively low resolution of thermal cameras (Akram et al., 2019; Deitsch et al., 2019). It is possible to overcome such problems by EL imaging. EL test is a non-destructive method that effectively reveals various faults on the PV modules. EL imaging can provide a detailed examination of all cells in PV modules and can easily detect defective cells and regions. This makes it possible to determine the overall module condition and lifetime at all stages (Jahn et al., 2018). However, the biggest drawback in EL imaging is the interpretation process after the imaging. This is mostly carried out by experts who manually inspect every single EL image taken. This process is usually very time consuming and requires specially trained staff. Therefore, manual inspection in large-scale production facilities and power plants is not efficient enough. An automated EL image interpretation would make a big leap forward to speed up production, fault detection and quality control. Thus, automated inspection of EL images has become an active research area, and also brought new challenges. Due to the intrinsic crystal structure of solar cells and the wide variety of unique defect patterns, it is a crucial task to automatically Interpret and classify EL images (Balzategui, Eciolaza, & Arana-Arexolaleiba, 2020; Su & Chen, 2020). Especially for the polycrystalline cells, inhomogeneous cell structure and grain boundaries make it confusing to distinguish between the defective and non-defective cell parts (Rahman & Chen, 2020). In addition, complex structures of solar cell and variable defect scales are the other issues. All these issues make the solar cell classification a real challenge.

### 1.2. Related work

Automated classification and defect prediction on EL images have been shown to produce successful results, especially using artificial intelligence techniques (Akram et al., 2019; Bartler, Mauch, Yang, Reuter, & Stoicescu, 2018; Deitsch et al., 2019). The use of Deep Neural Networks (DNN) in image recognition and classification has become quite common due to the high scores they achieved. In the ImageNet (ILSVRC) contest, DNNs such as AlexNet, GoogleNet, VGG have become very popular with very high accuracy (Krizhevsky, Sutskever, & Hinton, 2012; Russakovsky et al., 2015; Simonyan & Zisserman, 2015; Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, and Rabinovich, 2015). Anwar and Abdullah (2013) implemented a shape feature extraction on polycrystalline EL images and classified them with Support Vector Machines

(SVM). However, the study was only carried out on the multicrystalline type cells. (Bartler et al., 2018) performed a classification of polycrystalline PV cell images on the dataset created with CNN. The study is carried out by adapting the VGG-16 deep network; data augmentation is implemented and solar cells were classified as functional and defective (2-class). Here again, only polycrystalline solar cells have been studied. (Karimi, Fada, Liu, Braid, Koyuturk, & French, 2018) implemented supervised and unsupervised learning on EL cell images for 2-class classification. SVM and CNN were used as supervised learning mechanism and clustering is used as unsupervised learning mechanism. (Karimi et al., 2019) classified the EL image dataset of PV modules using SVM, Random Forest (RF) and CNN. Authors implemented data augmentation strategies by dividing the cells into 3 categories as good, corroded and cracked. Despite the high accuracy, the ratio of the cracked cells was 3.5% in the dataset. Wuqin and Quiang Yang (2019) classified monocrystalline EL cell images by training DNN, and achieved over 90% accuracy with an augmented dataset. (Tong, Haiyong, Image, & Cells, 2018) classified EL PV cell images using Multi-Channel Convolutional Neural Networks (MCCNN) and RF classifier. In the study, only polycrystalline cell data set was used and high performance is achieved for the classification of 5 defect classes. (Mathias et al., 2020) proposed a micro-crack detection method of solar cell EL images using discrete wavelet transform and stationary wavelet transform. Cell classification is assigned as cracked and non-cracked using SVM and Back Propagation Neural Network (BPNN).

In summary, most of these studies performed well. However, it is not possible to make an objective comparison due to the use of different datasets. Therefore, these solutions mostly remain problem-specific. In the literature, there is a PV cell EL image data set named "ELPV", which is the first and only one available to the public, to our best knowledge (Buerhop-Lutz, Deitsch, Maier, Gallwitz, & Brabec, 2018; Deitsch et al., 2018). Using this dataset, (Deitsch et al., 2019) performed PV cell classification on the original dataset with 4-class (i.e. Non-defected, Possibly normal, Possibly defected and Defected). Classification with SVM and CNN is performed, and 82.44% and 88.42% accuracy is achieved for SVM and CNN, respectively. (Akram et al., 2019) classified the EL images as 2-class (i.e. functional, defective) using the same ELPV dataset. They trained the DNN from scratch, applied different data augmentation techniques and achieved 93.02% accuracy. (Tang et al., 2020) also based their research on ELPV dataset and classified PV cell EL images according to their defect types by creating synthetic data using Generative Adversarial Networks (GAN). CNN were trained from scratch and the classification accuracy between 81% and 84% is achieved. Another GAN - based approach is proposed by Luo, Cheng, and Zheng (2019). A method to augment the monocrystalline EL image dataset with generated samples is proposed. They achieved up to 14% improvement on classification accuracy using the pre-trained models of SqueezeNet, ResNet and AlexNet.

For DNN training, it is common to use transfer learning with pre-trained DNN instead of training it from scratch. In our previous study, transfer learning is performed on the above-mentioned ELPV dataset with different pre-trained DNNs and the highest accuracy of 78.96% is achieved. Using transfer learning, training time and computational load have been significantly reduced, but the performance has remained relatively low (Demirci, Beşli, & Gümüşçü, 2019). The superior success of DNN architectures in image recognition is provided by the use of DNN in feature extraction. CNNs have succeeded in extracting high-level distinctive features with their multi-layered structure. The use of deep features provides high performance in different defect detection and object recognition applications (Cibuk, Budak, Guo, Cevdet Ince, & Sengur, 2019; Demir, Şengür, & Çavaş, 2019; Hou, Wei, Jin, & Zhu, 2019; Jiao, Gao, Wang, & Li, 2016; Maeda-Gutiérrez, Galván-Tejada, Zanella-Calzada, Celaya-Padilla, Galván-Tejada, Gamboa-Rosales, & Olvera-Olvera, 2020; Sengur, Akbulut, Budak, & Comert, 2019; Toğaçar, Ergen, Cömert, & Özurt, 2020). In addition, deep features could be used for defect location. (Qian, Li, Cao, Wu, & Wang, 2020)

performed micro-crack detection on the solar cell images using short-term and long-term deep features using pre-trained VGG-16 DNN model. However, this method only locates defective regions, and requires pre-processing of the images. (Su & Chen, 2020) proposed a complementary attention network for defective PV cell classification using pre-trained weighted DNNs. (Rahman & Chen, 2020) utilized a multi attention network using modified U-net architecture based on DNN to detect defects on solar cell EL images.

In this study, deep features and efficient CNN architectures are used for the end-to-end robust classification of the EL images. Initially, data augmentation techniques are utilized to increase defect variety and to address class imbalance. Then, image features are extracted using activations of DNNs such as DarkNet-19, ResNet-50, VGG-16 and VGG-19 on the ELPV dataset mentioned above. The extracted features were concatenated to obtain new larger feature vectors. Then, feature selection is performed by using the mRMR algorithm to find out the best features of these concatenated feature vectors. Thus, only the most distinctive features are selected. After that, classification is performed using Support Vector Machine (SVM), k-Nearest Neighborhood (KNN), Random Forest (RF) and Naive Bayes (NB) on the selected features. Moreover, a Lightweight CNN (L-CNN) approach is proposed and this network is trained from scratch without any initial weights. The results were obtained for the 2-class and 4-class classification defects on the ELPV dataset. The performances of the classifications are compared, and quality metrics are calculated. The technical contributions of this study are summarized below:

1. A novel compact model framework is developed to accurately classify the solar cell EL images.
2. End-to-end Lightweight Convolutional Neural Network architecture (L-CNN) is proposed for fast learning and efficient training.
3. Using the proposed system, 90.57% test accuracy in 4-class dataset and 94.52% test accuracy in 2-class dataset are obtained. Thereby, higher accuracy scores were obtained compared to other studies using the same dataset.
4. In the proposed Deep Feature-Based (DFB) method, DNNs were not trained from scratch. Only activation values were extracted as feature vectors from specific layers of different pre-trained DNNs. Then, these feature vectors are classified with SVM. In this way, a general scalable lightweight architecture with a lower resource usage was developed.
5. Despite the small number of data and imbalanced dataset, high performance has been achieved by utilizing simple data augmentation methods and efficient feature selections. Owing to the feature selection, better results were obtained with fewer features.

Rest of the article is organized as follows; in Section 2, an overview about PV cell defects and EL imaging is given. In Section 3, main methodology and the components of the proposed methods such as dataset, data augmentation, feature extraction, feature selection and classification algorithms are explained. In Section 4, the experimental results of all proposed models and the comparison with previous works are presented. Section 5 contains discussions and implications. Finally, conclusions and future works are given in Section 6.

## 2. EL imaging and PV cell defects

### 2.1. EL imaging

Electroluminescence imaging is an imaging technique, based on the principle that when the c-Si semiconductor is forward biased, the energy released by the electron-hole recombination emits as photons and these spreading photons are captured by suitable digital cameras (Fuyuki, Kondo, Kaji, Yamazaki, Takahashi, & Uraoka, 2005). In this method, a specific amount of current passes through the PV cell or whole module and emitting radiation from the module in a dark environment is

captured by a camera. This radiation with the wavelength of approximately 1100 nm is in invisible spectrum and is indicated as near-infrared (Evans, 2014). A typical EL imaging setup is given in Fig. 1.

### 2.2. PV cell defects

Although the defects encountered in PV cells are various, every defect may not indicate a serious power loss. Some defects do not affect the power output of the module, but some defects may cause the cells to disconnect or to decrease the power produced by the module over time (Köntges M. et al., 2014). Cell damages that could be revealed by EL imaging could generally be categorized as cracks, micro cracks and fractures, disconnections, silicone material defects, finger interruptions and disconnected cells. In Fig. 2, cells with various defects are given as examples, and defected areas are marked with red rectangles. EL imaging is the only way to detect micro cracks especially among these defects. Defects occur often due to excessive mechanical stress in the manufacturing processes or in the transportation or during installation phases of solar power plants. Micro cracks could be caused by stress, weather conditions, etc. over time. Some parts of the PV modules may become completely inactive and disconnected because of broken cells due to environmental effects such as hail or snow. When inactive cells are between 12% and 50% of the total area in a 60-cell PV module, a linear increase from zero to 100% power loss occurs (Kontges, Kunze, Kajari-Schröder, Breitenmoser, & Bjørneklett, 2010). Automatic inspection of the EL cell images provides predetermination of defects effectively and would help prevent power losses due to defective modules.

## 3. Methodology

In this study, two different classification frameworks are proposed. For both methods, dataset, data augmentation settings and calculated metrics are the same. In the first approach, data augmentation is applied to the dataset first and features are extracted from the dataset with four different deep network architectures. Feature vectors are obtained using DarkNet-19, ResNet-50, VGG-16 and VGG-19 pre-trained DNNs. Then, the extracted features are combined. Finally, mRMR algorithm is utilized to determine the best features. After the feature selection, classification with different machine learning methods is performed. In the second approach, a lightweight CNN architecture (L-CNN) is developed and trained from scratch without initial weights. General representation of the proposed system is given in Fig. 3.

### 3.1. Dataset and data augmentation

#### 3.1.1. Dataset

ELPV dataset, which is created by Buerhop et.al., is available for public use. This unique dataset consists of 2624 EL solar cell images obtained from different monocrystalline and polycrystalline PV panels.

Dataset is originally labeled with 4 classes according to the defect condition by experts (Buerhop-Lutz et al., 2018). These classes are designated as Non-defected (0% defect rate), Possibly normal (33% defect rate), Possibly defected (66% defect rate) and Defected (100% defect rate). Number of images in these classes are 1508, 295, 106, 715, respectively. Fig. 4 shows the distribution of cell images in the dataset by classes. As can be seen, the dataset has relatively small number of images for machine learning and deep learning training, and the inter-class ratio is highly imbalanced. For this reason, data augmentation and balancing strategies should be considered. Moreover, DNNs are often sensitive to the class imbalance (Johnson & Khoshgoftaar, 2019). Therefore, data augmentation strategies have been considered in this study. In addition, decision of the classification algorithm is also important here and this situation is discussed in Section 3.3.

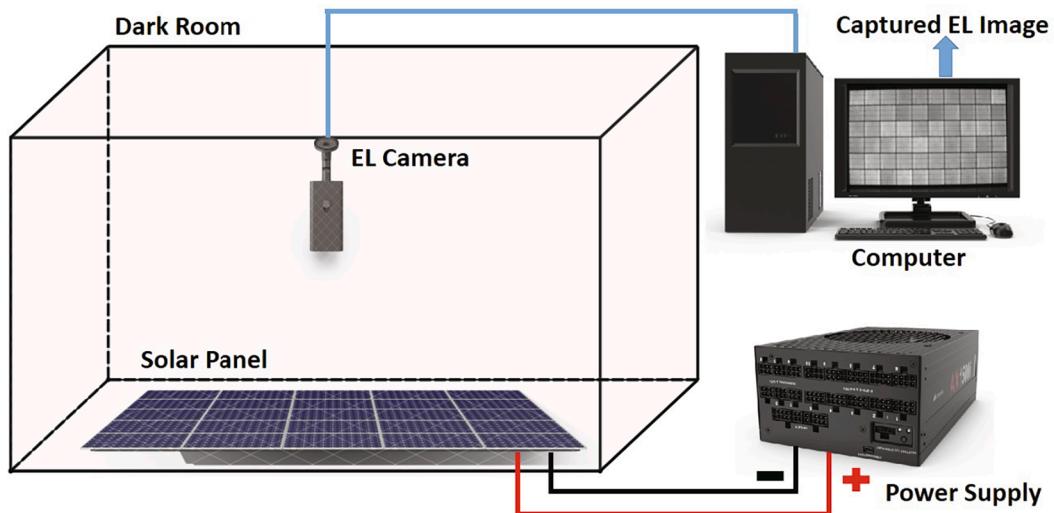


Fig. 1. Typical EL Imaging Setup.

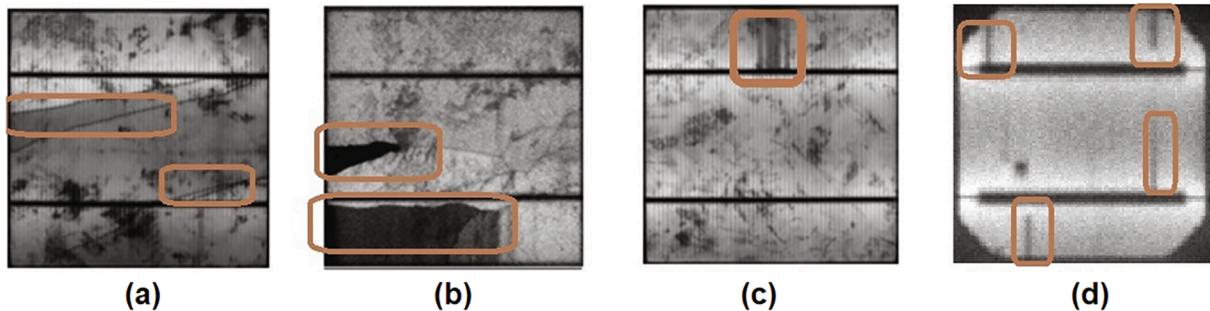


Fig. 2. EL Images of various Defected PV Cells (a) various crack patterns on the cell (b) partially broken cell (c) finger interruption (d) shunt fault (Köntges et al., 2014).

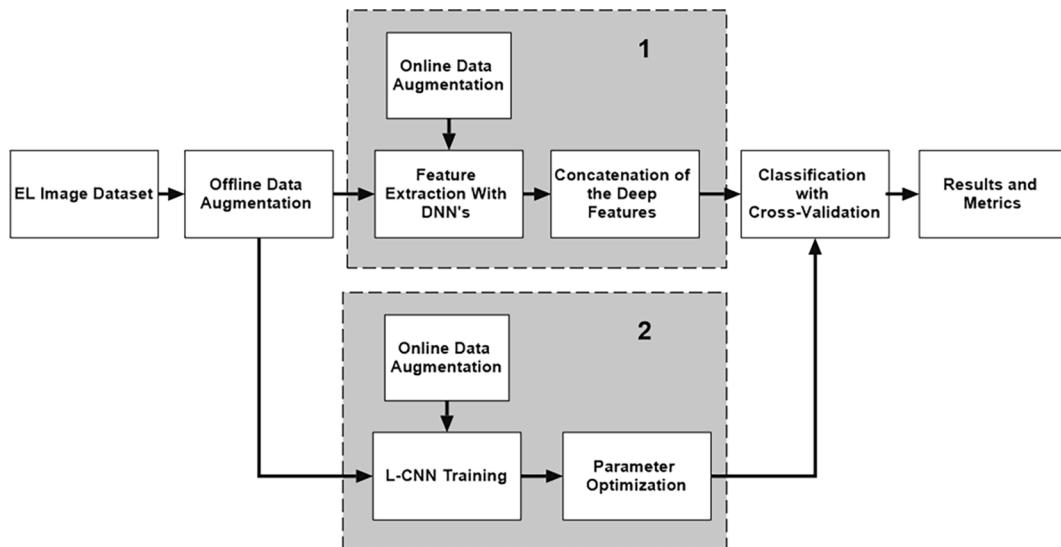
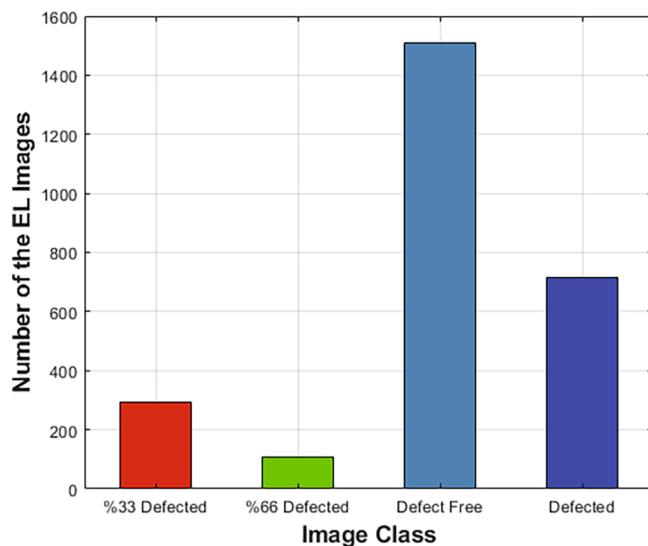


Fig. 3. Flowchart of the Proposed Methodology.

### 3.1.2. Data augmentation

Amount of data has a serious effect on learning process, especially for DNN training. In ELPV dataset, data amount is fairly low for DNN training. Therefore, data augmentation is performed both online and offline in the proposed method. In online data augmentation, random

reflection, translation and scaling are applied to the dataset in both horizontal and vertical directions. Since defect patterns in the EL images are very complicated and cracked cell images are difficult to learn for the detection system, translation ratio is limited to  $\pm 2$  pixels and scaling ratio is limited to  $\pm 2\%$ . The highest accuracy is obtained with these



**Fig. 4.** EL Dataset Distribution.

limitations. In addition, extra offline augmentation is performed on the data set before CNN training. In the offline data augmentation, all images were rotated  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$  and added to the original dataset to increase variations of defect patterns. The impact of data augmentation to the model performance is discussed in Section 4.2.

### 3.2. Feature extraction with CNN

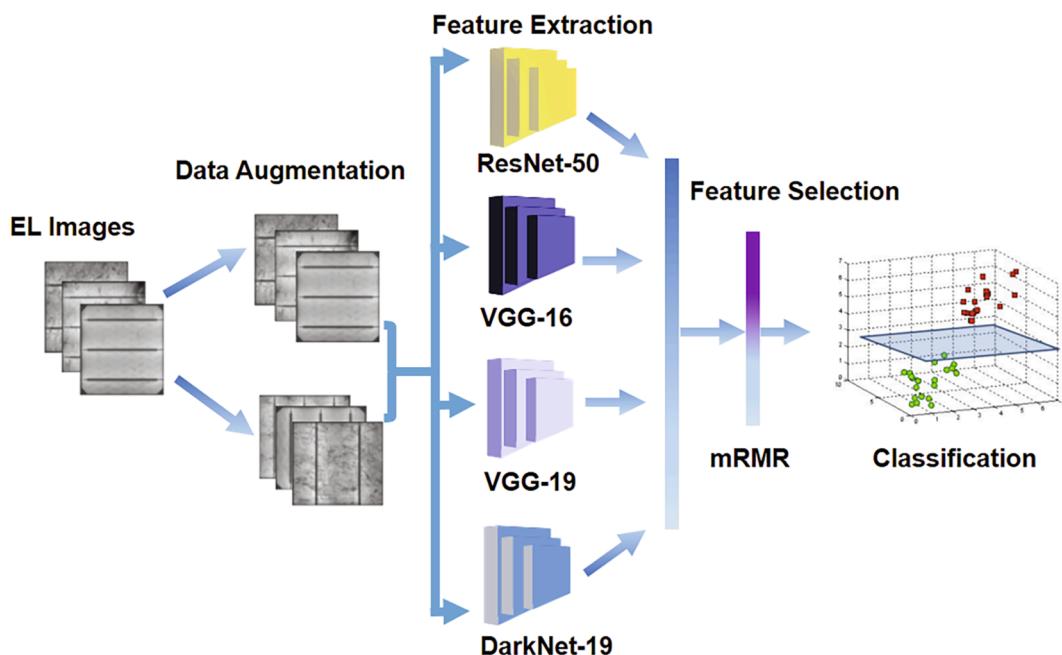
In this section, feature extraction stages are given in detail. In the first approach, image features are extracted with pre-trained Deep Neural Networks. A detailed flowchart of the proposed method is given in Fig. 5.

#### 3.2.1. Convolutional Neural networks

Convolutional Neural Networks (CNN) are a type of artificial neural networks and are often used in image recognition. In CNN, images are fed directly to the input and the data passes through different kernel

filters in sequential layers. Parameters are optimized with each iteration. This process is called training, and optimization methods such as back propagation and stochastic gradient descent are used for determining the most appropriate class of the given data (O'Shea & Nash, 2015). A typical CNN consists of convolutional layers, pooling, and fully connected layers in different numbers and combinations. There are also activation layers between the convolutional layers and input layers at the beginning and output layers at the end of the network. The number of layers and the filter kernel and filter size of the convolutional layers determine the number of the learnable parameters of network, hence the complexity. As an example, there are 138 million learnable parameters in VGG-16 architecture (Simonyan & Zisserman, 2015). Therefore, training of CNNs from scratch requires a lot of computational power. GPUs are often used for this job because of the high computational power demand (Yamashita, Nishio, Do, & Togashi, 2018). Unlike the human brain, DNNs could learn and generalize well only if there is sufficient labeled data. CNNs generally require very large amount of data and often need many epochs for training from scratch (Khan, Sohail, Zahoor, & Qureshi, 2020). This is one of the main reasons for using the pre-trained weights with the CNNs in this work. Due to the fact that the amount of data in ELPV dataset is very low and the computational resource demand is very high, feature extraction is performed by extracting the activation values from only certain layers of networks along with pre-trained weights.

In the proposed method, several CNN architectures were studied, and experiments were conducted using different pre-trained CNNs. As mentioned in the previous section, feature extraction is performed using the pre-trained deep networks of DarkNet-19, ResNet-50, VGG-16 and VGG-19. By default, all these networks are trained for 1000 object classes with 1 million images in ImageNet dataset. The main reasons for selection of these models are their high classification performances in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) and their simpler structures compared to the other larger models (Russakovsky et al., 2015). DarkNet is an open source high performance deep network architecture and is mostly known for YOLO real-time object recognition platform (Redmon, 0000; Redmon & Farhadi, 2017). The Architecture used in this study is called DarkNet-19 which is used in YOLOv2 and consists of 19 convolutional layers. In addition, DarkNet-19 has 64 layers total including normalization, pooling and activation layers along with 21 million parameters. Input size of the network is 256



**Fig. 5.** Feature Extraction Framework Representation.

$\times 256$  pixels. And  $3 \times 3$  kernel filters are used. DarkNet-19 is also known for its small and efficient architecture, and achieved successful results on the ImageNet and PASCAL VOC datasets (Abdelfalek, Ahmed, & Amine, 2019; Benali Amjoud & Amrouch, 2020). Another architecture called ResNet has residual network structure and is deeper unlike other architectures because of branches, but it has less complexity (He, Zhang, Ren, & Sun, 2016). ResNet architectures became popular with the winning of the ILSVRC 2015. Main advantage of the ResNet architecture is while it is deeper than linear network models such as AlexNet and VGG, it has less complexity, which made it suitable for the proposed work (Benali Amjoud & Amrouch, 2020; Fu & Rui, 2017). ResNet-50 network model used in the study is 50 layers deep and has 177 layers total along with 25.6 million parameters, including all residual layers. Input layer of the network is  $224 \times 224$  size and its main convolutional layers have  $3 \times 3$  kernel filters. VGG is another DNN architecture developed by Oxford University Visual Geometry Group in 2014 and it has a simple linear network structure using  $3 \times 3$  kernel filters like other networks. VGG architecture has gained attention with the winning of the ILSVRC 2014 and has been used in wide range of applications since then. However, VGG networks contain many fully connected layers, which resulted in much high number of parameters (Simonyan & Zisserman, 2015). This makes the usage of VGG architecture computationally intensive (Khan et al., 2020). The VGG-16 and VGG-19 networks used in the study have 16 and 19 layers, and they have 138 million and 144 million learnable parameters, respectively. Despite their complexity, VGG models are found suitable for the proposed method because only some layers are used and networks are not trained for multiple epochs in the proposed method. Fig. 6 shows the simple schematic representations of the networks and the layers used for feature extraction.

### 3.2.2. Feature extraction and concatenation

In this section, feature extraction and the usage of the DNNs as feature extractor are explained. Feature extraction is acquisition of information that defines shape and pattern from the data. In image recognition and machine vision, aim is to extract the most valuable

information from an image at the lowest size (Kumar & Bhatia, 2014). Useful information extracted from the image is transformed into feature vectors. Deep networks may be used for the classification or regression as well as for extracting the image features successfully. These extracted features are called deep features. In the first proposed method, activation values are extracted from different layers of the networks and feature vectors are obtained using the pre-trained DNNs ResNet-50, VGG-16, VGG-19 and DarkNet-19. The extracted features are concatenated then. For calculating the activations, all image data are fed to the CNN only once. Then, activation values are taken from the desired layer of the network and are transformed into feature vectors. This method has several advantages. First, features could be extracted directly from the raw image data and this eliminates the need for pre-processing. In addition, classification stage could be performed with the desired machine learning method using deep features. Moreover, deep feature based methods have far more efficient implementations than the conventional CNN training because the conventional CNN training requires multiple epochs for a reasonable accuracy, while this method requires only single pass on the training data (Kim, Kim, Noh, & Park, 2017).

In the CNNs, earlier layers tend to learn basic features such as lines, edges, corners or other shape information, while latter layers tend to learn more complex higher-level features such as object specific parts (Demir, Sengur, & Bajaj, 2020; Garcia-Gasulla et al., 2018; Namatévs, 2018). Since EL images are grayscale and all have similar quadratic cell shapes, there is no need to extract features from the higher layers as higher-level semantic information is not required. That is why higher convolutional layers are not used in the future extraction. As a result of different experiments, it has been observed that the features from the middle activation layers gave the best result in the EL image classification problem as can be seen in the Fig. 6.

Activation function layer, also known as non-linear detection layer, is used for detection of the linear activations through non-linear activation functions such as sigmoid, hyperbolic tangent or Rectified Linear Unit (ReLU). ReLU is the most used activation function in CNNs because it performs better and faster compared to other functions (Namatévs,

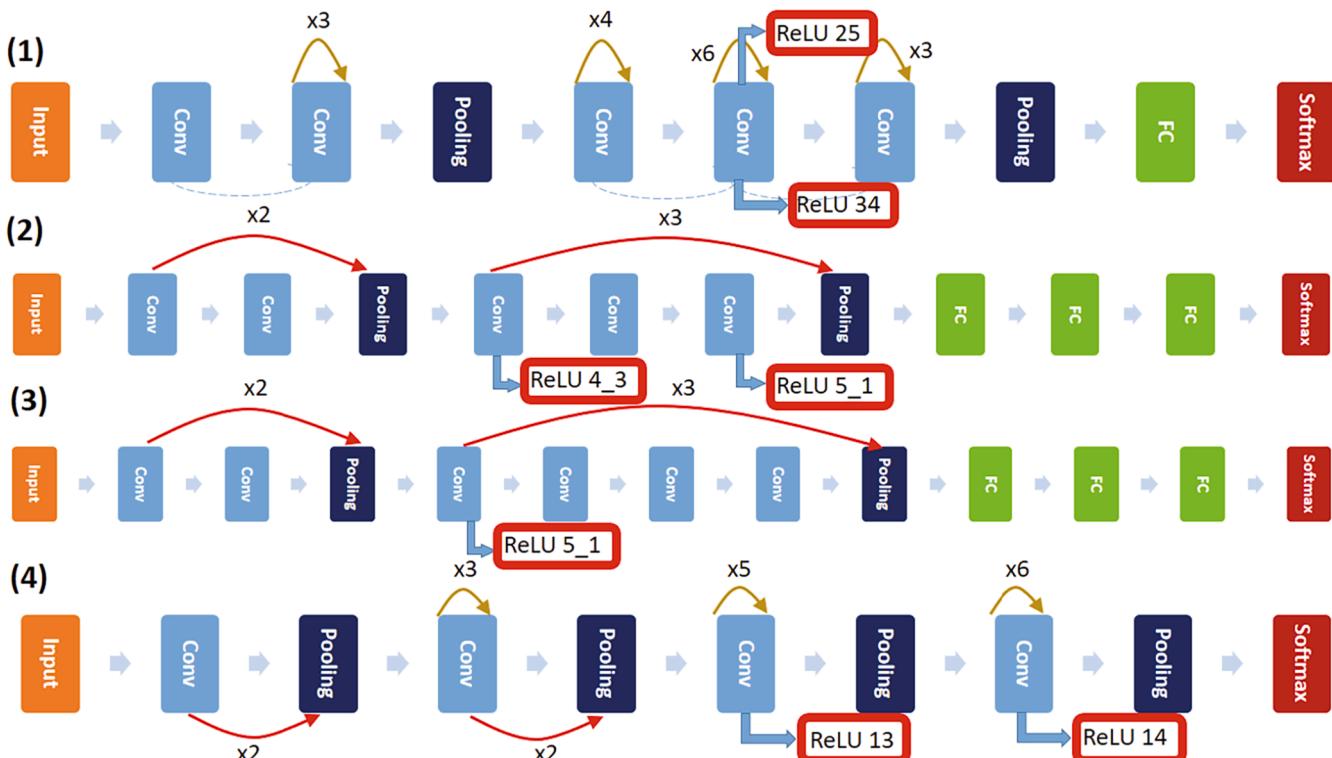


Fig. 6. Schematic of the Deep Neural Network Architectures (1) ResNet-50 (2) VGG-16 (3) VGG-19 (4) DarkNet-19.

2018; Nwankpa, Ijomah, Gachagan, & Marshall, 2018). ReLU operation applies a general thresholding to the input and values less than zero are set to zero. Hereby, positive values are rectified. Function of the ReLU is given as follows:

$$f(x) = \max(0, x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \quad (1)$$

The features are extracted from ReLU 25 and ReLU 34 layers in ResNet-50 network, ReLU 4\_3 and ReLU 5\_1 layers in VGG-16, ReLU 5\_1 layer in VGG-19 and ReLU 13 and ReLU 14 layers in DarkNet-19 network. In Fig. 6, the exact locations of the extracted feature layers on deep networks are marked with red squares. Extracted features are concatenated in 6 different structures; (1) VGG-16 (ReLU 4\_3 + ReLU 5\_1), (2) ResNet-50 (ReLU 25 + ReLU 34), (3) VGG-16 + VGG-19, (4) DarkNet-19 (ReLU 13 + ReLU 14), (5) VGG-16 + VGG-19 + ResNet and (6) DarkNet-19 + ResNet-50 + VGG-16 + VGG-19. The concatenated feature vectors and their dimensions are given in Table 1.

### 3.2.3. Feature selection

High-dimensional datasets could contain data which are repetitive, non-discriminating between classes, unrelated to the rest of the data, or do not contribute to the learning process. For this reason, it could be helpful to select more discriminative features. Feature selection is the process of determining attributes that have the best representation ability within a certain feature set. It is aimed to find the most distinctive features. There are generally three types of feature selection methods, which are filter-based, wrapper, and embedded methods (Khalid, Khalil, & Nasreen, 2014). Among these methods with the least number of repetitions, the minimum relationship (minimum Redundancy Maximum Relevance – mRMR) method is used in the proposed system. The mRMR method is a filter-based feature selection algorithm that selects the most relevant features sequentially based on a binary relationship. In this method repetitive features are minimized, and the most relevant features are determined by scoring mutual information. Peng, Long, and Ding (2005). Mutual information between two random variables given as  $X$  and  $Z$  is calculated as follows:

$$I(X, Z) = \sum_{i,j} P(X = x_i, Z = z_j) \log \frac{P(X = x_i, Z = z_j)}{P(X = x_i)P(Z = z_j)} \quad (2)$$

Here  $I(X, Z)$  represents the mutual information factor, while  $P(X, Z)$  is the probability distribution between variables  $X$  and  $Z$ , while  $P(X)$  and  $P(Z)$  are the probabilistic boundary distribution functions of  $X$  and  $Z$ , respectively. The purpose of the mRMR algorithm is to find the most relevant feature set  $F_S$ .  $S$  indicates an attribute set.  $V_s$  is the relevance coefficient and  $W_s$  is the redundancy coefficient. To find  $F_S$ ,  $V_s$  should be maximized, while  $W_s$  should be minimized.  $V_s$  and  $W_s$  are defined in Eqs. (3) and (4):

$$V_s = \frac{1}{|S|} \sum_{x \in S} I(x, y) \quad (3)$$

**Table 1**  
Extracted and Concatenated Feature Sets.

Feature vector	Used Conv. Net	Dimension
VGG-16 (ReLU 4_3 + ReLU 5_1)	VGG-16	1024
DarkNet-19 (ReLU 13 + ReLU 14)	DarkNet-19	1536
VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1)	VGG-16, VGG-19	1536
ResNet-50 (ReLU 25 + ReLU 34)	ResNet-50	2048
VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	ResNet-50, VGG-16, VGG-19	3584
DarkNet-19 (ReLU 13 + ReLU 14) + VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	DarkNet-19, ResNet-50, VGG-16, VGG-19	5120

$$W_s = \frac{1}{|S|^2} \sum_{x,z \in S} I(x, z) \quad (4)$$

Where the number of features of the  $S$  vector is represented as  $|S|$ ,  $y$  is the response variable at which the relationship will be found. Then, these two conditions are combined and put together as in Eq. (5):

$$\max(V_s - W_s) \quad (5)$$

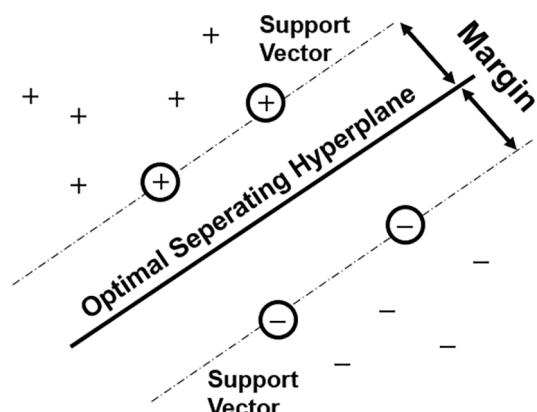
According to these formulas, all feature combinations must be calculated to find the optimal  $S$  vector, which means  $2^{|S|}$  calculations. Because of this, the mutual information factor is calculated as in Eq. (6) in order to reduce computational complexity.

$$\max(V_s/W_s) \quad (6)$$

### 3.3. Classification

In the study, KNN, DT, RF, NB, SVM methods are used as deep feature-based classifiers. K-Nearest Neighborhood is a distance-based learning algorithm, where measures the distance between instance and whole dataset. Test data is classified according to the closest distance in  $N$  neighbors' class. It is a lazy learning algorithm due to its instance-based structure. Selection of the parameter  $k$  is critical for the accuracy (Cunningham and Delany, 2020). Decision Tree is a predictive machine learning method, which aimed constructing an optimal tree from the given dataset. Tree consists of leaves and nodes, where conditions are represented as leaves and attributes are represented as nodes. Decision trees are constructed based on splitting criteria (Cycles, 2005). Random Forest is an ensemble learning method, based on constructing multiple unpruned decision trees and classifying the data based on the majority voting of the individual trees (Ali, Khan, Ahmad, and Maqsood, 2012). Naive Bayes is another classification method based on the Bayesian decision theorem. It is a probabilistic classification algorithm, which assume that class of given data is independent of the features. With its simple structure, Bayes classifiers perform well and fast, especially with large datasets (Langley, 1994).

SVM is a supervised learning method, which could be used in classification and regression problems. The purpose of SVM is to create hyperplanes which can separate data best. That means separating data at the largest interval margin, which is the maximum distance parallel to the hyperplane between two classes, and lines that intersect the closest data points are also called support vectors (Vapnik, 2000). A simple optimal hyperplane representation of the linear SVM is given in Fig. 7. Here, the plus signs represent one class, the minuses represent the other class, and the dashed lines represent support vectors, the thick line represents the optimal hyperplane. SVM is originally found as a linear classification algorithm and better classification performance could be achieved by using kernel trick, for datasets that cannot be separated



**Fig. 7.** Schematic of the SVM Classification (Vapnik, 2000).

linearly (Kancherla, Bodapati, & Veeranjaneyulu, 2019).

SVM is considered one of the most efficient and robust machine learning methods. Due to its extensive generalization capacity and discriminative ability, SVM is widely used in image classification and pattern recognition tasks. Main benefits of the SVMs are better generalization ability and better performance on limited data (Nurhanna & Othman, 2017). In addition, SVMs could be used in large variety of applications successfully through different kernels. Researchers obtained superior results with high dimensional image data using SVM (Demir, Bajaj, Ince, Taran, & Şengür, 2019; Jiao et al., 2016). Another main reason for using SVM classifiers is that SVM performs quite well in imbalanced datasets and is almost insensitive to the class imbalance (Prati, Batista, & Silva, 2015; He & Garcia, 2009). Despite its superior features, SVMs also have disadvantages such as right parameter selection necessity, computational load in large datasets and problems in handling multiclass (Cervantes, Garcia-Lamont, Rodríguez-Mazahua, & Lopez, 2020).

SVMs are essentially binary classification algorithm but training SVMs for multi-class problems is also possible. Mostly, one vs. one and one. vs. all methods are used in classification of multiclass data according to the principle of employing multiple binary classifiers. Let  $k$  be the number of classes. In one vs. one method, there are  $k(k-1)/2$  classifiers constructed and trained using all the binary combinations of  $k$  classes. Each classifier is trained on the binary class data. The first class is marked as positive and the second is marked as negative. When new data is fed to the classifiers, majority voted positive classifier is chosen as prediction. In one vs. all method,  $k$  classifiers are constructed and trained consecutively, where the first class is labeled as positive and the others are labeled as negative. When new data is fed, scores are obtained from all binary models and the class with maximum output is chosen as prediction (Nurhanna & Othman, 2017).

#### 3.4. Classification with L-CNN

As mentioned in the previous section, EL images contain neither semantic information nor color pixels, and earlier layers of CNNs tend to learn more basic features. Based on this idea, more shallow CNN models are constructed for the second approach. A series of experiments are conducted with different network architectures. For the purpose of simplicity, linear architectures are chosen. As a starting point, DarkNet and VGG - like architectures are formed, and parameters are optimized. Then, a lightweight CNN architecture (L-CNN) is proposed. The proposed network architecture contains 3 Convolutional layers with Max Pooling and ReLU layers. Also, after the last max pooling, a batch normalization layer is added. Complete schematic of the proposed model is given in Fig. 8.

Since effect of input size is important for the CNNs, several input sizes are considered. According to the experimental results, the best scores are obtained by setting up input layer to  $100 \times 100$  pixels with single channel, similar to the study (Akram et al., 2019). Filter parameters of the convolutional layer 1, 2 and 3 are chosen as 16, 32 and 64, respectively. This light single-channel architecture significantly reduced the training time while accelerating the accuracy. Complete hyperparameters for the L-CNN are given in Table 2.

#### 4. Experimental results

All studies are carried out in MATLAB environment. Experimental works and coding have been realized by using MATLAB R2020a with

**Table 2**  
Hyperparameters of the Proposed L-CNN.

Layer type	Paramaters
Image Input	[ $100 \times 100 \times 1$ ] – zero center normalization
Convolution 1	[ $3 \times 3$ ] Num. Filter :16 Stride (1 1)
Max Pooling + ReLU	Size : (55) Stride (1 1)
Convolution 2	[ $3 \times 3$ ] Num. Filter :32 Stride (1 1)
Max Pooling + ReLU	Size: (55) Stride (1 1)
Convolution 3	[ $3 \times 3$ ] Num. Filter :64 Stride (1 1)
Max Pooling + ReLU	Size: (55) Stride (1 1)
Batch Normalization	Mean [ $1 \times 1 \times 64$ ] epsilon $1e^{-5}$
Fully Connected	Input: [ $2 \times 64 \times 10^4$ ]
Classification Output	Output: 2/4 Class

Statistical and Machine Learning Toolbox and Deep Learning Toolbox. For the DFB models, DNNs are trained on GPU, using a laptop computer with Intel © Core-i5 4200U processor, 8 GB of RAM and Nvidia GT-740 M graphics unit. For the L-CNN training, a workstation with Intel © Xeon E-2224G processor, 16 GB of RAM and Nvidia Quadro P630 graphic unit is used. In the feature extraction stage, all DNNs are trained with Stochastic Gradient Descent (SGD) optimizer, single epoch and learning rate of  $n = 10^{-4}$ . Minibatch parameters were determined as 8 for VGG-16 and VGG-19, 16 for DarkNet-19 and 32 for ResNet-50. For the L-CNN training, SGD optimizer with initial learning rate of  $n = 10^{-4}$  is used. Minibatch value is selected as 32 and the network is trained for 120 epochs. In the classification stage; for KNN,  $k = 3$  and Euclidean distance is determined. For DT, maximum split is set to 100 and Gini index is chosen as split criteria. For RF, bagging is used as ensemble method, number of learners is  $l = 50$ , maximum split is set to 8396. For NB, Gaussian kernel is selected. For SVM, 3rd order polynomial (cubic) SVM kernel was used in all studies and parameters was determined manually as  $C = 900$ . Also, one vs. all is selected as multiclass method for 4-class experiments. All results are validated with cross validation on the training set. The dataset is divided into 80% for training and 20% for testing for the all experiments. However, Deitsch et al. (2019) performed classification for original 4-class dataset using 75% of the data for training and 25% for testing. For a better comparison, the dataset ratios of both 80% to 20% and 75% to 25% are used for 4-class experiments.

#### 4.1. Performance metrics

Accuracy, Precision, Recall, F-Score, Receiver Operating Characteristic (ROC) and Area Under Curve (AUC) metrics are calculated for each model as performance measurements. Prediction results are qualified as True Positive ( $TP$ ), True Negative ( $TN$ ), False Positive ( $FP$ ) and False Negative ( $FN$ ).  $TP$  represents a correct prediction of completely defect-free PV cell and  $TN$  represents a correct prediction of defected PV cell, while  $FP$  represents an incorrect prediction of defect-free PV cell and  $FN$  represents an incorrect prediction of defected PV cell. ROC is a graphical classifier assessment method based on the  $TP$  rate and  $FP$  rate. AUC represents the two-dimensional measured area under the ROC curve. Accuracy represents the ratio of correct predictions to total predictions. Precision represents the ratio of true positive predictions to total positive predictions. Recall represents the ratio of true positive predictions to actual positives. F-Score is the harmonic mean of precision and recall (Tharwat, 2020). These metrics are given in Formula (7)–(10).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$



**Fig. 8.** Proposed L-CNN Architecture.

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$F - Score = 2x \frac{Precision \times Recall}{Precision + Recall} \quad (10)$$

#### 4.2. Impact of data augmentation

As mentioned in the previous section, online and offline data augmentation is applied to the dataset together in the feature extraction stage. To measure the effect of data augmentation, experiments were carried out using DFB models with different data augmentation settings. In these experiments, only extracted features from ResNet-50 are used, the dataset is divided into 80% for training and 20% for testing and SVM classification is performed with the original 4 - class only. Other variables are kept the same as in the all experiments. According to the augmentation modes, testing accuracies are given in the Table 3. As can be seen from Table 3, the highest accuracy is achieved when online and offline data augmentation are used together. Therefore, features are extracted with this structure in all the following experiments.

#### 4.3. SVM classification results

Experimental results are given for 2-class and 4-class predictions using SVMs on the 20% test data. For 2-class experiments, Possibly normal cells are re-labeled as Non-defected and Possibly defected cells are re-labeled as Defected for the exact comparison. In the experiments, extracted features from DNNs were used both individually and together in many combinations for classification. All results were verified with cross-validation and accuracy, precision, recall and F-Scores were given in Tables 4 and 5. Highest scores were marked as bold in the tables.

In the Tables 4 and 5, features extracted from VGG-16, DarkNet-19 and ResNet-50 were given on the top, and the combined features were given at the bottom. The length of the feature vectors become 1024, 1536, 2048, 3584 and 5120 depending on the number of layers combined. The best results were obtained with the combined feature vectors of 5120 for both 4-class and 2-class datasets. The accuracy rates were achieved as 88.90% for 4-class and 93.62% for 2-class. Features extracted from the VGG-16 network provided the worst result among them. It is worth noting that ResNet-50 extracted features alone performed almost better than the combined features of ResNet-50 + VGG-16 + VGG-19.

From the concatenated features of 3584 and 5120, the best 2000 and 2500 features were chosen with the mRMR algorithm. As seen in Fig. 9, the highest accuracy is achieved with the selected features of 2000 and 2500 for 4-class and 2-class datasets.

Although the accuracy increases with more features, there is no significant performance increase after 2500 features. The effect of feature size on the classification performance is given in Fig. 9. After the feature selection, selected features are classified. SVM classification scores of the proposed feature sets are given in Tables 6 and 7 in comparison with the concatenated features.

When the classification results with the selected features were examined, one can see that selected features perform better in all scores and in all experiments as shown in Tables 6 and 7. These results indicate

**Table 3**  
ResNet-50 Classification Results with Different Data Augmentation Modes.

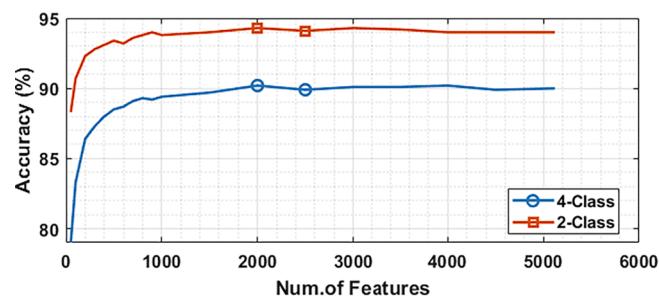
Experiment No.	Augmentation mode	Accuracy (%)
1	No Augmentation	76.76
2	Online Augmentation Only	77.14
3	Offline Augmentation Only	80.09
4	Online + Offline Augmentation	<b>80.71</b>

**Table 4**  
Concatenated Features SVM Classification Results (4 - Class).

Obtained feature vectors	Dimension	Accuracy (%)	Precision (%)	Recall (%)	F-Score
VGG-16 (ReLU 4_3 + ReLU 5_1)	1024	81.85	79.65	91.77	85.28
DarkNet-19 (ReLU 13 + ReLU 14)	1536	83.18	80.03	93.39	86.19
VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1)	1536	<b>82.28</b>	79.55	92.62	85.59
ResNet-50 (ReLU 25 + ReLU 34)	2048	85.18	82.84	93.82	87.99
VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	3584	84.47	81.88	93.49	87.31
DarkNet-19 (ReLU 13 + ReLU 14) + VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	5120	<b>88.90</b>	<b>86.03</b>	<b>96.55</b>	<b>90.99</b>

**Table 5**  
Concatenated Features SVM Classification Results (2 - Class).

Obtained feature vectors	Dimension	Accuracy (%)	Precision (%)	Recall (%)	F-Score
VGG-16 (ReLU 4_3 + ReLU 5_1)	1024	88.76	91.24	92.62	91.92
DarkNet-19 (ReLU 13 + ReLU 14)	1536	91.09	92.22	94.99	93.58
VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1)	1536	89.61	90.98	94.36	92.64
ResNet-50 (ReLU 25 + ReLU 34)	2048	91.28	92.24	95.16	93.68
VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	3584	91.23	91.66	95.78	93.67
DarkNet-19 (ReLU 13 + ReLU 14) + VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	5120	<b>93.62</b>	<b>93.99</b>	<b>97.07</b>	<b>95.50</b>



**Fig. 9.** SVM Classification Accuracy of 4 - class and 2 - class Datasets with Feature Dimension.

**Table 6**

SVM Classification Results with Selected Features (4 - Class).

Obtained feature vectors	Dimension	Accuracy (%)	Precision (%)	Recall (%)	F-Score
VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	3584	84.47	81.88	93.49	87.31
DarkNet-19 (ReLU 13 + ReLU 14) + VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	5120	88.90	86.03	96.55	90.99
<b>Selected 2000</b> Features from VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	2000	89.76	87.10	96.23	91.44
<b>Selected 2500</b> Features from VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	2500	88.85	85.50	96.42	90.63
<b>Selected 2000</b> Features from Darknet-19 (ReLU 13 + ReLU 14) + VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	2000	90.57	88.12	96.99	92.34
<b>Selected 2500</b> Features from DarkNet-19 (ReLU 13 + ReLU 14) + VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	2500	89.61	86.55	96.73	91.36

efficiency of the feature selection with the mRMR algorithm. Highest scores were obtained with the selected 2000 features among 5120 features are 90.57% and 94.52% accuracy for 2-class and 4-class dataset, respectively. The second-best results were achieved by using the selected 2000 features from 3584 features for 4-class. For 2-class experiments, the performances with the selected 2000 and selected 2500 features were close. As a result, the use of the selected features outperformed the full features with fewer attributes. In Fig. 10, t-SNE (t-Distributed Stochastic Neighbor Embedding) maps of the dataset with selected 100, 1000 and 2000 features are given. It could be observed that features with distinct classes can be separated better with the increasing dimensions.

#### 4.4. L-CNN classification results

Proposed L-CNN was trained for both 4-class and 2-class datasets. Training and validation curves were given in Fig. 11 for 2-class and 4-

**Table 7**

SVM Classification Results with Selected Features (2 - Class).

Obtained feature vectors	Dimension	Accuracy (%)	Precision (%)	Recall (%)	F-Score
VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	3584	91.23	91.66	95.78	93.67
DarkNet-19 (ReLU 13 + ReLU 14) + VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	5120	93.62	93.99	97.07	95.50
<b>Selected 2000</b> Features from VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	2000	94.04	93.90	<b>97.54</b>	95.68
<b>Selected 2500</b> Features from VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	2500	94.00	<b>94.81</b>	96.74	95.76
<b>Selected 2000</b> Features from DarkNet-19 (ReLU 13 + ReLU 14) + VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	2000	94.52	94.79	97.36	<b>96.06</b>
<b>Selected 2500</b> Features from DarkNet-19 (ReLU 13 + ReLU 14) + VGG-16 (ReLU 4_3 + ReLU 5_1) + VGG-19 (ReLU 5_1) + ResNet-50 (ReLU 25 + ReLU 34)	2500	93.76	94.12	97.04	95.55

class models. As seen from the graphs, both trained CNN models plotted uneven curves in the first epochs, but it converged after 60 epochs and validation line became flattened. Average training time took around 135 min for 120 epochs. Performance metrics were calculated for predictions on the 20% test data. Results were compared with the DFB models and comparison results were given in the Table 8.

L-CNN results were compared to only the best performing DFB-SVM models with the selected 2000 features. Metrics such as accuracy, precision, recall, F-Scores, ROC curves and AUCs were calculated for each model. When the results are examined, it can be inferred that proposed DFB-SVM models outperformed L-CNN models for both 2-class and 4-class datasets.

It is worth noting that accuracy differences of L-CNN and DFB-SVM models for the 2-class and 4-class were almost similar. Both methods performed well in terms of ROC. The Area Under Curve(AUC) values of L-CNN and DFB-SVM classifiers were similar for both 2-class and 4-class datasets. However, DFB-SVM models obtained better AUC for both 2-

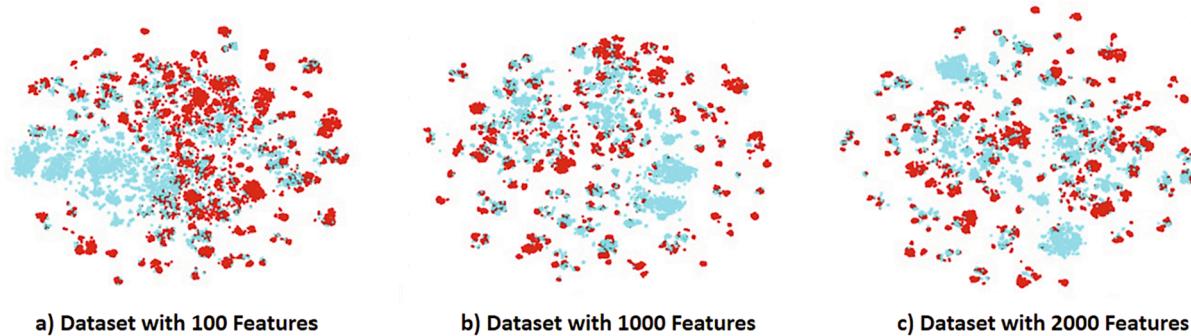


Fig. 10. t-SNE maps of the 2 - class EL Dataset with Selected Features.

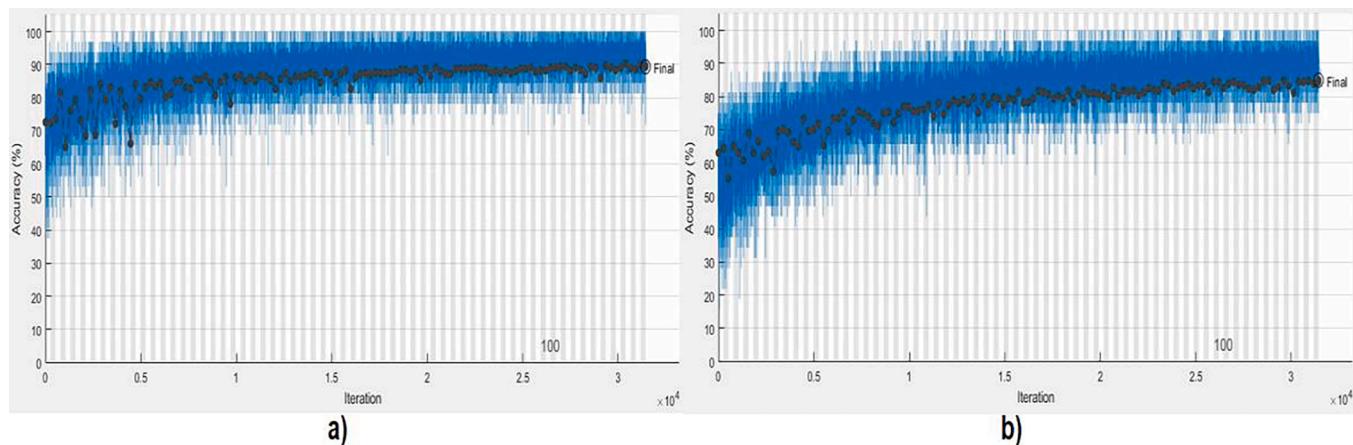


Fig. 11. Accuracy Curves for the L-CNN a) 2 Class Dataset b) 4 Class Dataset.

**Table 8**  
Performance Comparison of L-CNN and DFB-SVM Models.

Dataset	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-Score	AUC
ELPV- 4 Class	L-CNN	84.95	84.98	89.64	87.25	93.53
	DFB-SVM	<b>90.57</b>	<b>90.60</b>	<b>90.61</b>	<b>90.62</b>	<b>97.00</b>
ELPV- 2 Class	L-CNN	89.33	90.44	95.42	92.86	93.36
	DFB-SVM	<b>94.52</b>	<b>94.79</b>	<b>97.36</b>	<b>96.06</b>	<b>97.86</b>

class and 4-class datasets. Confusion matrices are given in the Figs. 12 and 13 for the proposed L-CNN and DFB-SVM models. In the confusion matrices, vertical axes represent the predictions and horizontal axes represent the target classes (ground truth). ROC curves of the DFB-SVM and L-CNN models were given in Fig. 14.

For the 4-class confusion matrices, it could be seen that both methods were performed close with the Non-defected (%0) and Possibly normal (%33) classes. However, with the Possibly defected (%66) and 100% Defected (%100) classes, DFB-SVM models clearly outperformed the L-CNN models. For the 2-class confusion matrices, DFB-SVM clearly outperformed L-CNN for the both Defected (%100) and Non-defected (%0) classes. Another important point is that both of the models did not perform on Possibly normal (%33) and Possibly defected (%66) classes

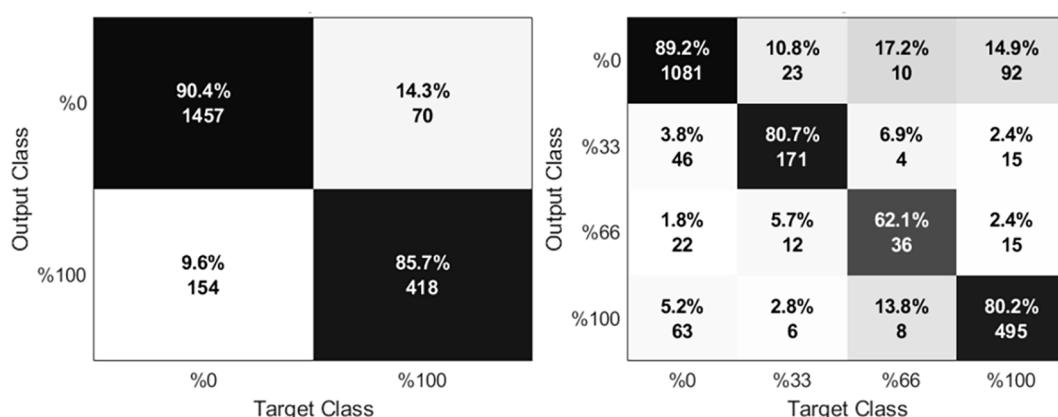


Fig. 12. Confusion matrices for the L-CNN Classifiers.

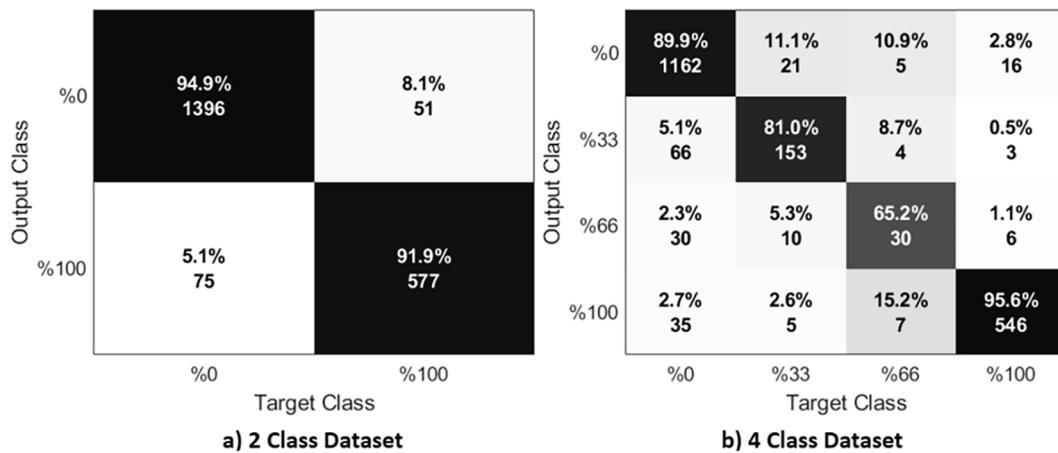


Fig. 13. Confusion matrices for the DFB-SVM Classifiers.

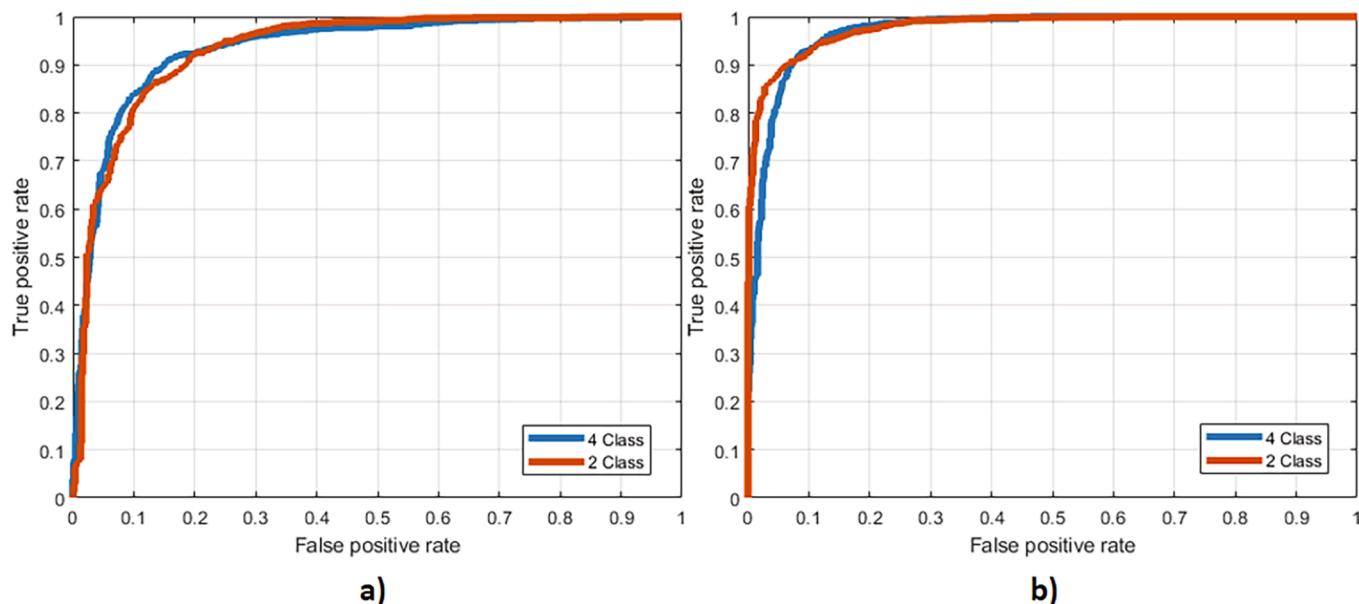


Fig. 14. ROC Curves of the proposed models. a) L-CNN Models b) DFB-SVM Models.

as well as on the other classes. This is most likely caused by insufficient training data for these classes.

#### 4.5. Comparison with other Machine learning methods

In this section, selected deep features were classified with the other machine learning methods such as Decision Tree, Random Forest, K-Nearest Neighborhood and Naïve Bayes. Experiments were carried out only on the best performing selected 2000 and 2500 features for both 2-class and 4-class datasets. Results were given in the Tables 9 and 10.

As seen from tables, DFB-SVM models performed best among all the classifiers for both 2-class and 4-class datasets. Second best scores were obtained with KNN models. NB models performed worst in both 2 and 4-class datasets. It is worth noting that DFB-SVM models were generally performed better with selected 2000 features while other models are generally performed better with 2500 features. Another important point is that DFB-SVM models performed similarly well on both 2-class and 4-class datasets in terms of metrics while other methods performed with a larger difference.

**Table 9**  
Comparison with Different Machine Learning Algorithms (4-class).

Obtained feature vectors	Method	Accuracy (%)	Precision (%)	Recall (%)	F-Score
Selected 2000 Features	DFB-SVM	90.57	88.12	96.99	92.34
Selected 2500 Features		88.85	85.50	96.42	90.63
Selected 2000 Features	KNN	82.66	78.15	96.84	86.50
Selected 2500 Features		83.37	79.01	96.86	87.03
Selected 2000 Features	RF	80.56	75.32	98.34	85.30
Selected 2500 Features		79.56	75.03	96.69	84.50
Selected 2000 Features	DT	73.70	71.06	91.36	79.94
Selected 2500 Features		74.04	71.53	91.23	80.19
Selected 2000 Features	NB	69.89	73.33	74.67	73.99
Selected 2500 Features		72.27	74.86	78.08	76.44

**Table 10**

Comparison with Different Machine Learning Algorithms (2-class).

Obtained feature vectors	Method	Accuracy (%)	Precision (%)	Recall (%)	F-Score
Selected 2000 Features	DFB-SVM	94.52	94.79	97.36	96.06
Selected 2500 Features		93.76	94.12	97.04	95.55
Selected 2000 Features	KNN	90.42	89.50	97.50	93.33
Selected 2500 Features		90.81	89.86	97.81	93.67
Selected 2000 Features	RF	89.23	88.14	97.43	92.56
Selected 2500 Features		88.99	88.38	96.92	92.45
Selected 2000 Features	DT	86.71	87.59	93.97	90.67
Selected 2500 Features		86.71	87.58	94.24	90.79
Selected 2000 Features	NB	86.47	88.55	92.23	90.35
Selected 2500 Features		85.23	89.11	89.72	89.41

#### 4.6. Comparison with previous works

In this section, experimental results were compared with previous state-of-art results. Deitsch et al. (2019) performed classification for original 4-class dataset and they divided 25% of the data as a testing and 75% as training. However, Akram et al. (2019) divided dataset as 80% training 20% testing. For this reason, proposed best performing DFB-SVM models with the 2000 and 2500 features along with 4-class, are divided again as 75% training and 25% testing, while other parameters were fixed, then results were validated with cross-validation. In addition, L-CNN model is trained with the 75% training and 25% validation data, while other hyperparameters kept same as other experiments. Comparison of the results is given with the proposed methods results is given in Table 11 for 2-class and in Table 12 for 4-class dataset.

As seen in Table 11, the proposed DFB-SVM model achieved 94.52% of test accuracy with the selected 2000 features while Akram et al. achieved 93.02% of test accuracy in the experiments on 2-class dataset. Proposed L-CNN model performed worst in terms of accuracy. However, L-CNN achieved better recall and F-Score values than the previous work.

In the experiments with 4-class dataset, Deitsch et al. achieved 82.44% of accuracy and 82.52 of F-Score using SVM in their studies. Besides, they have achieved 88.42% of accuracy and 88.89 of F-Score with CNN.

Proposed DFB-SVM model reached 89.63% of accuracy and 91.55 of F-score with the selected 2500 features in the experiments, which was carried out under the same conditions as 75% training and 25% testing data. While proposed L-CNN model performed close to the SVM model of Deitsch et al., it obtained a better F-Score. It can be clearly seen that the proposed DFB-SVM models performed best among the previous works in terms of accuracy and all other metrics.

**Table 11**

Comparison of the 2 – Class Dataset Results (80% training – 20% test).

Method name	Classification method	Accuracy (%)	Precision (%)	Recall (%)	F-Score
Akram Et AL	CNN	93.02	93	93	92.49
Proposed L-CNN	CNN	89.33	90.44	95.42	92.86
Proposed DFB-SVM (Selected 2000 From 5120 Features)	SVM	94.52	94.79	97.36	96.06

**Table 12**

Comparison of the 4 – Class Dataset Results (75% training – 25% test).

Method name	Classification method	Accuracy (%)	Precision (%)	Recall (%)	F-Score
Deitsch et al.	SVM	82.44	–	–	82.52
	CNN	88.42	–	–	88.89
Proposed L-CNN	CNN	82.58	80.71	91.58	85.80
Proposed DFB-SVM (Selected 2500 From 5120 Features)	SVM	89.63	87.01	96.59	91.55

#### 5. Discussions

Classification of solar cell defects in EL image is a challenging task in general because solar cells contain crystal grain boundaries caused by the internal silicon structure. This makes distinguishing defective and normal areas harder. Another important fact is that there is not a sufficiently large dataset containing both monocrystalline and polycrystalline cell images. In the proposed work, extensive experimentation is carried out on a series of different DNNs and feature sets. The results indicated that the proposed models classified defective and normal solar cells accurately. However, when the confusion matrices of 4-class models are examined, it can be seen that Possibly normal (33% defected) and Possibly defected (66% defected) cells were not classified as well as the other cells (i.e. Non-defected and Defected). This is the main drawback of this work caused by the lack of enough EL image data and the extreme class imbalances. These issues are addressed using data augmentation methods at a certain level. For further improvement, more data for these classes might be collected, which could be a subject for future work.

Both proposed methods are well suited for the other image classification and defect detection applications with their high performance and light structure. Proposed L-CNN performed worse than DFB-SVMs in general. However, L-CNN obtained reasonable scores with its simple structure and has the advantage of fast learning as it generalizes well under 100 epochs. In the DFB-SVMs feature extraction stage, it is critical to extract features from the appropriate level of layer. For the ELPV dataset, the use of low and mid-level CNN layers resulted in better performance. Therefore, the proposed method may perform better on the problems with detailed images with non-semantic patterns. In DFB-SVMs, accuracy is accelerated with the concatenation of different CNN feature sets and feature selection. In addition, training stages of the proposed methods are more efficient than conventional DNN training because they do not require long training time and high-end graphics card. The proposed models perform well on the imbalanced datasets using SVM for classification, and DFB-SVM framework can classify a single cell image within a very short time (approx. 13 ms) on a standard portable computer. The proposed system classifies a 60-cell solar module less than a second. Therefore, the use of the proposed methods on the automatic PV module test systems for the production facilities can be considered. However, more labeled image data should be provided for more accurate classifier system. In addition, it would be important to capture the module EL images on the site under similar conditions with ELPV dataset, which would also help to augment the EL image dataset.

#### 6. Conclusions

In this study, a novel automatic defect detection and classification framework for solar cell EL images is proposed. Feature extraction, selection and classification of defective solar cells is performed using a public dataset consisting of both monocrystalline and polycrystalline solar cell EL images. Compared to previous works, higher performed

models are obtained by using DNNs and ML methods together and a general efficient classification framework is proposed. In the first method, images in the dataset are fed directly to the DNNs and feature vectors are obtained with the activations taken from middle activation layers of these networks. Extracted features concatenated in different combinations were classified with different ML methods. By combining the extracted features, new and larger size feature vectors were obtained and the ones with the best representation capabilities from these features were selected with the mRMR algorithm. Obtained feature vectors were classified with Decision Tree, Random Forest, K-Nearest Neighborhood, Naïve Bayes and Support Vector Machines. In the second method, a novel lightweight fast learning CNN structure called L-CNN is proposed, trained from scratch and optimized. As seen from the experimental results, DFB-SVM models performed the best, and the selected feature sets outperformed the full feature sets, while L-CNN obtained competitive results. The highest accuracy achieved by the proposed models are 90.57% for 4-class (i.e. Non-defected, Possibly normal, Possibly defected and Defected) and 94.52% for 2-class (i.e. functional, defective), which are higher than previous studies. Results are compared with the previous studies and verified by cross-validation. Accuracy, precision, recall and F-Score values are calculated. ROC curves, AUC values and confusion matrices are given in detail.

In conclusion, DFB method improves performance on EL image classification problem. Additionally, combining SVM classifier with DBF for both 2-class and 4-class EL image classification problems gives better results than Decision Tree, Random Forest, K-Nearest Neighborhood, Naïve Bayes classification methods do. Considering the class distributions of the samples obtained by the DFB method, the SVM classification method performs better on imbalanced data sets.

In future work, a novel end-to-end CNN model may be constructed for the prediction of defect types such as crack, finger interruption or busbar fault etc. with the sufficient labeled data. In addition, additional methods would be investigated to eliminate the disadvantages of imbalanced class distribution in the EL image data set.

#### CRediT authorship contribution statement

**Mustafa Yusuf Demirci:** Conceptualization, Methodology, Investigation, Software, Visualization, Writing - original draft. **Nurettin Beşli:** Conceptualization, Formal analysis, Supervision, Project administration, Resources. **Abdulkadir Gümüşü:** Conceptualization, Formal analysis, Supervision, Project administration, Resources.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### References

- Abdelmalek, B., Ahmed, K., & Amine, T. M. (2019). A survey on lightweight CNN-based object detection algorithms for platforms with limited computational resources. *International Journal of Informatics and Applied Mathematics*, 2(2), 28–44.
- Akram, M. W., Li, G., Jin, Y., Chen, X., Zhu, C., Zhao, X., & Ahmad, A. (2019). CNN based automatic detection of photovoltaic cell defects in electroluminescence images. *Energy*, 189. <https://doi.org/10.1016/j.energy.2019.116319>
- Ali, J., Khan, R., Ahmad, N., & Maqsood, I. (2012). Random Forests and Decision Trees. *International Journal of Computer Science Issues*, 9(5), 272–278.
- Anwar, S. A., & Abdullah, M. Z. (2013). Micro-crack detection of multicrystalline solar cells featuring shape analysis and support vector machines. *Proceedings - 2012 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2012*, 143–148. 10.1109/ICCSCE.2012.6487131.
- Balzategui, J., Eciolaza, L., & Arana-Arexolaleiba, N. (2020). Defect detection on Polycrystalline solar cells using Electroluminescence and Fully Convolutional Neural Networks. In 2020 IEEE/SICE International Symposium on System Integration (SII) (pp. 949–953). IEEE. 10.1109/SII46433.2020.9026211.
- Bartler, A., Mauch, L., Yang, B., Reuter, M., & Stoicescu, L. (2018). Automated detection of solar cell defects with deep learning. *European Signal Processing Conference, 2018-Septe*, 2035–2039. 10.23919/EUSIPCO.2018.8553025.
- Benali Amjoud, A., & Amrouch, M. (2020). Convolutional Neural Networks Backbones for Object Detection. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 12119 LNCS, pp. 282–289). Springer International Publishing. 10.1007/978-3-030-51935-3\_30.
- Buerhop-Lutz, C., Deitsch, S., Maier, A., Gallwitz, F., & Brabec, C. J. (2018). A Benchmark for Visual Identification of Defective Solar Cells in Electroluminescence Imagery. 35th European PV Solar Energy Conference and Exhibition, 1287–1289.
- Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408(xxxx), 189–215. <https://doi.org/10.1016/j.neucom.2019.10.118>
- Cibuk, M., Budak, U., Guo, Y., Cevdet Ince, M., & Sengur, A. (2019). Efficient deep features selections and classification for flower species recognition. *Measurement: Journal of the International Measurement Confederation*, 137, 7–13. <https://doi.org/10.1016/j.measurement.2019.01.041>
- Cunningham, P., & Delany, S. J. (2020). k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples). *Multiple Classifier Systems*, (April 2007), 1–17. UCD-CSI-2007-4.
- Cycles, S. (2005). Data Mining and Knowledge Discovery Handbook. (O. Maimon & L. Rokach, Eds.), Cycle (Vol. 1897). New York: Springer-Verlag. 10.1007/b107408.
- Deitsch, S., Buerhop-Lutz, C., Sovetkin, E., Steland, A., Maier, A., Gallwitz, F., & Riess, C. (2018). Segmentation of Photovoltaic Module Cells in Electroluminescence Images. Retrieved from <http://arxiv.org/abs/1806.06530>.
- 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(February), 455–468. <https://doi.org/10.1016/j.solener.2019.02.067>
- Demir, F., Bajaj, V., Ince, M. C., Taran, S., & Şengür, A. (2019). Surface EMG signals and deep transfer learning-based physical action classification. *Neural Computing and Applications*, 31(12), 8455–8462. <https://doi.org/10.1007/s00521-019-04553-7>
- Demir, F., Sengür, A., & Bajaj, V. (2020). Convolutional neural networks based efficient approach for classification of lung diseases. *Health Information Science and Systems*, 8 (1), 4. <https://doi.org/10.1007/s13755-019-0091-3>
- Demir, F., Şengür, A., & Çavaş, M. (2019). Heart Sounds Classification With Deep Features and Support Vector Machines. 2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018, (Dvm). 10.1109/IDAP.2018.8620733.
- Demirci, M. Y., Beşli, N., & Gümüşü, A. (2019). Defective PV Cell Detection Using Deep Transfer Learning and EL Imaging. In International Conference on Data Science, Machine Learning and Statistics 2019 (DMS-2019).
- Evans, R. (2014). Interpreting module EL images for quality control. Proceedings of the 52nd Annual Conference, Australian Solar Energy Society, (May 2014). Retrieved from [http://solar.org.au/papers/14papers/%23117\\_final.pdf](http://solar.org.au/papers/14papers/%23117_final.pdf).
- Fu, J., & Rui, Y. (2017). Advances in deep learning approaches for image tagging. *APSIPA Transactions on Signal and Information Processing*, 6, e11. 10.1017/ATSP.2017.12.
- Fuyuki, T., Kondo, H., Kaji, Y., Yamazaki, T., Takahashi, Y., & Uraoka, Y. (2005). One shot mapping of minority carrier diffusion length in polycrystalline silicon solar cells using electroluminescence, 1343–1345. 10.1109/pvsc.2005.1488390.
- Garcia-Gasulla, D., Parés, F., Vilalta, A., Moreno, J., Ayguadé, E., Labarta, J., & Suzumura, T. (2018). On the behavior of convolutional nets for feature extraction. *Journal of Artificial Intelligence Research*, 61, 563–592. <https://doi.org/10.1613/jair.5756>
- Haibo He, & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284. 10.1109/TKDE.2008.239.
- Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), 1226–1238. <https://doi.org/10.1109/TPAMI.2005.159>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 770–778. 10.1109/CVPR.2016.90.
- Hou, W., Wei, Y., Jin, Y., & Zhu, C. (2019). Deep features based on a DCNN model for classifying imbalanced weld flaw types. *Measurement: Journal of the International Measurement Confederation*, 131, 482–489. <https://doi.org/10.1016/j.measurement.2018.09.011>
- Jahn, U., Herz, M., Köntges, M., Parlevliet, D., Paggi, M., Tsanakas, I., ... Tanahashi, T. (2018). Review on Infrared and Electroluminescence Imaging for PV Field Applications.
- Jiao, Z., Gao, X., Wang, Y., & Li, J. (2016). A deep feature based framework for breast masses classification. *Neurocomputing*, 197, 221–231. <https://doi.org/10.1016/j.neucom.2016.02.060>
- Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0192-5>
- Kancherla, D., Bodapati, J. D., & Veeranjaneyulu, N. (2019). Effect of different kernels on the performance of an SVM based classification. *International Journal of Recent Technology and Engineering*, 7(5), 1–6.
- Karimi, A. M., Fada, J. S., Hossain, M. A., Yang, S., Peshek, T. J., Braid, J. L., & French, R. H. (2019). Automated pipeline for photovoltaic module electroluminescence image processing and degradation feature classification. *IEEE Journal of Photovoltaics*, 9(5), 1324–1335. <https://doi.org/10.1109/jphotov.2019.2920732>
- Karimi, A. M., Fada, J. S., Liu, J., Braid, J. L., Koyuturk, M., & French, R. H. (2018). Feature Extraction, Supervised and Unsupervised Machine Learning Classification of PV Cell Electroluminescence Images. In 2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC) (A Joint Conference of 45th IEEE PVSC,

- 28th PVSEC & 34th EU PVSEC (pp. 0418–0424). IEEE. 10.1109/PVSC.2018.8547739.
- Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. Proceedings of 2014 Science and Information Conference, SAI 2014, 372–378. 10.1109/SAI.2014.6918213.
- Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53 (8), 5455–5516. <https://doi.org/10.1007/s10462-020-09825-6>
- Kim, S., Kim, W., Noh, Y., & Park, F. C. (2017). Transfer learning for automated optical inspection. In 2017 International Joint Conference on Neural Networks (IJCNN) (Vol. 2017-May, pp. 2517–2524). IEEE. 10.1109/IJCNN.2017.7966162.
- Köntges, M., Kunze, I., Kajari-Schröder, S., Breitenmoser, X., & Björneklett, B. (2010). Quantifying the risk of power loss in PV modules due to micro cracks. 25th European Photovoltaic Solar Energy Conference and Exhibition, (September), 6–10. 10.4229/25thEUPVSEC2010-4BO.9.4.
- Köntges M., Kurtz S., Packard C., Jahn U., Berger K.A., Kato, K., ... Van Iseghem M. (2014). IEA-PVPS T13-01 2014 Review of Failures of Photovoltaic Modules Final (Vol. 1911). 978-3-906042-16-9.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2, 1097–1105.
- Kumar, G., & Bhatia, P. K. (2014). A detailed review of feature extraction in image processing systems. International Conference on Advanced Computing and Communication Technologies, ACCT, 5–12. 10.1109/ACCT.2014.74.
- Langley, P., & Sage, S. (1994). Induction of Selective Bayesian Classifiers. In Uncertainty Proceedings 1994 (pp. 399–406). Elsevier. 10.1016/B978-1-55860-332-5.50055-9.
- Luo, Z., Cheng, S. Y., & Zheng, Q. Y. (2019). Corrigendum: GAN-Based Augmentation for Improving CNN Performance of Classification of Defective Photovoltaic Module Cells in Electroluminescence Images (IOP Conf. Ser.: Earth Environ. Sci. 354 012106). IOP Conference Series: Earth and Environmental Science, 354, 012132. 10.1088/1755-1315/354/1/012132.
- Maeda-Gutiérrez, V., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., ... Olvera-Olvera, C. A. (2020). Comparison of convolutional neural network architectures for classification of tomato plant diseases. *Applied Sciences (Switzerland)*, 10(4). 10.3390/app10041245.
- Mathias, N., Shaikh, F., Thakur, C., Shetty, S., Dumane, P., & Chavan, D. S. (2020). Detection of Micro-Cracks in Electroluminescence Images of Photovoltaic Modules. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3563821>
- Namatévs, I. (2018). Deep convolutional neural networks: Structure, feature extraction and training. *Information Technology and Management Science*, 20(1), 40–47. <https://doi.org/10.1515/itms-2017-0007>
- Nurhanna, A. A., & Othman, M. F. (2017). Multi-class support vector machine application in the field of agriculture and poultry: A review. *Malaysian Journal of Mathematical Sciences*, 11(S), 35–52.
- Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *Journal of Remanufacturing*, 1–20. Retrieved from <http://arxiv.org/abs/1811.03378>.
- O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks, (December). Retrieved from <http://arxiv.org/abs/1511.08458>.
- Prati, R. C., Batista, G. E. A. P. A., & Silva, D. F. (2015). Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowledge and Information Systems*, 45(1), 247–270. <https://doi.org/10.1007/s10115-014-0794-3>
- Qian, X., Li, J., Cao, J., Wu, Y., & Wang, W. (2020). Micro-cracks detection of solar cells surface via combining short-term and long-term deep features. *Neural Networks*, 127, 132–140. <https://doi.org/10.1016/j.neunet.2020.04.012>
- Rahman, M. R. U., & Chen, H. (2020). Defects Inspection in Polycrystalline Solar Cells Electroluminescence Images Using Deep Learning. *IEEE Access*, 8, 40547–40558. <https://doi.org/10.1109/ACCESS.2020.2976843>
- Redmon, J. (n.d.). DarkNet: Open Source Neural Networks in C. Retrieved from <http://pjreddie.com/darknet>.
- Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua, 6517–6525. 10.1109/CVPR.2017.690.
- REN21. (2020). Renewables 2020 Global Status Report. REN21 Secretariat. Retrieved from <http://www.ren21.net/resources/publications/>.
- Russakovskiy, O., Deng, J., Su, H., Krause, J., Sathesh, S., Ma, S., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211–252. <https://doi.org/10.1007/s11263-015-0816-y>
- Sengur, A., Akbulut, Y., Budak, U., & Comert, Z. (2019). White Blood Cell Classification Based on Shape and Deep Features. *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*, 1–4. 10.1109/IDAP.2019.8875945.
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.
- Su, B., Chen, H. yong, Chen, P., Bian, G., Liu, K., & Liu, W. (2020). Deep Learning-based Solar-Cell Manufacturing Defect Detection with Complementary Attention Network. *IEEE Transactions on Industrial Informatics*, 3203(c), 1–10. 10.1109/TII.2020.3008021.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June, 1–9. 10.1109/CVPR.2015.7298594.
- Tang, W., Yang, Q., Xiong, K., & Yan, W. (2020). Deep learning based automatic defect identification of photovoltaic module using electroluminescence images. *Solar Energy*, 201(November 2019), 453–460. 10.1016/j.solener.2020.03.049.
- Tang Wuqin, Quiang Yang, W. Y. (2019). Deep Learning Based Models For Defect Detection of Mono Cr. PV Modules with EL Images Using Data Augmentation. In 2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Macao, Macao. (pp. 1–5).
- Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*, ahead-of-p(ahead-of-print). 10.1016/j.aci.2018.08.003.
- Toğacıar, M., Ergen, B., Cömert, Z., & Özyurt, F. (2020). A Deep Feature Learning Model for Pneumonia Detection Applying a Combination of mRMR Feature Selection and Machine Learning Models. *IRBM*, 41(4), 212–222. <https://doi.org/10.1016/j.irbm.2019.10.006>
- Tong, W., Haiyong, C., Image, A., & Cells, M. (2018). Automatic detection of photovoltaic module cells using multi-channel convolutional neural network. *2018 Chinese Automation Congress (CAC)*, 60741307, 3571–3576. <https://doi.org/10.1109/CAC.2018.8623258>
- Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. New York, NY: Springer New York. 10.1007/978-1-4757-3264-1.
- Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4), 611–629. <https://doi.org/10.1007/s13244-018-0639-9>