

ARTICLE

[DDSWF] Deep Defect-Specific Weighted Features: An Advanced Framework for High-Precision Identification of Critical Defects in ELPV Images

Uğur Çiğdem^{1,*} and Kenan Zengin^{2,*}

¹Computer Programming, Computer Technology, Tokat Gaziosmanpasa University, Tokat, Türkiye

²Computer Engineering, Engineering Faculty, Tokat Gaziosmanpasa University, Tokat, Türkiye

*Corresponding Authors: Uğur Çiğdem. Email: ugur.cigdem@gop.edu.tr; Kenan Zengin. Email: kenan.zengin@gop.edu.tr

Received: 15 July 2026; Accepted: 23 September 2026; Published: 22 October 2026

ABSTRACT: Defects such as finger defects, broken cells, and dark areas in solar panels reduce module efficiency and may cause premature failures. Although electroluminescence (EL) imaging is widely used as an industrial inspection standard because of its practicality, complex defect morphologies and expert-dependent manual inspection procedures increase the demand for automated and highly reliable classification systems. In this study, a hybrid framework called Deep Defect-Specific Weighted Features (DDSWF) is proposed for the reliable identification of critical defect types in electroluminescence (EL) images of photovoltaic (PV) cells. Existing public datasets generally provide defect severity-based labels and contain insufficient samples for classifying specific defect morphologies. Therefore, a defect-oriented Electroluminescence Photovoltaic (ELPV) subset containing three critical defect classes was created from the publicly available ELPV dataset and annotated by experts according to the International Electrotechnical Commission (IEC) TS 60904-13:2018. Since deep learning and object detection architectures are prone to overfitting under limited data conditions, this study adopts a hybrid approach combining Gray-Level Co-occurrence Matrix-derived texture features, Histogram of Oriented Gradients-based shape descriptors, and convolutional neural network representations. For feature selection, an Adaptive Neighborhood Component Analysis method assisted by Quantum-Behaved Particle Swarm Optimization was developed to dynamically optimize data distribution characteristics, model performance feedback, and Fisher score contributions. Evaluation on the expert-relabeled ELPV subset showed that the proposed DDSWF framework achieved 96.46% overall test accuracy, 96.49% precision, 96.49% recall, 96.49% F1-score, and 99.0% Area Under the Curve (AUC) using a Support Vector Machine classifier. These results indicate that hybrid feature integration and adaptive weighting can provide high discriminative capability and promising classification performance under limited-data conditions, offering a potential framework for photovoltaic inspection applications.

KEYWORDS: Photovoltaic defect classification; feature extraction; adaptive neighborhood component analysis; quantum-behaved particle swarm optimization; feature selection; machine learning

1 Introduction

The global energy sector has experienced a substantial transition from fossil fuel-based energy generation to renewable energy sources, driven by climate objectives and energy security concerns. According to recent reports from the international energy agencies, solar photovoltaic (PV) technology has become a primary driver of this transformation. The International Energy Agency (IEA) projects that a substantial portion of renewable energy capacity growth will be attributed to solar PV systems, while the International Renewable Energy Agency (IRENA) reports [1] that renewable sources accounted for more than 85% of newly installed global power capacity in 2024. As global demand for PV modules continues to increase,

manufacturers are expected to expand production capacity while maintaining high product quality through scalable and reliable inspection processes.

Defects introduced during PV module manufacturing can significantly reduce energy conversion efficiency and may lead to premature field failures. Consequently, accurate and early defect detection has become a critical requirement for reducing production waste, improving manufacturing yield, and ensuring long-term module reliability. Among various inspection techniques, electroluminescence (EL) imaging has emerged as an industrial standard because of its ability to reveal microstructural defects and crystal irregularities in photovoltaic cells through non-destructive testing. EL imaging operates by capturing the infrared radiation emitted from PV cells when a forward electrical current is applied, enabling detailed visualization of internal defects at high spatial resolution [2].

In industrial environments, EL inspections are typically conducted before and after lamination. This strategy prevents defective cells from advancing to subsequent manufacturing stages, thereby reducing production costs and improving process efficiency. Furthermore, the EL inspection data support closed-loop manufacturing optimization. Defect maps generated from EL images facilitate root-cause analysis by identifying process-related issues such as improper lamination temperature, cell-handling robot vibrations, or soldering pressure variations. Corrective actions can subsequently be implemented and validated through EL inspections of future production batches. However, the effectiveness of such feedback mechanisms strongly depends on the rapid, consistent, and automated interpretation of EL images [2,3].

Despite the widespread adoption of EL imaging, image evaluation in industrial practice still relies heavily on visual inspection by experienced operators. Manual inspection introduces substantial bottlenecks in high-throughput manufacturing environments. Reliance on expert inspectors increases inspection time, introduces operator fatigue, and may lead to inconsistent classification results. Moreover, defects exhibiting similar morphological characteristics, such as finger defects, broken cells, and dark areas, are often difficult to distinguish reliably through subjective visual assessment. These limitations can negatively affect both quality assurance processes and manufacturing feedback systems [2,3].

To address these challenges, numerous automated defect-detection approaches have been proposed. However, most existing studies have focused on binary classification (defective/non-defective) or defect severity assessments. In contrast, defect-type classification, which is essential for process optimization and root-cause analysis, remains relatively underexplored. In addition, publicly available datasets generally lack detailed annotations describing specific defect morphologies. Although end-to-end deep learning architectures provide powerful representation capabilities, their large number of trainable parameters often leads to overfitting when trained on limited and imbalanced datasets. Furthermore, such models typically require high computational resources and provide limited interpretability regarding the contribution of individual features [3].

To overcome these limitations, this study proposes a novel hybrid framework termed Deep Defect-Specific Weighted Features (DDSWF) for the reliable classification of critical defect types in EL images. The proposed framework integrates hybrid feature extraction with adaptive feature selection to achieve strong discriminative capability and robust predictive performance on the held-out test set under limited-data conditions.

The main contributions of this study are summarized as follows:

- Expert-Annotated Defect-Oriented Dataset Subset: A new subset was constructed from the publicly available Electroluminescence Photovoltaic (ELPV) dataset and annotated according to the International Electrotechnical Commission (IEC) TS 60904-13:2018 guidelines. The subset contains three

critical defect categories: Finger Defect, Dark Area, and Broken Cell and was independently evaluated by two domain experts followed by consensus-based labeling.

- **Noise-Resistant Hybrid Feature Extraction:** A robust feature extraction architecture was developed by integrating texture features derived from Gray-Level Co-occurrence Matrix (GLCM), shape features extracted using Histogram of Oriented Gradients (HOG), and deep semantic features obtained from Convolutional Neural Networks (CNNs). This hybrid representation improves robustness while mitigating overfitting under limited-data conditions.
- **Quantum-Behaved Particle Swarm Optimization (QPSO) Based Adaptive Neighborhood Component Analysis (A-NCA):** A novel feature weighting and selection mechanism was introduced by extending conventional Neighborhood Component Analysis. The proposed method dynamically balances data distribution characteristics, model performance feedback, and Fisher score contributions through optimization using QPSO.
- **High Classification Performance with Enhanced Interpretability:** The proposed framework achieves high classification accuracy while maintaining feature-level interpretability and lower computational complexity compared with end-to-end deep learning models containing millions of parameters. Experimental results demonstrate classification performance exceeding 96% across multiple evaluation metrics.

The remainder of this paper is organized as follows. [Section 2](#) presents the related work. [Section 3](#) describes the dataset. [Section 4](#) presents the proposed DDSWF framework in detail. [Section 5](#) presents the experimental setup, performance evaluation, and discussion of the results. Finally, [Section 6](#) concludes the paper and outlines future research directions.

2 Related Work

Automatic defect detection in photovoltaic (PV) modules using electroluminescence (EL) imaging has become one of the most widely investigated topics in computer vision and artificial intelligence over the past decade. Early studies primarily relied on conventional image processing and machine learning techniques. Tsai et al. [4] introduced a defect detection approach that combines Fourier image reconstruction with statistical texture analysis for EL image inspection. Similarly, Anwar and Abdullah [5] employed anisotropic diffusion filtering and Support Vector Machines (SVM) to identify microcracks in photovoltaic cells. Deitsch et al. [6] introduced the ELPV dataset, one of the first publicly available benchmark datasets for EL-based PV defect analysis, and evaluated traditional machine learning classifiers, including Random Forest (RF) and SVM. Their work established a benchmark that has been widely adopted in subsequent studies. Karimi et al. [7] further developed an automated inspection pipeline that integrates EL image preprocessing with RF and SVM classifiers to categorize cells into healthy, cracked, and corroded classes.

The emergence of Convolutional Neural Networks (CNNs) significantly advanced the field by enabling automatic feature learning. Akram et al. [8] designed a lightweight CNN architecture for the ELPV dataset and achieved classification accuracies exceeding 93%, outperforming conventional machine learning approaches. Chen et al. [9] investigated automated defect identification in electroluminescence images of solar modules, while Chen et al. [10] focused on accurate crack detection in electroluminescence images of solar cells. Tang et al. [11] integrated data augmentation strategies with CNN models and achieved an average accuracy of approximately 92% on polycrystalline silicon cells. In the object detection domain, Zhang et al. [12] proposed a hybrid framework based on Region-based Convolutional Neural Network (R-CNN) and Region-based Fully Convolutional Network (R-FCN), while Zhao et al. [13] and Meng et al. [14] developed customized You Only Look Once (YOLO)-based architectures capable of real-time defect localization. Despite their strong performance, these deep learning models typically contain millions of trainable

parameters and are therefore prone to overfitting when trained on limited or imbalanced datasets. Moreover, they generally offer limited interpretability regarding feature contributions to classification decisions.

More recently, Vision Transformers (ViTs) and hybrid deep architectures have been introduced to photovoltaic defect analysis. Liu et al. [15] combined adaptive histogram equalization with EfficientNet-B0 and graph channel attention mechanisms to improve detection sensitivity. Wang et al. [16] employed a channel-attention-enhanced ResNet-152 architecture together with feature fusion strategies and reported a multiclass accuracy of 92.13%. Zhao et al. [17] proposed a High-Resolution Network (HRNet)-based framework for EL image analysis, whereas Zhang et al. [18] utilized Neural Architecture Search (NAS) and knowledge distillation techniques to design lightweight networks for photovoltaic inspection. Although transformer-based architectures demonstrate superior capability in capturing global contextual information, their practical deployment in industrial manufacturing environments remains challenging because of high computational complexity and the requirement for large-scale annotated datasets.

Although significant progress has been achieved, most existing studies focus on binary classification (defective vs. non-defective) or defect severity assessment based on predefined degradation levels (e.g., 0%, 33%, 66%, and 100%). In contrast, accurate identification of specific defect types, such as finger defects, broken cells, and dark areas, remains relatively underexplored. From an industrial perspective, defect-type classification is considerably more valuable because different defect categories often indicate different root causes within the manufacturing process. For example, finger defects may be associated with metallization failures, broken cells may result from cell handling and transportation issues, whereas dark areas can indicate material degradation or process inconsistencies. Therefore, reliable defect-type classification can directly support root-cause analysis, process optimization, and manufacturing feedback systems [19–22].

A detailed review of the literature reveals three major challenges that continue to limit the development of defect-type-oriented photovoltaic defect classification systems:

- **Lack of Defect-Specific EL Datasets:** Existing public datasets, including ELPV, primarily provide defect severity labels rather than detailed morphological annotations. Consequently, datasets suitable for fine-grained defect-type classification remain scarce.
- **Challenges in Cell-Level Segmentation:** Automatic cell extraction from EL images remains difficult because of varying panel configurations, perspective distortions, illumination inconsistencies, and complex background structures. Reliable segmentation is a prerequisite for obtaining high-quality samples for defect-type classification.
- **Limited Use of Hybrid Feature Learning Frameworks:** End-to-end deep learning models generally require large amounts of labeled data, whereas traditional machine learning approaches often rely on a single feature representation and therefore provide limited discriminative capability. Hybrid frameworks that integrate texture, shape, and deep semantic features while dynamically weighting their contributions through adaptive feature selection remain largely unexplored in the photovoltaic defect analysis literature.

Motivated by these challenges, the present study directly addresses each of these three limitations. First, a defect-oriented ELPV subset containing expert-annotated critical defect categories was constructed. Second, a robust hybrid feature extraction framework combining texture, shape, and deep semantic representations was developed. Third, a novel QPSO-based Adaptive Neighborhood Component Analysis (A-NCA) mechanism was proposed to dynamically weight and select discriminative features for reliable defect-type classification under limited-data conditions.

3 Dataset Description

The publicly available ELPV dataset proposed by Deitsch et al. [6] was used in this study. The publicly available ELPV dataset contains 2624 electroluminescence (EL) images, each with a resolution of 300×300 pixels, an 8-bit grayscale representation, and 256 gray-level intensities. These images were acquired from 44 photovoltaic modules exhibiting different levels of degradation and defect severity. Among the available samples, 1508 images correspond to defect-free cells, whereas 1116 images contain various types of defects.

The original ELPV dataset categorizes photovoltaic cells according to defect severity levels (0%, 33%, 66%, and 100%) and does not provide detailed annotations regarding the physical or morphological characteristics of defects. Although such severity-based labeling is suitable for general condition assessment, it provides limited support for defect-type-oriented analysis required in manufacturing quality control and maintenance applications. From an industrial perspective, different defect mechanisms can have substantially different impacts on module reliability, energy yield, and economic losses [6].

To address this limitation, a defect-oriented subset was constructed from the original ELPV dataset. The primary motivation behind this approach is to identify specific defect categories that can support early intervention strategies and proactive decision-making processes in photovoltaic manufacturing environments. Consistent with findings reported in the literature, defect types associated with the highest performance degradation and economic impact were prioritized, including finger defects, dark areas, and broken cells.

During the dataset construction process, all selected images were independently re-annotated manually by two domain experts using a defect-type-based labeling strategy. This methodology is consistent with previous studies that employed expert-guided annotation procedures for electroluminescence image analysis. Furthermore, the annotation process was conducted in accordance with the electroluminescence imaging and defect morphology guidelines specified in *IEC TS 60904-13*. The resulting subset provides a more detailed and application-oriented representation of photovoltaic defects, thereby enabling the development and evaluation of defect-type classification frameworks beyond conventional severity-based assessment.

Finger defect, dark area, and broken cells are among the defect types in photovoltaic cells that not only directly affect energy production performance but also exhibit distinct morphological characteristics in electroluminescence (EL) images. Finger defects increase electrical losses within the cell by causing discontinuities in current conduction pathways, whereas dark area defects indicate localized losses in active regions and irregularities in current generation. Broken cells, on the other hand, compromise the structural integrity of the cell, leading to power degradation and potentially resulting in long-term cell failures. Furthermore, these three defect categories represent different degradation mechanisms originating from manufacturing errors, mechanical stresses, and operational conditions, thereby providing an appropriate framework for evaluating the effectiveness of the proposed method across diverse defect characteristics.

Due to their significant impact on photovoltaic efficiency, these defect types were prioritized in this study. A total of 596 images were manually re-annotated by domain experts, including 263 finger defect images, 104 dark area images, and 229 broken cell images. The original ELPV dataset was not designed for fine-grained defect-type classification. Therefore, a defect-oriented subset was constructed by selecting images belonging to the three defect categories investigated in this study: Broken cell, dark area, and finger defect. The remaining 520 images were excluded because they represented other defect types outside the scope of the present three-class classification problem. No additional quality-based exclusion criterion was applied.

The limited number of samples in the resulting subset further increases the difficulty of automatic defect classification. In addition to causing substantial reductions in panel efficiency, these defect types may induce localized temperature increases, leading to hot-spot formation and consequently reducing system

reliability. The process of constructing the new dataset is illustrated in Fig. 1. Representative examples of the re-annotated defect classes are presented in Fig. 2a–c.

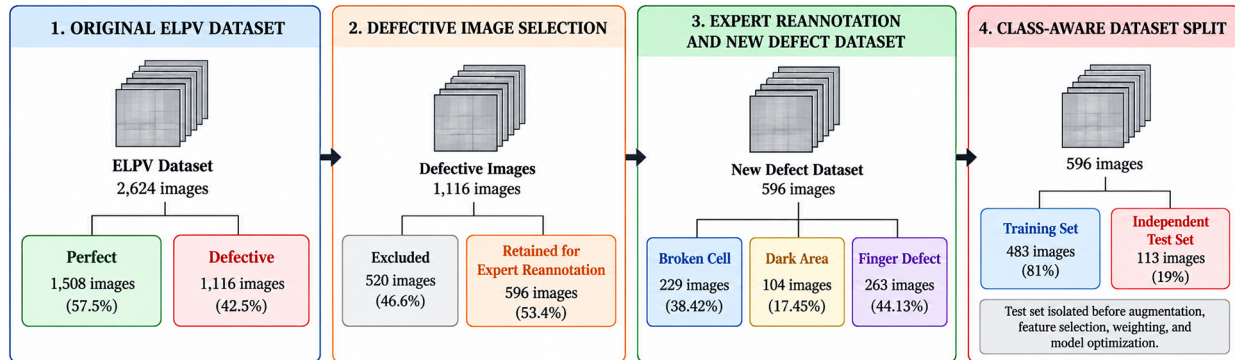


Figure 1: Overview of the expert-guided dataset construction and reannotation process.

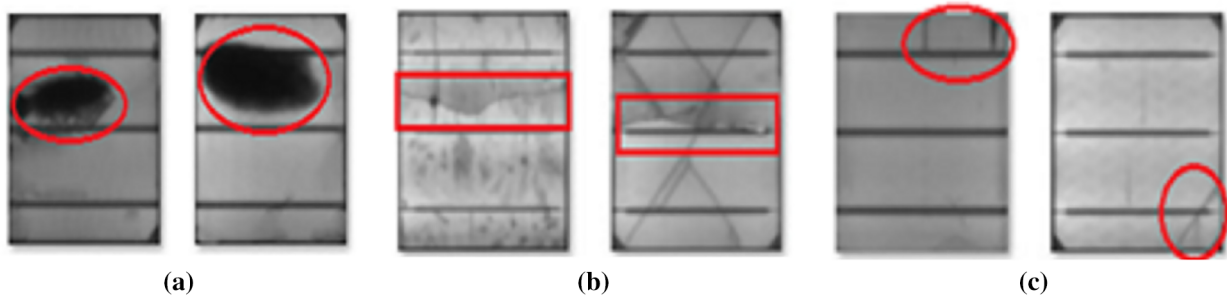


Figure 2: Representative electroluminescence images of the three defect categories included in the proposed dataset: (a) Dark area, (b) Broken cell, and (c) Finger defect.

4 Materials and Methods

4.1 Data Preprocessing

The dataset was partitioned using a class-aware holdout strategy designed to preserve adequate representation of minority defect categories within the held-out test subset. This approach was adopted to enable reliable estimation of class-specific evaluation metrics, particularly for the Dark Area category, which contains substantially fewer samples than the remaining defect classes. The resulting overall split was 81% ($n = 483$) training and 19% ($n = 113$) testing; class-specific proportions were intentionally non-proportional to ensure adequate representation of the minority Dark Area class in the held-out set. To guarantee reproducibility of the experiments, the random seed was fixed at 42 during the partitioning process. This approach helped preserve the non-proportional class-aware split while reducing sampling bias, which is particularly important in imbalanced datasets. The splitting procedure was performed independently for each defect class to ensure adequate representation of all defect categories in both the training and test sets. As the publicly available ELPV dataset does not provide module-level identification information, the split was conducted at the image level, and potential overlap of visually similar images from the same module between the two subsets cannot be completely excluded. Nevertheless, the test set was separated before model development and kept isolated from subsequent training and optimization procedures to minimize the potential risk of data leakage. To improve the model's robustness to noise and data variations, six different image enhancement and augmentation techniques were applied to the training data. First, Contrast

Limited Adaptive Histogram Equalization (CLAHE) was employed to enhance local image details and improve contrast. Subsequently, Gamma Correction and Histogram Equalization were applied to enhance the robustness of the model to illumination variations. To further enrich the dataset, Random Illumination, Random Crop and Resize, and Elastic Deformation techniques were applied. Through elastic deformation, localized spatial distortions were introduced into the images, thereby generating diverse geometric variations of the defects. To alleviate class imbalance, random combinations of these augmentation techniques were applied to the Broken Cell and Finger Defect classes, generating synthetic samples at a $3\times$ ratio relative to the original training sample count. Since the Dark Area class initially contained a limited number of samples (104 images), which could negatively affect classification performance, all augmentation techniques were applied more extensively to this class, generating synthetic samples at a $5\times$ ratio relative to the original training sample count.

For the Broken Cell, Dark Area, and Finger Defect classes, 567, 355, and 669 synthetic images were generated, corresponding to synthetic-to-original ratios of $3.0\times$ ($567/189$), $5.0\times$ ($355/71$), and $3.0\times$ ($669/223$), respectively. These images were added to the 483 original training images, resulting in 2074 training images. The resulting class-wise distribution is presented in Table 1.

Table 1: Distribution of training images before and after data augmentation.

Class Name	Train Image Numbers	Augmentation Image Numbers	Synthetic-to-Original Ratio	Total
Broken Cell	189	567	$3.0\times$	756
Dark Area	71	355	$5.0\times$	426
Finger Defect	223	669	$3.0\times$	892
Total	483	1591	$3.3\times$	2074

The test subset was physically separated before model development and remained unseen during feature weighting, QPSO optimization, classifier selection, and training. In five-fold cross-validation, one fold was used for validation, while the remaining four folds were augmented and used for training. The final model was evaluated once on the held-out test subset.

4.2 Feature Extraction

Feature extraction was performed using a hybrid representation to capture complementary characteristics of photovoltaic defects. Texture features were extracted using GLCM, shape features using HOG, and deep semantic features using a pretrained 18-layer Residual Network (ResNet-18) network. The resulting feature vectors were concatenated to form a unified hybrid feature space for subsequent feature selection. Fig. 3a presents the HOG-based feature extraction, Fig. 3b shows the GLCM-based feature extraction, and Fig. 3c illustrates the CNN-based feature extraction.

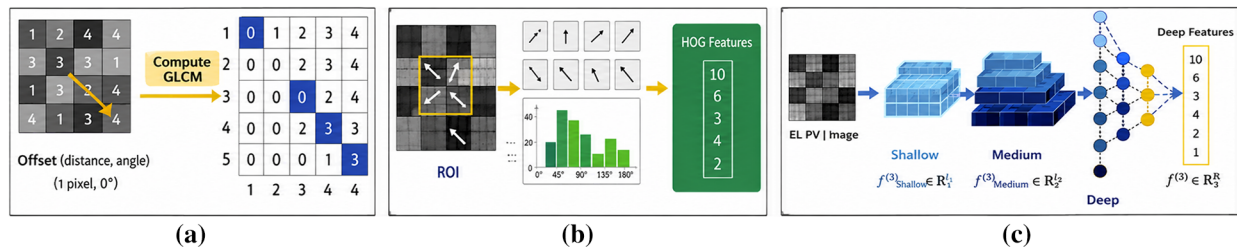


Figure 3: Overview of the hybrid feature extraction framework: (a) GLCM-based texture feature extraction, (b) HOG-based shape descriptor extraction, and (c) CNN-based deep feature extraction.

HOG descriptors were extracted using a cell size of 32×32 , a block size of 2×2 , a block overlap of 1×1 , and 9 unsigned orientation bins. GLCM features were computed using the Dark Mask configuration with distances of 1, 2, 4, 8, and 16 pixels in four directions (0° , 45° , 90° , and 135°) [23].

Deep features were extracted from three different layers of the pretrained ResNet18 network (Conv1_Relu, Res3a_Relu, and Pool5) to represent shallow, intermediate, and deep semantic information. Each EL image was converted into a defect-aware three-channel input and resized to the network input size before feature extraction. For each CNN layer, a fixed-dimensional feature vector of 300 activation features was retained based on preliminary experiments. Preliminary experiments with different feature vector lengths showed that increasing the number of retained activation features resulted in only marginal improvements in classification performance while substantially increasing the computational cost. This dimensionality provided a good trade-off between computational efficiency and classification performance, while ensuring a balanced representation with the HOG and GLCM descriptors. The retained features were subsequently refined by the proposed Adaptive-NCA feature selection algorithm. Finally, all feature vectors were concatenated and provided to the proposed Adaptive-NCA feature selection framework.

4.3 QPSO-Based Adaptive-NCA Feature Selection Method

The hybrid feature representation contains a large number of redundant features that may increase computational cost and reduce classification performance. Therefore, the proposed Adaptive-NCA method was employed to identify the most informative features. Unlike conventional NCA, which evaluates features using a single criterion, Adaptive-NCA combines distribution-based, performance-based, and Fisher score-based weighting to estimate feature importance more comprehensively.

It should be noted that the proposed Adaptive-NCA algorithm does not directly compute the feature weights. Instead, the distribution-based, performance-based, and Fisher-score-based weighting components are first computed independently. The performance-based weights are obtained from SVM (fitcecoc) classification results, whereas the remaining components are calculated using statistical feature distribution and Fisher discriminative analysis. These three components are then fused using the QPSO-optimized weighting coefficients (α , β , and γ) to generate the final hybrid feature weights, which serve as the input to the Adaptive-NCA algorithm for the final feature selection. As defined in Eqs. (1) and (2), the final hybrid feature weights were obtained by combining the three weighting components using the QPSO-optimized coefficients α , β , and γ .

$$W_{\text{hybrid},i} = \alpha * W_{\text{distribution},i} + \beta * W_{\text{performance},i} + \gamma * W_{\text{fisher},i} \quad (1)$$

$$\alpha + \beta + \gamma = 1 \quad (2)$$

The optimized coefficients obtained by QPSO were $\alpha = 0.210$, $\beta = 0.362$, and $\gamma = 0.428$. This adaptive weighting strategy improves feature discrimination by integrating complementary information from data distribution, classifier performance, and class separability, resulting in a compact and highly discriminative feature subset for classification. Fig. 4 illustrates the proposed QPSO-Based Adaptive-NCA framework.

The QPSO population size was set to 30 particles based on preliminary experiments, which provided a good balance between optimization accuracy and computational efficiency. Since QPSO optimizes only the three weighting coefficients (α , β , and γ) in the proposed Adaptive-NCA framework, the population size was selected according to the complexity of the optimization problem rather than the number of training samples. Increasing the population size beyond 30 did not yield noticeable improvements in optimization or classification performance, while increasing computational cost.

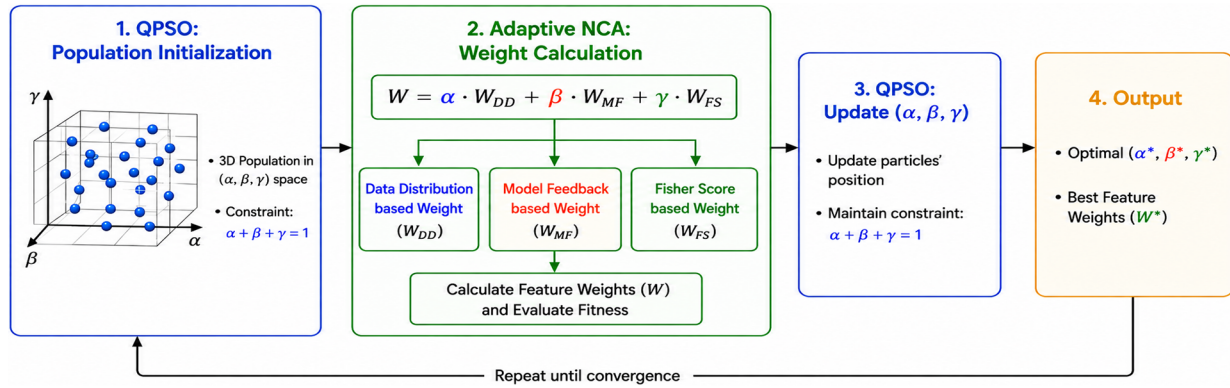


Figure 4: Workflow of the proposed QPSO-based Adaptive-NCA feature selection framework.

4.3.1 Weighting Based on Data Distribution ($W_{\text{distribution}}$)

The distribution-based weighting strategy evaluates the discriminative capability of each feature according to its statistical distribution across classes. Features with greater inter-class variability receive higher weights, whereas features with limited variation are assigned lower weights. This mechanism improves class separability by emphasizing informative features while suppressing redundant ones [24].

Fig. 5 presents the raw variance values calculated from the extracted features before normalization. Fig. 5a shows the values in the original feature order, whereas Fig. 5b presents the same values sorted in descending order. The variance values are normalized according to Eq. (3) to obtain the distribution-based weighting component ($W_{\text{distribution}}$). These weights constitute the first component of the proposed hybrid weighting strategy and are subsequently combined with the performance-based and Fisher score-based weighting components.

$$W_{\text{distribution},i} = \frac{\text{Var}(X_i)}{\sum_{j=1}^d \text{Var}(X_j)} \quad (3)$$

where i denotes the index of the feature whose weight is being calculated, j is the summation index covering all features, and d is the total number of features. This notation was revised to prevent the same index from being used simultaneously as a free index and a summation index.

4.3.2 Weighting Based on Performance Model Feedback ($W_{\text{performance}}$)

The performance-based weighting strategy estimates feature importance according to the classification performance obtained from the SVM classifier. Specifically, each feature is evaluated independently by using only the corresponding feature values as the input to the SVM classifier, while the class labels remain unchanged. The classification performance of the (i) -th feature is evaluated using K-fold cross-validation, with ($K = 5$), and its classification error (e_i) is calculated according to Eq. (4). Features associated with lower classification errors therefore receive higher performance-based weights, whereas features associated with higher classification errors receive lower weights. This mechanism enables Adaptive-NCA to prioritize features that contribute more strongly to classification performance. The performance-based weights are computed using Eq. (5). To avoid division by zero when $e_i = 0$, a small positive constant ε is added to the error term.

$$e_i = 1 - \frac{1}{5} \sum_{k=1}^5 \text{Accuracy}_{i,k} \quad (4)$$

where $Accuracy_{i,k}$ denotes the classification accuracy obtained using only the (i) -th feature in the (k) -th fold.

$$W_{performance,i} = \frac{\frac{1}{e_i + \varepsilon}}{\sum_{j=1}^n \left(\frac{1}{e_j + \varepsilon} \right)} \quad (5)$$

where e_i denotes the error rate associated with the contribution of the i -th feature to the classification model. The term $\frac{1}{e_i}$ represents the inverse error rate, which is used to emphasize features with lower classification errors. The denominator $\sum_{j=1}^n \left(\frac{1}{e_j} \right)$ corresponds to the normalization term computed as the sum of inverse error rates over all n features, ensuring that the resulting weights are properly scaled and comparable across features.

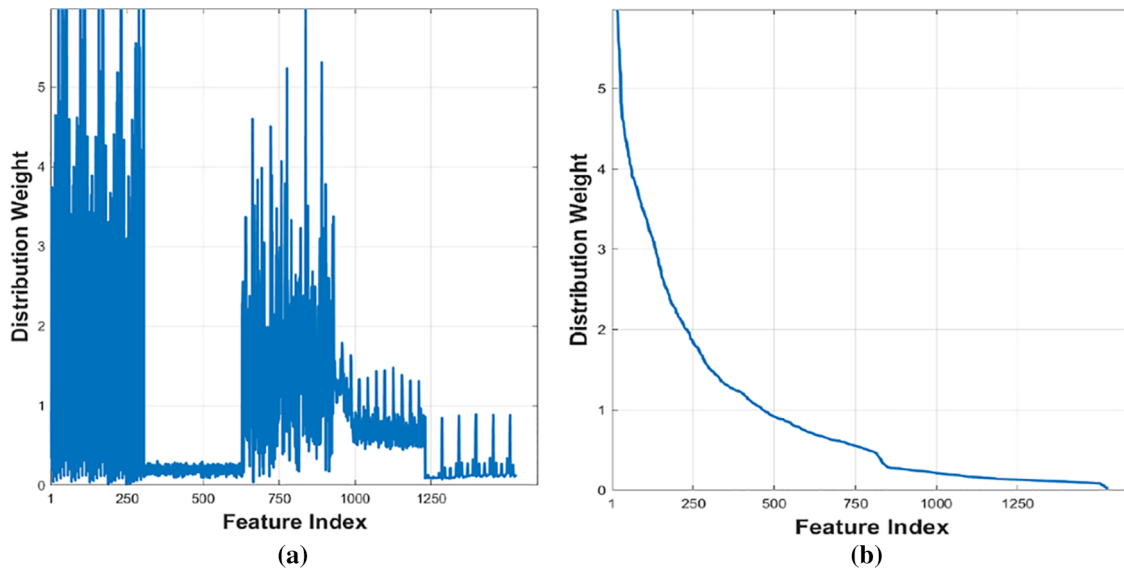


Figure 5: (a) Unsorted raw variance values; (b) Sorted raw variance values of the extracted features used for distribution-based weighting prior to Adaptive-NCA.

Fig. 6 illustrates the performance-based weighting component used in the proposed framework. After feature extraction, the extracted features are evaluated using an SVM classifier implemented with MATLAB's `fitcecoc` function to obtain the classification accuracy and corresponding error rate for each feature. These values are subsequently converted into normalized performance-based feature weights. The resulting weights constitute one of the three inputs to the proposed hybrid weighting strategy and are later combined with the distribution-based and Fisher-score-based weights before being supplied to the Adaptive-NCA algorithm for the final feature selection. It should be noted that the feature numbers evaluated in Fig. 6a,b correspond to the original extracted feature space before Adaptive-NCA optimization and do not represent the final feature subset selected by Adaptive-NCA. Instead, the performance-based weights computed in this stage serve as one of the three inputs to the proposed hybrid weighting strategy. After QPSO optimizes the weighting coefficients (α , β , and γ), the resulting Adaptive-NCA weights are used to determine the final discriminative feature subset.

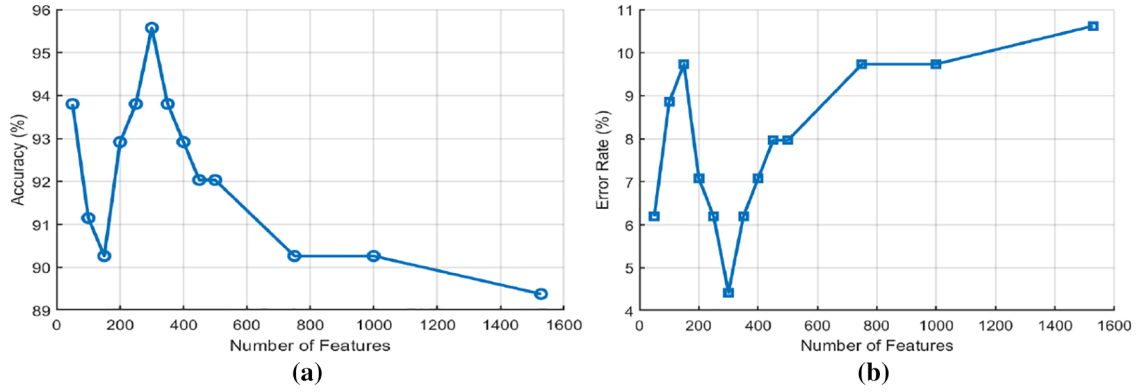


Figure 6: Performance-based feature weights computed prior to Adaptive-NCA: (a) Classification accuracy and (b) Classification error rate.

4.3.3 Fisher Score-Based Weighting (W_{fisher})

Fisher score was used to evaluate the discriminative capability of each feature according to its between-class separability and within-class variance. Features with higher Fisher scores were assigned larger weights, indicating greater importance for classification [25]. The Fisher-based weights were computed using Eqs. (6) and (7).

$$W_{\text{fisher},i} = \frac{F_i}{\sum_{j=1}^n F_j} \quad (6)$$

where F_i denotes the Fisher score of the i -th feature, which is computed using the formulation given in (7).

$$F_i = \frac{\sum_{c=1}^C N_c (\mu_{i,c} - \mu_i)^2}{\sum_{c=1}^C N_c \sigma_{i,c}^2} \quad (7)$$

where, C denotes the number of classes, N_c the number of samples in class c , $\mu_{i,c}$ and $\sigma_{i,c}^2$ the class-wise mean and variance of feature i . μ_i represents the overall mean of feature i .

Fig. 7a shows the unsorted Fisher score-based feature weights, while Fig. 7b shows the sorted Fisher score-based feature weights computed for all extracted features prior to the Adaptive-NCA stage. Both the original feature-order representation and the descendingly sorted representation are presented. The unsorted representation preserves the original feature indices, whereas the sorted representation facilitates direct comparison of the relative weight magnitudes. The features are ranked in descending order according to their Fisher scores, where higher values indicate greater discriminative ability. These normalized Fisher-score-based weights constitute the third component of the hybrid weighting framework and are subsequently combined with the distribution-based and performance-based components according to Eq. (1).

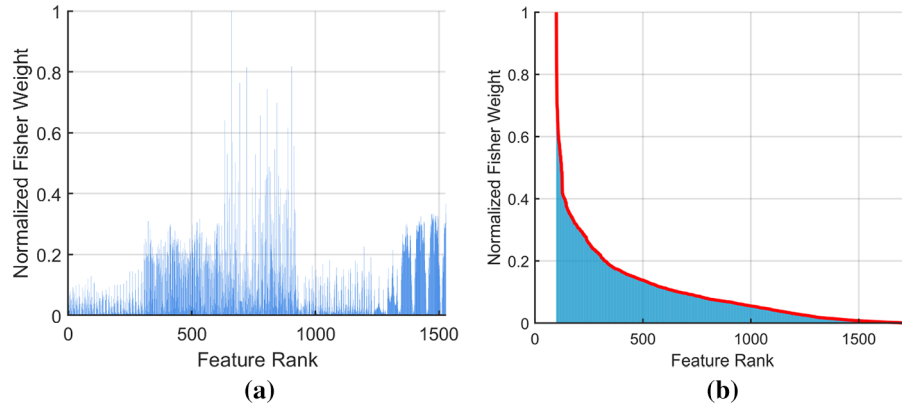


Figure 7: (a) Unsorted Fisher score-based feature weights; (b) Sorted fisher score-based feature weights computed for all extracted features prior to Adaptive-NCA.

The original Neighborhood Component Analysis (NCA) loss function is computed using the formulation given in (8). Here, y_i and y_j denote the class labels of samples x_i and x_j , respectively, and $y_i = y_j$ indicates that the two samples belong to the same class.

$$L_{NCA} = - \sum_{i=1}^N \log \left(\sum_{\substack{j \neq i \\ y_j = y_i}} P_{ij} \right) \quad (8)$$

where N denotes the total number of data points in the dataset, and P_{ij} represents the probability of data point i selecting data point j as its neighbor. The probability P_{ij} is calculated using the formulation given in (9).

$$P_{ij} = \frac{\exp \left[- \|Ax_i - Ax_j\|^2 \right]}{\sum_{k \neq i} \exp \left[- \|Ax_i - Ax_k\|^2 \right]} \quad (9)$$

where A denotes the learned transformation matrix, x_i and x_j represent data points, and $\|\cdot\|^2$ indicates the squared Euclidean distance. In the Adaptive-NCA framework, the hybrid updated loss function is formulated using the expression given in (10).

$$L_{NCA-hybrid} = - \sum_{i=1}^N \log \left(\sum_{\substack{j \neq i \\ y_j = y_i}} P_{ij}^{hybrid} \right) \quad (10)$$

where P_{ij}^{hybrid} denotes the hybrid neighborhood probability between data points i and j . Based on this hybrid weighting scheme, the neighborhood probability is updated by incorporating the hybrid feature weights into the distance calculation, as given in (11).

$$P_{ij}^{hybrid} = \frac{\exp \left[- \sum_{f=1}^D W_{hybrid,f} \left((Ax_i)_f - (Ax_j)_f \right)^2 \right]}{\sum_{k \neq i} \exp \left[- \sum_{f=1}^D W_{hybrid,f} \left((Ax_i)_f - (Ax_k)_f \right)^2 \right]} \quad (11)$$

where $W_{hybrid,f}$ denotes the hybrid weight assigned to the f -th feature, D denotes the total number of

features, A denotes the learned transformation matrix, and x_i , x_j , and x_k denote the feature vectors of samples i , j , and k , respectively. The term $(Ax_i)_f$ denotes the f -th component of the transformed feature vector Ax_i . Eq. (12) presents the feature selection scheme, where $W_{sorted,i}$ denotes the normalized feature weights in descending order, $K_{selected}$ denotes the number of selected features, and ρ denotes the cumulative contribution threshold.

$$K_{selected} = \min \left\{ m: \sum_{i=1}^m W_{sorted,i} \geq \rho \right\} \quad (12)$$

4.3.4 Quantum-Behaved Particle Swarm Optimization Algorithm (QPSO)

Quantum-Behaved Particle Swarm Optimization (QPSO) was employed to optimize the weighting coefficients (α , β , γ) of the proposed Adaptive-NCA framework. In this study, QPSO optimization is applied exclusively to the proposed SVM-based framework. The neural network, K-Nearest Neighbors (KNN), and tree-based classifiers were not optimized using QPSO; they were included only as baseline models for comparative evaluation using MATLAB's built-in implementations. Each particle represents a candidate coefficient set, and the fitness function simultaneously minimizes the SVM classification error and the number of selected features. Model performance was evaluated using 5-fold cross-validation with an Error-Correcting Output Codes Support Vector Machine (ECOC-SVM) classifier, and the resulting fitness values guided the QPSO search toward the optimal coefficient combination. The optimized coefficients were then used to compute the hybrid feature weights for the final feature selection process. All QPSO fitness evaluations and feature-weight optimization were performed exclusively within the training data; the held-out test set was never used during optimization. Eqs. (13)–(17) describe the QPSO update and fitness formulations [26]. The overall implementation procedure of the proposed QPSO-based Adaptive-NCA method is summarized in Algorithm 1.

Algorithm 1: Hybrid QPSO-based Adaptive-NCA

Input:	Output:
X_train, Y_train // Training feature matrix and class labels	BestWeights // Final Adaptive-NCA feature
N // Number of QPSO particles	BestK // Number of selected features
MaxIter // Maximum number of QPSO iterations	SelectedFeatures // Indices of selected features
ρ // Cumulative contribution threshold	gBestPosition // Optimized [α , β , γ]
ϕ // Feature-selection penalty coefficient	
$k(t)$ // QPSO contraction–expansion coefficient	
$K = 5$ // Number of cross-validation folds	
1: Normalize X_train	
2: Initialize N QPSO particles	// Update Personal Best
For each particle i:	28: if Fitness_i < pBestFitness_i then
Particle_i = [α_i , β_i , γ_i]	29: pBestFitness_i ← Fitness_i
Initialize α_i , β_i , γ_i randomly	30: pBestPosition_i ← Particle_i
subject to:	31: end if
$\alpha_i + \beta_i + \gamma_i = 1$	// Update Global Best
$\alpha_i, \beta_i, \gamma_i \geq 0$	32: if Fitness_i < gBestFitness then
3: Evaluate the initial fitness of each particle	33: gBestFitness ← Fitness_i

(Continued)

Algorithm 1 (continued)

using K-fold cross-validation on X_train, Y_train	34:	gBestPosition $\leftarrow$ Particle_i
4: Initialize:	35:	BestK $\leftarrow$ K_selected
pBestFitness_i $\leftarrow$ Fitness_i	36:	end if
pBestPosition_i $\leftarrow$ Particle_i	37:	end for
5: Determine:		// QPSO Position Update
gBestFitness $\leftarrow$ minimum(Fitness_i)	38:	Compute mean best position:
gBestPosition $\leftarrow$ corresponding Particle_i		mbest_d(t) = (1/N) $\sum$ pBestPosition_i,d(t)
6: for t = 1 to MaxIter do	39:	for each particle i do
7: for each particle i do	40:	for each dimension d $\in$ { α , β , γ } do
// Cross-validated Adaptive-NCA Weight Learning	41:	Generate:
8: for each fold f = 1 to K do		$\mu_{i,d}(t) \sim U(0,1)$
9: Split X_train, Y_train into:	42:	Generate:
X_CVTrain, Y_CVTrain		$u_{i,d}(t) \sim U(0,1)$
X_CVVal, Y_CVVal	43:	Compute local attractor:
		$P_{i,d}(t) = \mu_{i,d}(t) \times$
// Weight learning is performed only		pBestPosition_i,d(t)
// using the CV training fold		+ [1 - $\mu_{i,d}(t)$]
10: Compute Fisher-based feature weights:		$\times$ gBestPosition_d(t)
W_fisher =FisherScore(X_CVTrain,	44:	Generate:
Y_CVTrain)		$r \sim U(0,1)$
11: Compute performance-based feature weights:		
W_performance =PerformanceFeedback	45:	if r < 0.5 then
(X_CVTrain, Y_CVTrain)		
12: Compute distribution-based feature weights:	46:	$x_{i,d}(t+1) =$
W_distribution =DistributionWeight		$P_{i,d}(t) + k(t) \times mbest_d(t)$
(X_CVTrain)	–	$x_{i,d}(t) $
13: Compute hybrid feature weights:		$\times \ln(1/u_{i,d}(t))$
$W = \alpha_i \times W_distribution + \beta_i \times$	47:	else
$W_performance + \gamma_i \times W_fisher$		
14: Apply Adaptive-NCA optimization:	48:	$x_{i,d}(t+1) = P_{i,d}(t) - k(t) \times$
$W = \text{OptimizeNCA}(X_CVTrain,$		$ mbest_d(t) x_{i,d}(t) $
Y_CVTrain,W)		
15: Normalize feature weights:		$\times \ln(1/u_{i,d}(t))$
$W = W / \sum W$	49:	end if
// Feature Selection	50:	end for
16: Sort feature weights in descending order:	51:	Update Particle_i subject to:
[W_sorted, S_rank] =Sort(W, descending)		$\alpha_i + \beta_i + \gamma_i = 1$
17: Normalize sorted weights:		$\alpha_i, \beta_i, \gamma_i \geq 0$
$W_n = W_sorted / \sum W_sorted$	52:	end for
18: Compute cumulative contribution:	53:	end for

(Continued)

Algorithm 1 (continued)

CW = cumsum(W_n) 19: Determine the number of selected features: K_selected = min{m: CW(m) ≥ ρ} 20: Select the top-ranked features: S = S_rank(1:K_selected) 21: Reduce CV training and validation data: X_CVTrain_sel = X_CVTrain(:, S) X_CVVal_sel = X_CVVal(:, S) // Classification 22: Train ECOC-SVM using: X_CVTrain_sel, Y_CVTrain 23: Evaluate the classifier on: X_CVVal_sel, Y_CVVal 24: Store the validation accuracy 25: end for // Fitness Evaluation 26: Compute model error: Model_Error = 1 – mean(CV accuracy) 27: Compute fitness: Fitness_i = Model_Error + φ × (Len(S)/size(X_train, 2))	// Final Feature Weighting and Selection 54: Compute the final Adaptive-NCA feature weights using the optimized gBestPosition on X_train, Y_train: BestWeights = AdaptiveNCAWeights(X_train, Y_train, gBestPosition) 55: Normalize BestWeights. BestWeights = BestWeights / Σ BestWeights 56: Sort final feature weights in descending order: [W_sorted, S_rank] = Sort(BestWeights, descending) 57: Normalize sorted weights: W_n = W_sorted / Σ W_sorted 58: Compute cumulative contribution: CW = cumsum(W_n) 59: Determine the final number of selected features: BestK = min{m: CW(m) ≥ ρ} 60: Select the final top-ranked features: SelectedFeatures = S_rank(1:BestK) 61: Return: BestWeights, BestK, SelectedFeatures, gBestPosition
--	--

In QPSO, for each particle, the local attractor position $P_{i,d}(t)$ is computed using the individual best position $Pbest_{i,d}$ and the global best position $Gbest_d$. The formulation of $P_{i,d}$ is given in (13).

$$P_{i,d}(t) = \mu_{i,d}(t)Pbest_{i,d}(t) + [1 - \mu_{i,d}(t)]Gbest_d(t) \quad (13)$$

where μ is a random coefficient uniformly generated in the interval $([0, 1])$. Subsequently, the mean best position of all particles, denoted as $mbest$, is computed as the average of the personal best positions of the swarm. The formulation of $mbest$ is given in (14).

$$mbest_d(t) = \frac{1}{N} \sum_{i=1}^N Pbest_{i,d}(t) \quad (14)$$

where $mbest_d(t)$ denotes the mean best position, N is the population size, and $Pbest_{i,d}(t)$ represents the personal best position of the i -th particle. Here, i , d , and t denote the particle, dimension, and iteration indices, respectively. The updated particle position is then calculated using the quantum-based update equation in (15).

$$x_{i,d}(t+1) = P_{i,d}(t) \pm k(t) |mbest_d(t) - x_{i,d}(t)| \ln\left(\frac{1}{u_{i,d}(t)}\right) \quad (15)$$

where $x_{i,d}(t)$ is the particle position, $p_{i,d}(t)$ is the local attractor, and $mbest_d(t)$ is the mean best position. $k(t)$ is the contraction-expansion coefficient, u is a random variable in $[0, 1]$, and i , d , t denote the particle, dimension, and iteration indices. The QPSO algorithm fitness function is defined in (16).

$$Fitness = Model_{Error} + \varphi \left(\frac{idx}{size(X, 2)} \right) \quad (16)$$

where φ denotes the feature-selection penalty coefficient, idx denotes the number of selected features, and X denotes the training feature matrix. The model error is defined as the complement of the mean validation accuracy, as given in (17).

$$Model_{Error} = 1 - mean(accuracy) \quad (17)$$

where $Model_{Error}$ denotes the model error and accuracy denotes the validation accuracy obtained during cross-validation.

4.4 Confusion Matrix and Performance Metrics

Model performance was evaluated using the confusion matrix and the corresponding Accuracy, Precision, Recall, and F1-score metrics [19–23]. For the three-class classification problem, a one-vs.-rest strategy was adopted to compute the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values for each class individually. An example confusion matrix for the Broken cell class is presented in Table 2. The mathematical formulations of the evaluation metrics are given in Eqs. (18)–(21).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (18)$$

$$Precision = \frac{TP}{TP + FP} \quad (19)$$

$$Recall = \frac{TP}{TP + FN} \quad (20)$$

$$F1 - Score = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (21)$$

Table 2: Confusion matrix for broken cell class.

		Prediction Class		
		Broken	Dark Area	Finger Defect
True Class	Broken Cell	TP	FN	FN
	Dark Area	FP	TN	TN
	Finger Defect	FP	TN	TN

5 Results

In this study, defect classification of photovoltaic (PV) cells was performed. From the publicly available ELPV dataset [6], defective images were manually re-annotated and categorized into three classes: Broken Cell, Dark Area, and Finger Defect, thereby creating a customized sub-dataset. These defect types were selected because of their frequent occurrence in photovoltaic cells and their potential impact on energy conversion efficiency. Due to the limited number of samples in each defect class, the resulting dataset of 596 images was initially divided into training and testing subsets, with 483 images (81%) allocated to the training set and 113 images (19%) reserved for testing. Since the publicly available ELPV dataset does not provide module-level identification or grouping information, a reliable module-level split was not possible, and the partitioning was therefore performed at the image level. Consequently, the possibility that visually similar images originating from the same module may be present in both the training and test sets cannot be completely excluded. To minimize the potential risk of data leakage, the test set was separated prior to model development and was kept isolated from all subsequent data augmentation, feature selection, feature weighting, optimization, and model training procedures. Data augmentation was applied exclusively to the training set, while the test set remained unchanged for the final evaluation. Thus, the test set represents a held-out image subset rather than a strictly module-independent test set, and this limitation is considered when interpreting the generalization performance of the proposed method. The detailed dataset distribution and processing steps are presented in Table 3.

Table 3: Distribution of training and test images.

Class Name	Number of Train Images	Train Rate (%)	Number of Test Images	Test Rate (%)
Broken Cell	189	82.5	40	17.5
Dark Area	71	68.3	33	31.7
Finger Defect	223	84.8	40	15.2
Total	483	81	113	19

The experiments were conducted on a desktop computer equipped with an Intel® Core™ i5-8400 CPU @ 2.80 GHz (6 CPUs), an NVIDIA GeForce GTX 1080 Ti GPU with CUDA® cores, and a Windows 10 64-bit operating system. Prior to applying the proposed QPSO-based Adaptive-NCA framework, the newly constructed dataset was evaluated using several conventional machine-learning algorithms, and the corresponding results are presented in Table 4. This allowed a direct comparison between the baseline models and the proposed approach. The highest test accuracy obtained before applying the proposed method was 72.5% using a neural network classifier, whereas the proposed QPSO-based Adaptive-NCA approach improved the overall test accuracy to 96.46%, demonstrating a significant performance gain. The proposed Adaptive-NCA-based approach demonstrated a substantial performance improvement compared to conventional machine learning methods.

Although the baseline classifiers achieved high accuracy on the training data, they exhibited a noticeable performance drop on the test data. This behavior indicates an overfitting tendency caused by the limited size of the dataset and the high-dimensional nature of the hybrid feature space. In contrast, the proposed Adaptive-NCA method significantly improved the predictive performance on the held-out image set.

To ensure a fair, reproducible, and scientifically sound comparison, all pretrained CNN models were trained under an identical experimental protocol, as summarized in Table 5. Each network was initialized with ImageNet pretrained weights and trained using the same training, validation, and test partitions, identical data augmentation procedures, and identical hyperparameter settings. The original fully connected

and classification output layers were replaced with new layers corresponding to the three defect classes, while the remaining network parameters were fine-tuned using the Stochastic Gradient Descent with Momentum (SGDM) optimizer. Training was performed with an initial learning rate of 0.001, a mini-batch size of 16, and 30 epochs. The same hold-out validation strategy and evaluation metrics were applied to all pretrained CNN models.

Table 4: Baseline classification performance prior to applying the proposed framework.

Methods	Accuracy Train (%)	Accuracy Test (%)
Neural Network	91.8	72.5
SVM	91.4	63.3
KNN	88.0	51.7
TREE	82.4	52.5
Ensemble	76.9	60.8

Table 5: Training configurations of the evaluated pretrained CNN models.

Parameter	Configuration
Pretrained models	ResNet18, MobileNetV2, GoogLeNet, ResNet50, DenseNet201, VGG16, EfficientNet-B0, ConvNeXt-Tiny, ViT-B16, Swin-Tiny
Pretrained weights	ImageNet
Dataset	Same ELPV sub-dataset training and test sets
Data augmentation	Same augmentation strategy applied to all models
Input image size	$224 \times 224 \times 3$
Network modification	Original fully connected, softmax, and classification layers replaced with new layers for three defect classes
Fine-tuning strategy	All layers fine-tuned
Initial learning rate	0.001
Maximum epochs	30
Optimizer	SGDM
Mini-batch size	16
Shuffle	Every epoch
Performance metrics	Accuracy, Precision, Recall, F1-score, ROC-AUC, Cohen's Kappa

For the conventional machine learning classifiers, commonly adopted parameter settings and MATLAB default implementations were used instead of extensive hyperparameter optimization. Specifically, the ECOC-SVM classifier employed a Radial Basis Function (RBF) kernel with automatic kernel scaling and feature standardization, the KNN classifier used five nearest neighbors, the Random Forest classifier consisted of 100 decision trees, and the Bagged Trees and Subspace KNN classifiers were implemented using MATLAB default ensemble settings.

The conventional classifiers were employed solely as baseline models to provide a comparative evaluation of the proposed framework. All baseline classifiers were implemented using MATLAB's built-in functions with standard or default parameter configurations. No exhaustive grid-search or separate hyperparameter optimization was performed for these classifiers. The optimization procedure in this study

refers specifically to the proposed QPSO-based Adaptive-NCA framework, in which QPSO optimizes the weighting coefficients α , β , and γ .

5.1 NCA and QPSO-Based Adaptive-NCA Comparison Results

In this section, the proposed feature selection method is compared with the conventional NCA approach. The comparative results in terms of classification performance and other evaluation metrics are presented for each feature selection strategy. Since the dataset used in this study is imbalanced in terms of sample distribution, and the cost of misclassifying defective photovoltaic panels during installation is high, the model performance is evaluated not only based on accuracy but also by considering additional evaluation metrics. This comprehensive evaluation ensures a more reliable and robust assessment of the proposed method under real-world industrial conditions.

Fig. 8 illustrates the feature selection algorithms used in this study, along with the percentage distribution of the number of selected features.

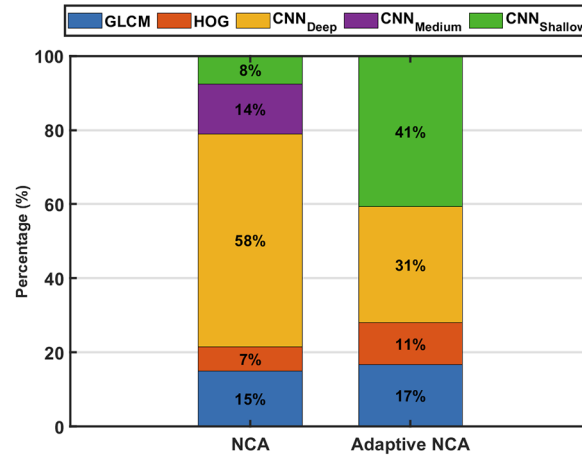


Figure 8: Distribution of selected feature subsets obtained using conventional NCA and the proposed Adaptive-NCA methods.

The number of extracted and selected features for each method is summarized in Table 6. A total of 1528 features were extracted, with 200 selected by conventional NCA and 150 by Adaptive-NCA. Despite using approximately 25% fewer features, Adaptive-NCA achieved higher classification performance, demonstrating its ability to identify more discriminative feature subsets. Notably, no CNN-medium features were selected by Adaptive-NCA, indicating their relatively lower discriminative capability.

Table 6: Number of extracted features and features retained by NCA and the proposed Adaptive-NCA.

Method Name	Number of Extracted Features	Number of Selected Features with NCA	Number of Selected Features with A-NCA
HOG	324	13	17
GLCM	304	30	25
CNN-Shallow	300	15	61
CNN-Medium	300	27	0
CNN-Deep	300	115	47
Total	1528	200	150

A lower number of selected features contributes to faster classification performance, particularly in large-scale datasets where high computational cost and memory requirements are critical limitations. In such scenarios, achieving high classification accuracy with a reduced feature set significantly improves the efficiency and effectiveness of the proposed method. Within the Adaptive-NCA framework, features are first ranked using the hybrid weighting strategy and then selected accordingly. The proposed method provides the TP, TN, FP, and FN values for each class, together with the corresponding recall values.

Comparative analysis demonstrates that the Adaptive-NCA approach consistently outperforms the conventional NCA across all classes. For photovoltaic (PV) systems, where production costs are relatively high, minimizing misclassification is of critical importance. Fig. 9a shows the classification results obtained using the NCA-based feature selection, whereas Fig. 9b presents the results obtained using the Adaptive-NCA approach in terms of correctly and incorrectly classified samples for each class. As shown in Fig. 9b, out of 33 Dark Area images, 1 sample was misclassified; out of 40 broken cell images, 2 samples were misclassified; and out of 40 Finger Defect images, 1 sample was misclassified. In applications where misclassification is critical, evaluation metrics such as precision, recall, and F1-score are also taken into account.

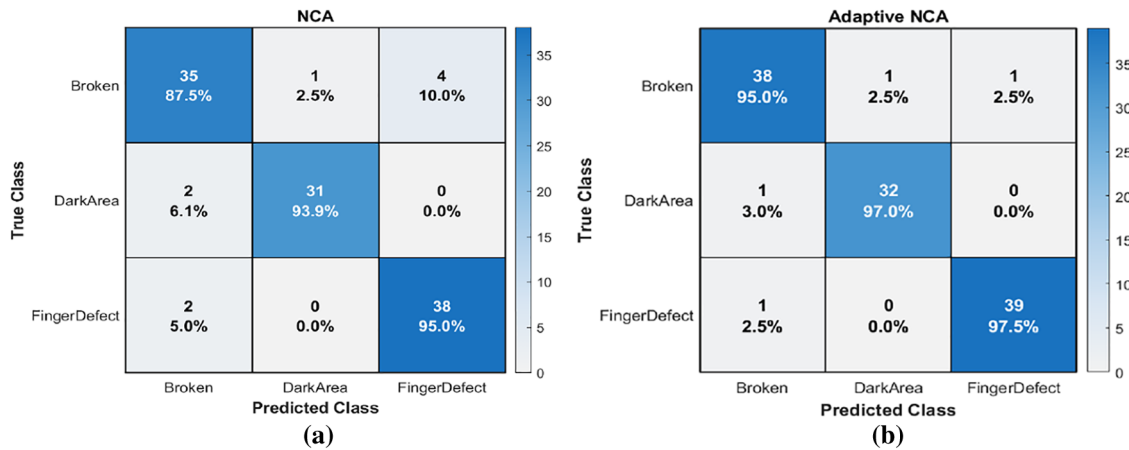


Figure 9: Comparison of confusion matrices obtained using (a) Conventional NCA and (b) The proposed QPSO-based Adaptive-NCA.

Fig. 10 illustrates the feature weights and ranking order obtained using Adaptive-NCA prior to feature selection. The x -axis represents the feature rank, while the y -axis shows the corresponding feature weights. The ranking highlights the relative importance of the extracted features and provides the basis for selecting the top 150 features.

As presented in Tables 7 and 8, the proposed Adaptive-NCA-based feature selection approach outperforms conventional NCA across the evaluated performance metrics. Despite using fewer features, the proposed method achieves strong and balanced classification performance on the held-out test set. The class-wise results further demonstrate consistent performance across the three defect classes. These findings indicate that the proposed hybrid weighting strategy provides an effective feature selection framework for solar cell defect classification. The macro class-wise accuracy reported in Tables 7 and 8 corresponds to the mean of the one-vs.-rest class accuracies and should not be confused with the standard overall multiclass test accuracy of 0.9646.

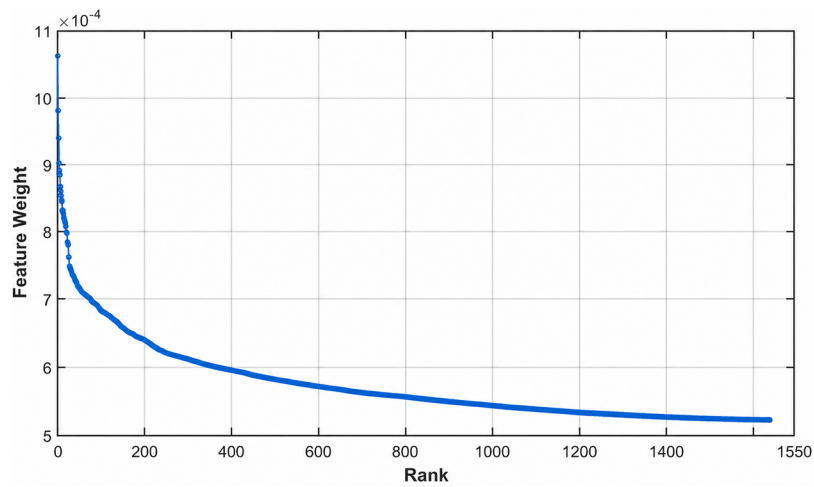


Figure 10: Ranked importance scores of features selected by the proposed Adaptive-NCA framework.

Table 7: Comparison of feature selection methods NCA and QPSO-based Adaptive-NCA.

Algorithm	Macro Train Accuracy	Macro One-vs.-Rest Test Accuracy	Macro Precision	Macro Recall	Macro F1-Score	Macro ROC-AUC
NCA	0.9856	0.9215	0.9237	0.9215	0.9223	0.9850
Adaptive-NCA	0.9729	0.9764	0.9649	0.9649	0.9649	0.9900

Table 8: Comparison of class results feature selection methods NCA and QPSO-based Adaptive-NCA.

Class	Test Accuracy		Precision		Recall		F1-Score	
	A-NCA	NCA	A-NCA	NCA	A-NCA	NCA	A-NCA	NCA
Dark Area	0.9823	0.8750	0.9697	0.9688	0.9697	0.9394	0.9697	0.9538
Broken Cell	0.9646	0.9394	0.9500	0.8974	0.9500	0.8750	0.9500	0.8861
Finger Defect	0.9823	0.9500	0.9750	0.9048	0.9750	0.9500	0.9750	0.9268
Macro Average	0.9764	0.9215	0.9649	0.9237	0.9649	0.9215	0.9649	0.9223

Fig. 11a presents the Receiver Operating Characteristic (ROC) curves for each method, which represent the discriminative capability of the models. The ROC curve's x -axis represents the False Positive Rate (FPR), while the y -axis represents the True Positive Rate (TPR), also known as Recall. These metrics are particularly useful in cases where class distributions are imbalanced, providing a more realistic evaluation of model performance. Since the dataset used in this study exhibits class imbalance, ROC analysis is employed as a reliable indicator of the overall classification performance. The feature selection results obtained using NCA and Adaptive-NCA, together with the corresponding classification performance of the model, are presented in Fig. 11b.

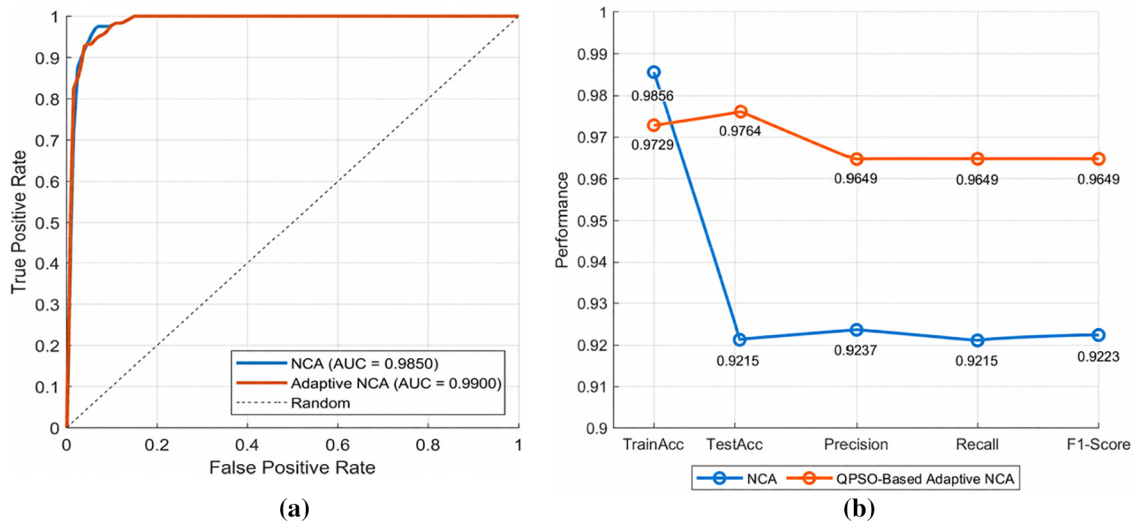


Figure 11: (a) Receiver operating characteristic (ROC) curves comparing conventional NCA and the proposed Adaptive-NCA methods; (b) Performance comparison between conventional NCA and the proposed Adaptive-NCA feature selection approaches.

Eq. (22) defines the improvement ratio achieved by the proposed Adaptive-NCA method. Overall, the Adaptive-NCA approach yielded approximately a 4.8% relative improvement in overall test accuracy compared to the conventional NCA.

$$\text{Relative Improvement Ratio} = \frac{0.9646 - 0.9204}{0.9204} \times 100 \quad (22)$$

To evaluate the statistical reliability of the proposed Adaptive-NCA framework, a bootstrap resampling analysis with 1000 iterations was conducted on the held-out test set. The direct evaluation on the held-out test set yielded an accuracy of 96.46%, precision of 96.49%, recall of 96.49%, F1-score of 96.49%, and Receiver Operating Characteristic–Area Under the Curve (ROC-AUC) of 99.0%, whereas the bootstrap analysis reported the corresponding mean values of $96.46 \pm 1.76\%$, $96.49 \pm 1.77\%$, $96.47 \pm 1.78\%$, $96.43 \pm 1.78\%$, and $99.26 \pm 0.06\%$, respectively. The bootstrap estimates represent the mean performance across the 1000 resampled test sets, and the reported 95% confidence intervals were calculated using the percentile bootstrap method based on the 2.5th and 97.5th percentiles of the bootstrap distributions. Accordingly, the bootstrap estimates may differ slightly from the single direct evaluation because they summarize performance across repeated resamples of the same held-out test set. Furthermore, the model consistently exhibited excellent discriminative capability, achieving a mean ROC-AUC of $99.26 \pm 0.06\%$ and a Cohen’s kappa coefficient of $94.62 \pm 2.68\%$. These results indicate that the proposed framework provides robust classification performance under repeated bootstrap resampling of the held-out test set. Table 9 presents the detailed results. Table 7 reports the results obtained directly from the held-out test set, whereas Table 9 presents the mean performance over 1000 bootstrap resamples of the same test set.

5.2 Ablation Studies

This section evaluates the contributions of the different feature extraction techniques employed in the proposed framework. In this study, the effects of three feature extraction methods are evaluated. Table 10 and Fig. 12 present the classification results obtained using the Histogram of Oriented Gradients (HOG) feature extraction method and the proposed approach. HOG is a gradient-based descriptor that captures

edge and corner information within images. Fig. 12a shows the classification performance obtained using HOG-based feature extraction. As a result, higher recall values are observed for broken cell and finger defect classes, which contain thin-line structures, compared to the dark area class. Fig. 12b presents the ROC curve corresponding to the HOG-based feature extraction method.

Table 9: Repeated bootstrap resampling of the held-out test set.

Metric	Mean $\pm$ SD	95% Confidence Interval
Accuracy	96.46 $\pm$ 1.76	92.92–99.12
Precision	96.49 $\pm$ 1.77	92.44–99.24
Recall	96.47 $\pm$ 1.78	92.47–99.24
F1-score	96.43 $\pm$ 1.78	92.35–99.18
ROC-AUC	99.26 $\pm$ 0.06	99.22–99.30
Cohen's Kappa	94.62 $\pm$ 2.68	89.02–98.67

Table 10: Class results of HOG methods.

Class	Accuracy	Precision	Recall	F1-Score
Dark Area	0.8938	0.9565	0.6667	0.7857
Broken Cell	0.8495	0.7447	0.8750	0.8046
Finger Defect	0.9558	0.9070	0.9750	0.9398
Macro Average	0.8997	0.8694	0.8389	0.8434

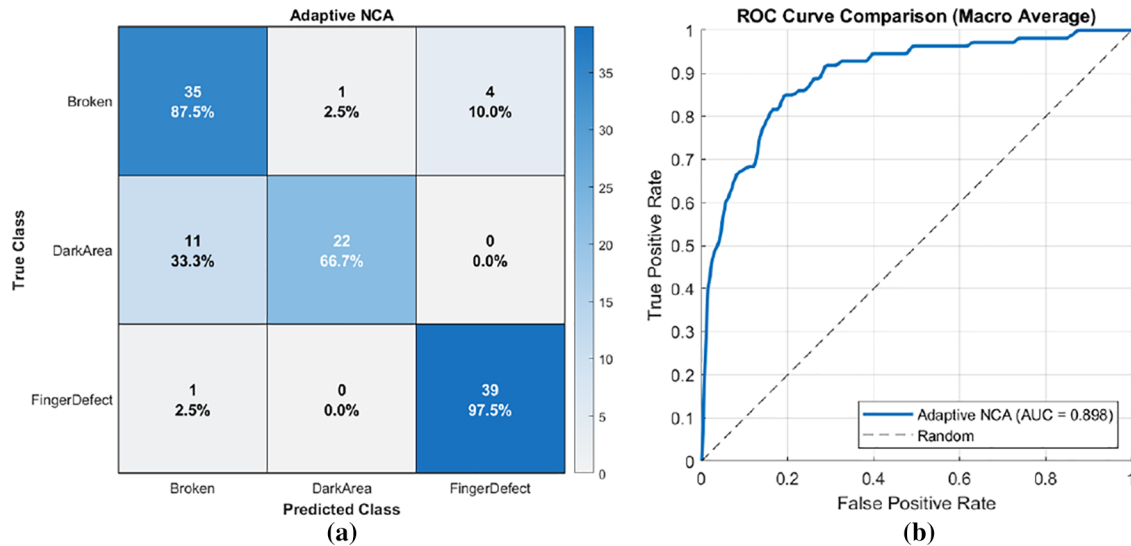


Figure 12: (a) Classification performance obtained using HOG-based feature extraction; (b) ROC analysis of the HOG-based feature extraction approach.

Table 11 and Fig. 13a present the classification results obtained using the Gray Level Co-occurrence Matrix (GLCM) feature extraction method in conjunction with the proposed classification framework. GLCM is widely used in image processing for texture analysis; therefore, it is expected to be more sensitive

to texture-dominant regions such as the Dark Area class. These results indicate that features extracted using the GLCM method contribute more effectively to the Dark Area class. Fig. 13b presents the ROC curve of the proposed method. As observed in Fig. 13b, the obtained results show that the ROC curves deviate significantly from the reference diagonal, indicating that the model achieves effective discrimination between classes.

Table 11: Class results of GLCM methods.

Class	Accuracy	Precision	Recall	F1-Score
Dark Area	0.9735	0.9688	0.9394	0.9538
Broken Cell	0.8673	0.8049	0.8250	0.8148
Finger Defect	0.8938	0.8500	0.8500	0.8500
Macro Average	0.9115	0.8746	0.8715	0.8729

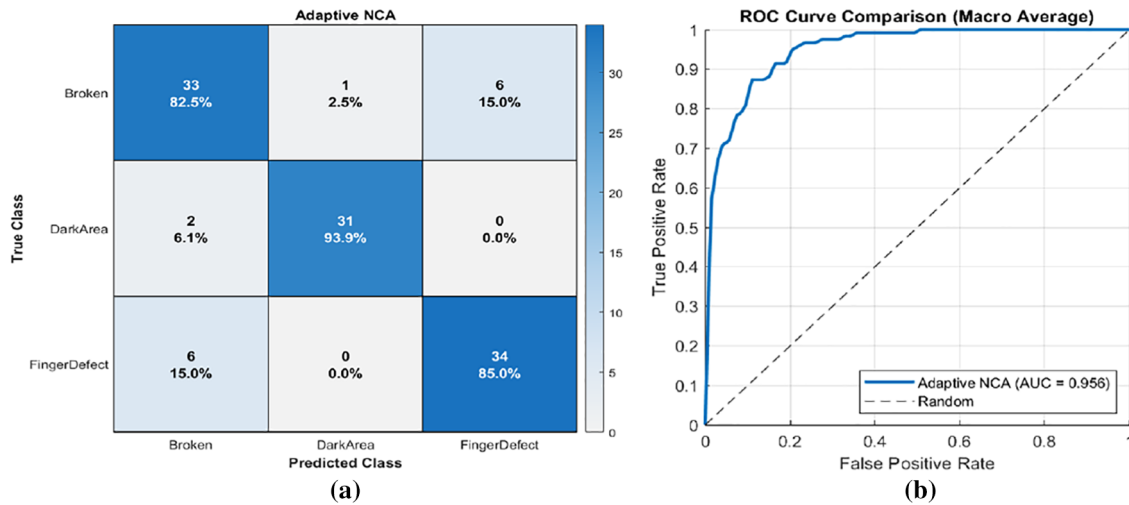


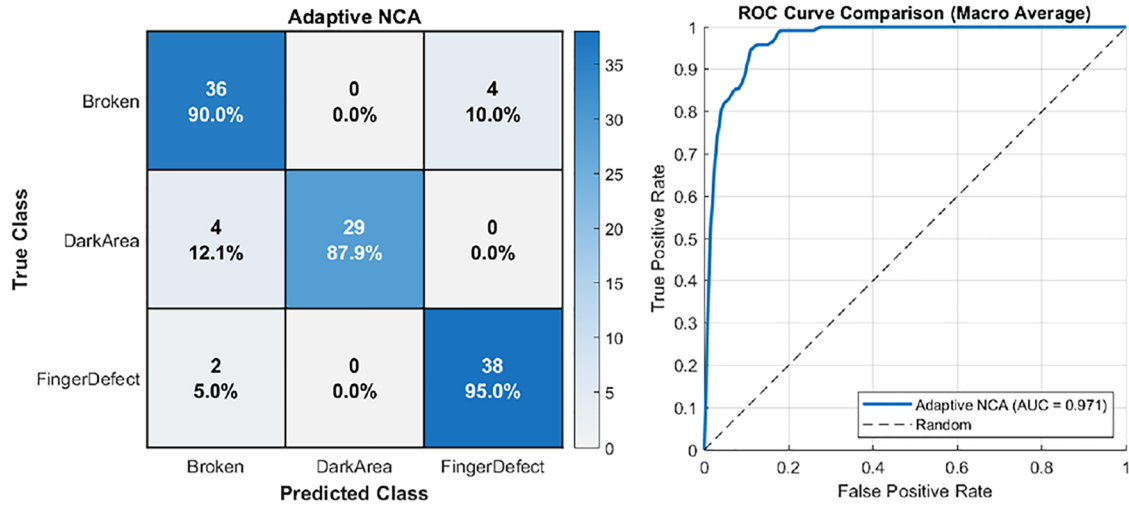
Figure 13: (a) Classification performance achieved using GLCM-based feature extraction; (b) ROC curves obtained using the GLCM-based feature extraction method.

Table 12 and Fig. 14a present the classification results obtained using the Convolutional Neural Network (CNN)-based feature extraction method in conjunction with the proposed framework. In CNN-based feature extraction, features are extracted from different network depths. Specifically, shallow layers capture low-level features, intermediate layers extract mid-level representations, and deep layers encode high-level semantic information. Shallow-layer features primarily consist of basic structures such as edges, corners, and lines. Intermediate layers capture texture patterns and shape variations, while deep layers learn more abstract and semantically meaningful representations. Fig. 14b presents the ROC curve of the CNN Feature extraction method.

A total of 900 CNN features were extracted by combining the shallow, intermediate, and deep layers. Table 12 reports the results obtained using these combined features. The average performance metrics are reported in Table 12, while the corresponding class-wise performance metrics and macro-average values are reported in Table 13.

Table 12: Adaptive-NCA-SVM classification results of CNN methods.

Method	Macro Train Accuracy	Macro Test Accuracy	Macro Precision	Macro Recall	Macro F1-Score
Adaptive-NCA-SVM	0.9857	0.9410	0.9206	0.9096	0.9134

**Figure 14:** (a) Classification performance achieved using CNN-based feature extraction; (b) ROC analysis of the CNN-based feature extraction framework.**Table 13:** Class results of CNN methods.

Class	Test Accuracy	Precision	Recall	F1-Score
Dark Area	0.9646	1.0000	0.8788	0.9355
Broken Cell	0.9115	0.8571	0.9000	0.8780
Finger Defect	0.9469	0.9048	0.9500	0.9268
Macro Average	0.9410	0.9206	0.9096	0.9134

For the CNN-based feature extraction approach, the training accuracy was obtained as 0.9857, whereas the macro test accuracy was 0.9410. The approximately 4.5% gap between training and testing accuracy indicates that the CNN features exhibit a tendency toward overfitting to the dataset. This limitation becomes more pronounced in studies with limited training data and may negatively affect generalization performance.

Therefore, instead of relying solely on CNN-based feature extraction, the proposed approach combines HOG, GLCM, and CNN features, which capture information at different levels. In this unified feature space, local edge information, texture descriptors, and high-level representations are jointly evaluated.

Table 14 summarizes the execution time of the individual computational stages of the proposed framework. The reported per-image times represent the average processing time of each individual stage and should not be interpreted as end-to-end inference time. Table 15 presents the total execution time of each stage of the proposed framework. Table 16 summarizes the contribution of the different techniques

used in the proposed study. The CNN-based feature extraction process requires significantly more computational time compared to the other methods. This increases both time and computational resource costs, particularly in large-scale datasets where efficiency becomes a critical factor. As shown in Table 14, the combined CNN-based feature extraction process required approximately 250 times more computational time than HOG. This result demonstrates that traditional methods still offer significant advantages in resource-constrained applications.

Table 14: Runtime of the computational stages of the proposed framework.

Algorithm	Total Runtime (Seconds)	Average Processing Time per Image (Seconds)
HOG	24.27	0.011
GLCM	589.33	0.269
CNN-Shallow	1906.12	0.872
CNN-Medium	2007.02	0.918
CNN-Deep	2254.97	1.031
QPSO	765	0.369
Adaptive-NCA	71	0.034
SVM Training	5	0.002
Total	7622.71	–

Table 15: Execution time of each stage of the proposed framework.

Algorithm	Average Processing Time per Image (Seconds)	Total Run Time (Seconds)	Image Number
Feature Extraction	3.101	6781.71	2187
Feature Selection	0.403	836	2074
Model Training	0.002	5	2074

Table 16: Ablation study of individual and hybrid feature representations.

Feature Representation	Feature Selection/Weighting	Overall Train Accuracy	Overall Test Accuracy	Precision	Recall	F1-Score	Cohen's Kappa
HOG	QPSO-based Adaptive-NCA	0.9921	0.8496	0.8694	0.8389	0.8434	0.7716
GLCM	QPSO-based Adaptive-NCA	0.9198	0.8673	0.8745	0.8715	0.8729	0.8000
CNN-Shallow	QPSO-based Adaptive-NCA	0.8893	0.7434	0.7585	0.7301	0.7308	0.6097
CNN-Medium	QPSO-based Adaptive-NCA	0.9531	0.8407	0.8599	0.8376	0.8433	0.7591
CNN-Deep	QPSO-based Adaptive-NCA	0.9857	0.9292	0.9357	0.9280	0.9307	0.8931
Hybrid (HOG+GLCM+CNNs)	QPSO-based Adaptive-NCA	0.9729	0.9646	0.9649	0.9649	0.9649	0.9467

The computation times reported in Table 14 correspond to the execution time of each computational stage evaluated independently rather than the runtime of a single end-to-end inference process. In the proposed framework, shallow, medium, and deep CNN features were extracted separately from different layers of the pretrained ResNet18 network, requiring an independent forward pass for each representation. Since deeper representations involve more convolutional layers, their computation time naturally increases with network depth. The feature extraction time was measured over all 2187 images, including 2074 training

and 113 held-out test images, whereas feature selection and model training were performed only on the 2074 training images.

It should be noted that CNN-based feature extraction is performed only once during the offline feature generation stage. The primary objective of the proposed DDSWF framework is not to reduce the computational cost of CNN feature extraction itself, but to decrease the computational burden of the subsequent learning process through Adaptive-NCA-based dimensionality reduction. By reducing the feature dimensionality from 1528 to 150 while preserving highly discriminative information, the proposed framework substantially reduces the computational cost of feature selection, classifier training, and classification, while eliminating the need for repeated end-to-end deep network optimization.

The total runtime represents the sum of the stage-wise runtimes. An overall average processing time per image is not reported because the computational stages use different image counts.

Table 16 presents the ablation results for the individual and hybrid feature representations. HOG, GLCM, CNN-Shallow, CNN-Medium, and CNN-Deep were evaluated individually using the same QPSO-based Adaptive-NCA and SVM framework, and their results were compared with the combined hybrid representation. These results demonstrate the benefit of combining complementary handcrafted and deep feature representations.

Table 17 presents the ablation results for the feature selection and weighting components of the proposed framework. Using the same hybrid feature representation and SVM classifier, the overall accuracy increased from 92.04% with conventional NCA to 93.81% with Adaptive-NCA using fixed equal weighting ($\alpha = \beta = \gamma = 1/3$). When the weighting coefficients were optimized using QPSO, the proposed Adaptive-NCA framework further increased the overall test accuracy to 96.46%. These results demonstrate the incremental contributions of the Adaptive-NCA weighting strategy and the subsequent QPSO-based coefficient optimization to the classification performance of the proposed framework.

Table 17: Ablation study of the feature selection and weighting components of the proposed framework.

Methods	Weighting/Feature Selection	Overall Test Accuracy	Precision	Recall	F1-Score
NCA + SVM	NCA	0.9204	0.9237	0.9215	0.9223
Adaptive-NCA + SVM	Fixed $\alpha = \beta = \gamma = 1/3$	0.9381	0.9399	0.9399	0.9398
QPSO-based Adaptive-NCA-SVM (This study)	QPSO-based Weighting	0.9646	0.9649	0.9649	0.9649

The results in Tables 6, 14–17 demonstrate that the performance improvement is attributable not only to the combination of multiple feature extraction methods but also to the proposed Adaptive-NCA feature selection strategy. From the initial 1528 features, Adaptive-NCA selects only 150 discriminative features while maintaining and improving classification performance. The selected features comprise 61 shallow CNN, 47 deep CNN, 25 GLCM, and 17 HOG features, whereas no CNN-Medium features are retained. This distribution indicates that shallow and deep CNN features provide complementary local and high-level representations, while HOG and GLCM contribute structural and statistical texture information. The exclusion of CNN-Medium features suggests that these intermediate representations provide limited additional discriminative information and may be redundant with the other feature groups.

In addition, QPSO optimizes the Adaptive-NCA weighting coefficients rather than relying on manually predefined weights. Thus, the performance gains reported in Tables 16 and 17 result from both the complementary hybrid feature representation and the adaptive feature selection and weighting strategy.

5.3 Comparison with Other Methods and Related Studies

5.3.1 Comparison of the Proposed Hybrid Model with Transfer Learning Methods

In this section, the proposed hybrid framework is compared with representative transfer-learning and modern deep-learning architectures under the same experimental protocol. The proposed method combines hybrid handcrafted and deep feature extraction, Adaptive-NCA-based feature selection, QPSO-based weighting optimization, and SVM classification. As reported in Table 18, the proposed framework achieved the highest overall test accuracy of 96%, while ConvNeXt-Tiny and Swin-Tiny achieved 95%, followed by EfficientNet-B0 and ViT-B/16 with 94%.

Table 18: Performance comparison of the proposed method with pretrained deep-learning architectures.

Methods	Overall Accuracy	Precision	Recall	F1-Score
MobileNet V2	0.92	0.93	0.92	0.92
GoogLeNet	0.93	0.88	0.96	0.92
ResNet50	0.92	0.89	0.91	0.90
VGG16	0.93	0.94	0.93	0.93
DenseNet201	0.85	0.78	0.85	0.81
EfficientNet-B0	0.94	0.95	0.93	0.94
ConvNeXt-Tiny	0.95	0.95	0.94	0.95
ViT-B16	0.94	0.94	0.94	0.94
Swin-Tiny	0.95	0.96	0.95	0.95
QPSO-based Adaptive-NCA-SVM (This study)	0.96	0.96	0.96	0.96

The comparison included MobileNetV2 (Mobile Network Version 2), GoogLeNet, ResNet-50 (Residual Network), VGG16 (Visual Geometry Group), DenseNet-201 (Densely Connected Convolutional Network), EfficientNet-B0, ConvNeXt-Tiny, ViT-B/16 (Vision Transformer-Base), and Swin Transformer-Tiny, covering a diverse range of lightweight, convolutional, densely connected, residual, and transformer-based architectures. Although these architectures have demonstrated strong performance in general vision tasks, their pre-training on large-scale natural image datasets such as ImageNet may introduce a domain gap when applied to photovoltaic electroluminescence (EL) images, particularly under limited-data conditions.

Unlike the end-to-end fine-tuning of deep-learning models, the proposed framework integrates handcrafted and deep features with Adaptive-NCA-based feature selection and QPSO-based weighting optimization. The superior accuracy achieved by the proposed approach indicates that it provides a competitive alternative to modern deep-learning architectures for fine-grained photovoltaic defect classification under the considered limited-data setting.

All transfer learning models were trained and evaluated under identical experimental conditions to ensure a fair and reproducible comparison. The experiments were conducted using the ELPV sub-dataset, which was partitioned into 81% training and 19% testing subsets using the same class-aware splitting strategy. All input images were resized to 224×224 pixels before being fed into the pretrained networks. Model training was performed using the SGDM optimizer with a learning rate of 0.001, a momentum coefficient of 0.9, a mini-batch size of 16, and 30 training epochs. The same data augmentation strategy was consistently applied across all transfer learning models. Model performance was assessed using accuracy, precision, recall, and F1-score.

5.3.2 Comparison of Adaptive-NCA with Other Feature Selection Methods

In this section, the proposed Adaptive-NCA feature selection method is compared with other commonly used feature selection techniques in the literature. In the experimental setup, feature selection is first applied using different algorithms, followed by classification using an SVM classifier. The obtained results are summarized in Table 19. The comparative results presented in Table 19 demonstrate that the proposed Adaptive-NCA method achieves superior performance compared to traditional feature selection techniques. Specifically, ReliefF (Relief Feature Selection), MRMR (Minimum Redundancy Maximum Relevance), Chi-square (χ^2) test, and ANOVA (Analysis of Variance) methods yield relatively lower classification accuracies ranging between 0.82 and 0.83, indicating limited discriminative capability for the given dataset. The conventional NCA method improves performance to 0.92; however, it still remains below the proposed approach. In contrast, Adaptive-NCA achieves the highest overall test accuracy of 0.96, outperforming all other methods.

Table 19: Comparison of Adaptive-NCA with conventional feature selection methods.

Methods	Overall Accuracy	Classification Algorithm
ReliefF	0.82	SVM
MRMR	0.83	SVM
Chi2	0.83	SVM
Anova	0.82	SVM
NCA	0.92	SVM
Adaptive-NCA	0.96	SVM

5.3.3 Comparison of the Proposed Model with Similar Studies in the Literature

Table 20 provides an overview of representative studies using the ELPV dataset. These studies employ different dataset configurations, class definitions, and classification tasks; therefore, their reported performance values are presented for contextual reference rather than as directly comparable benchmarks. For example, Akram et al. reported 0.93 accuracy for binary normal-vs.-defective classification, while Wang et al. reported 0.92 for multi-class defect detection task and Mahmoud et al. reported 0.98 for a coarse-grained multi-class classification. In contrast, the present study focuses on fine-grained three-class classification of finger defect, broken cell, and dark area using a manually re-annotated subset of 596 ELPV images. Accordingly, the results in Table 20 are not intended to establish a direct performance ranking among the reviewed approaches.

Table 20: Overview of related studies on the ELPV dataset and electroluminescence (EL) image-based classification.

Study	Method	Data Set	Classification Task	Image Number	Overall Accuracy	Precision	Recall	F1-Score
Akram et al. (2019)	CNN-based defect detection [8].	ELPV	Binary: Normal vs. Defect	2624	0.93	0.92	0.93	0.92
Chen et al. (2019)	Structure-aware-based crack defect detection scheme (SACDDS) [10].	EL Images	–	10,000	0.96	–	–	0.95
Wang et al. (2023)	ResNet152-Xception-CA [16].	ELPV	Multiclass	2624	0.92	0.89	0.90	0.89

(Continued)

Table 20 (continued)

Study	Method	Data Set	Classification Task	Image Number	Overall Accuracy	Precision	Recall	F1-Score
Rohith et al. (2025)	CNN + Vision Transformer-based feature extraction [20].	Real-world solar panel image dataset	–		0.95	0.95	0.95	0.95
Munawer Al-Hotum (2023)	ResNet18 + Shufflenet, Independent light-depth CNN (CNN-ILD) [21].	ELPV	Four-class severity(0.33-0.66-1.00-0.00)	2624	0.96	0.85	0.85	0.87
Dhimish (2021)	Quadratics Discriminant Classifier (QDC) [22].	ELPV	Four-class (Normal, Defect-1, Defect-2, Defect-3)	2624	0.98	–	–	–
This Study	QPSO-based Adaptive-NCA + SVM	ELPV	Three-class fine-grained defect type (Finger Defect, Broken Cell, Dark Area)	596	0.9646	0.9649	0.9649	0.9649

6 Conclusions

In this study, a hybrid framework, referred to as Deep Defect-Specific Weighted Features (DDSWF), was proposed for the accurate classification of critical defect types in electroluminescence (EL) images. While the original ELPV dataset categorizes photovoltaic cells according to defect severity levels (0%, 33%, 66%, and 100%), the proposed study constructed a defect-oriented sub-dataset through expert-guided re-annotation in accordance with the IEC TS 60904-13 standard. The resulting sub-dataset consists of three industrially significant defect categories, namely broken cell, dark area, and finger defect, thereby enabling a more meaningful defect-type classification for photovoltaic manufacturing processes.

An important limitation of this study concerns the image-level data split. Since module-level grouping information is unavailable in the ELPV dataset, cells from the same module may appear in both subsets, potentially leading to optimistic performance estimates. Therefore, the reported results should be interpreted as image-level holdout performance. Future studies will employ module-level splitting and external datasets to provide a more robust assessment of cross-module generalization.

The proposed DDSWF framework integrates texture features extracted using the Gray-Level Co-occurrence Matrix (GLCM), shape features derived from the Histogram of Oriented Gradients (HOG), and high-level semantic representations learned by Convolutional Neural Networks (CNNs) into a unified hybrid feature set. To improve feature discriminability and eliminate redundant information, a novel Adaptive Neighborhood Component Analysis (Adaptive-NCA) method was developed, whose parameters were optimized using the Quantum-Behaved Particle Swarm Optimization (QPSO) algorithm. By jointly considering data distribution characteristics, classifier performance feedback, and Fisher discriminative information, the proposed method effectively preserves complementary information extracted from heterogeneous feature extraction techniques, resulting in more effective feature selection under limited-data conditions.

Experimental results demonstrated that the proposed framework achieved higher classification performance than the evaluated handcrafted-feature baselines and pretrained deep-learning models under the experimental conditions considered in this study. Using a Support Vector Machine (SVM) classifier, the DDSWF framework achieved 96.46% overall test accuracy, 96.49% precision, 96.49% recall, 96.49% F1-score, and an AUC of 99.0%. Furthermore, the ablation study, ROC analysis, Cohen's kappa evaluation, and computational cost analysis further confirmed the accuracy, robustness, and practical applicability of the proposed framework.

From an industrial perspective, performing defect-type-oriented classification provides substantially more informative feedback than conventional binary defective/non-defective classification. Accurate identification of finger defect, broken cell, and dark area defects facilitates root-cause analysis, improves quality control processes, reduces production losses, and supports predictive maintenance strategies in photovoltaic manufacturing. Therefore, the proposed DDSWF framework not only serves as an effective academic classification model but also provides a promising decision-support framework for photovoltaic inspection applications. Future research will focus on extending the proposed framework to multi-defect and multi-label classification problems, as well as integrating the proposed model into real-time industrial inspection systems. These improvements are expected to further enhance the transparency of the decision-making process while increasing the reliability and applicability of intelligent photovoltaic inspection systems.

Overall, the proposed DDSWF framework demonstrates promising performance for photovoltaic defect-type classification by combining high classification accuracy with model interpretability. The proposed framework may provide a useful basis for intelligent photovoltaic quality control and inspection applications, particularly under limited-data conditions.

Acknowledgement: This work was produced from the Uğur Çiğdem's doctoral dissertation conducted at Tokat Gaziosmanpaşa University.

Funding Statement: This research received no external funding.

Author Contributions: Conceptualization, Uğur Çiğdem and Kenan Zengin; methodology, Uğur Çiğdem and Kenan Zengin; software, Uğur Çiğdem and Kenan Zengin; validation, Uğur Çiğdem and Kenan Zengin; formal analysis, Uğur Çiğdem and Kenan Zengin; investigation, Uğur Çiğdem and Kenan Zengin; resources, Uğur Çiğdem and Kenan Zengin; data curation, Uğur Çiğdem and Kenan Zengin; writing—original draft preparation, Uğur Çiğdem and Kenan Zengin; writing—review and editing, Uğur Çiğdem and Kenan Zengin; visualization, Uğur Çiğdem and Kenan Zengin; supervision, Uğur Çiğdem and Kenan Zengin; project administration, Uğur Çiğdem and Kenan Zengin; funding acquisition, Uğur Çiğdem and Kenan Zengin. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The original ELPV dataset used in this study is publicly available on GitHub at <https://github.com/zae-bayern/elpv-dataset>. A corresponding annotation file containing the identifiers of the images selected from the original ELPV dataset and their assigned defect classes in the customized 596-image subset is available from the authors upon reasonable request. The synthetic training samples generated through the data augmentation procedures are also available from the authors upon reasonable request. Researchers interested in obtaining the annotation file and/or the synthetic training samples may contact the corresponding authors by email. The data augmentation procedures used to generate the synthetic training samples, including the augmentation techniques and their application strategy, are described in detail in the Methods section of this manuscript.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript

DDSWF	Deep Defect-Specific Weighted Features
GLCM	Gray-Level Co-occurrence Matrix
HOG	Histogram of Oriented Gradients
NCA	Neighborhood Component Analysis
ELPV	Electroluminescence Photovoltaic
QPSO	Quantum-Behaved Particle Swarm Optimization
IEA	International Energy Agency
IRENA	International Renewable Energy Agency
CNN	Convolutional Neural Network
NAS	Neural Architecture Search
CLAHE	Contrast Limited Adaptive Histogram Equalization
PCA	Principal Component Analysis
GA	Genetic Algorithms
PSO	Particle Swarm Optimization
SVM	Support Vector Machine
ECOC	Error-Correcting Output Codes
ROC	Receiver Operating Characteristic
KNN	K-Nearest Neighbors

References

1. International Renewable Energy Agency (IRENA). Renewable power generation records its fastest growth ever [Internet]. [cited 2026 Jul 5]. Available from: <https://www.irena.org/News/pressreleases/2026/Jul/Renewable-Power-Generation-Records-Its-Fastest-Growth-Ever>.
2. Tang W, Yang Q, Hu X, Yan W. Convolution neural network based polycrystalline silicon photovoltaic cell linear defect diagnosis using electroluminescence images. *Expert Syst Appl*. 2022;202(1998):117087. doi:10.1016/j.eswa.2022.117087.
3. Otamendi U, Martinez I, Quartulli M, Olaizola IG, Viles E, Cambarau W. Segmentation of cell-level anomalies in electroluminescence images of photovoltaic modules. *Sol Energy*. 2021;220:914–26. doi:10.1016/j.solener.2021.03.058.
4. Tsai DM, Wu SC, Li WC. Defect detection of solar cells in electroluminescence images using Fourier image reconstruction. *Sol Energy Mater Sol Cells*. 2012;99:250–62. doi:10.1016/j.solmat.2011.12.007.
5. Anwar SA, Abdullah MZ. Micro-crack detection of multicrystalline solar cells featuring an improved anisotropic diffusion filter and image segmentation technique. *EURASIP J Image Video Process*. 2014;2014(1):15. doi:10.1186/1687-5281-2014-15.
6. Deitsch S, Christlein V, Berger S, Buerhop-Lutz C, Maier A, Gallwitz F, et al. Automatic classification of defective photovoltaic module cells in electroluminescence images. *Sol Energy*. 2019;185:455–68. doi:10.1016/j.solener.2019.02.067.
7. Karimi AM, Fada JS, Hossain MA, Yang S, Peshek TJ, Braid JL, et al. Automated pipeline for photovoltaic module electroluminescence image processing and degradation feature classification. *IEEE J Photovolt*. 2019;9(5):1324–35. doi:10.1109/JPHOTOV.2019.2920732.
8. Akram MW, Li G, Jin Y, Chen X, Zhu C, Zhao X, et al. CNN based automatic detection of photovoltaic cell defects in electroluminescence images. *Energy*. 2019;189(12):116319. doi:10.1016/j.energy.2019.116319.
9. Chen X, Karin T, Jain A. Automated defect identification in electroluminescence images of solar modules. *Sol Energy*. 2022;242(1):20–9. doi:10.1016/j.solener.2022.06.031.
10. Chen H, Zhao H, Han D, Liu K. Accurate and robust crack detection using steerable evidence filtering in electroluminescence images of solar cells. *Opt Lasers Eng*. 2019;118(1):22–33. doi:10.1016/j.optlaseng.2019.01.016.

11. Tang W, Yang Q, Yan W. Deep learning based model for defect detection of mono-crystalline-Si solar PV module cells in electroluminescence images using data augmentation. In: Proceedings of the 2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC); 2019 Dec 1–4; Macao, China. p. 1–5. doi:10.1109/APPEEC45492.2019.8994713.
12. Zhang X, Hao Y, Shangguan H, Zhang P, Wang A. Detection of surface defects on solar cells by fusing multi-channel convolution neural networks. *Infrared Phys Technol.* 2020;108:103334. doi:10.1016/j.infrared.2020.103334.
13. Zhao Y, Zhan K, Wang Z, Shen W. Deep learning-based automatic detection of multitype defects in photovoltaic modules and application in real production line. *Prog Photovolt Res Appl.* 2021;29(4):471–84. doi:10.1002/pip.3395.
14. Meng Z, Xu S, Wang L, Gong Y, Zhang X, Zhao Y. Defect object detection algorithm for electroluminescence image defects of photovoltaic modules based on deep learning. *Energy Sci Eng.* 2022;10(3):800–13. doi:10.1002/ese3.1056.
15. Liu Q, Liu M, Wang C, Wu QJ. An efficient CNN-based detector for photovoltaic module cells defect detection in electroluminescence images. *Sol Energy.* 2024;267(4):112245. doi:10.1016/j.solener.2023.112245.
16. Wang J, Bi L, Sun P, Jiao X, Ma X, Lei X, et al. Deep-learning-based automatic detection of photovoltaic cell defects in electroluminescence images. *Sensors.* 2022;23(1):297. doi:10.3390/s23010297.
17. Zhao X, Song C, Zhang H, Sun X, Zhao J. HRNet-based automatic identification of photovoltaic module defects using electroluminescence images. *Energy.* 2023;267:126605. doi:10.1016/j.energy.2022.126605.
18. Zhang J, Chen X, Wei H, Zhang K. A lightweight network for photovoltaic cell defect detection in electroluminescence images based on neural architecture search and knowledge distillation. *Appl Energy.* 2024;355(1):122184. doi:10.1016/j.apenergy.2023.122184.
19. Geng T, Li S, Yun Y, Kang Y, Li H, Zhu J. An RMD-YOLOv11 approach for typical defect detection of PV modules. *Comput Mater Continua.* 2025;1–10. doi:10.32604/cmc.2025.071644.
20. Rohith G, Rajalakshmi R, Manish DS, Rahul Narasimhan A. Fusion-Solar-Net for solar panel fault detection. *Results Eng.* 2025;27(3):106513. doi:10.1016/j.rineng.2025.106513.
21. Munawer Al-Otun H. Deep learning-based automated defect classification in electroluminescence images of solar panels. *Adv Eng Inform.* 2023;58:102147. doi:10.1016/j.aei.2023.102147.
22. Dhimish M. Defining the best-fit machine learning classifier to early diagnose photovoltaic solar cells hot-spots. *Case Stud Therm Eng.* 2021;25:100980. doi:10.1016/j.csite.2021.100980.
23. Kari T, Mahmood A, Del Prado Santamaria R, Alves dos Reis Benatto G, Poulsen PB, Spataru SV. Assessing electroluminescence image quality with machine-learning and grey-level co-occurrence matrix texture descriptors. In: Proceedings of the 41st European Photovoltaic Solar Energy Conference and Exhibition; 2024 Sep 23–27; Wien, Austria. p. 020345. doi:10.4229/EUPVSEC2024/4BV.3.30.
24. You H, Wang P, Li Z. Feature selection for label distribution learning based on the statistical distribution of data and fuzzy mutual information. *Inf Sci.* 2024;679:121085. doi:10.1016/j.ins.2024.121085.
25. Gu Q, Li Z, Han J. Generalized fisher score for feature selection. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI); 2011 Jul 14–17; Barcelona, Spain. p. 266–73.
26. Sun J, Feng B, Xu W. Particle swarm optimization with particles having quantum behavior. In: Proceedings of the 2004 Congress on Evolutionary Computation; 2004 Jun 19–23; Portland, OR, USA. p. 325–31. doi:10.1109/CEC.2004.1330875.