

Automatic Classification of Defective Photovoltaic Module Cells in Electroluminescence Images

Sergiu Deitsch^{a,b,d,*}, Vincent Christlein^d, Stephan Berger^c, Claudia Buerhop-Lutz^c, Andreas Maier^d, Florian Gallwitz^{a,b}, Christian Riess^d

^aEnergie Campus Nuremberg, Fürther Str. 250, 90429 Nuremberg, Germany

^bNuremberg Institute of Technology, Department of Computer Science, Hohfelderstr. 40, 90489 Nuremberg, Germany

^cZAE Bayern, Immerwahrstr. 2, 91058 Erlangen, Germany

^dPattern Recognition Lab, University of Erlangen-Nuremberg, Martensstr. 3, 91058 Erlangen, Germany

Abstract

Electroluminescence (EL) imaging is a useful modality for the inspection of photovoltaic (PV) modules. EL images provide high spatial resolution, which makes it possible to detect even finest defects on the surface of PV modules. However, the analysis of EL images is typically a manual process that is expensive, time-consuming, and requires expert knowledge of many different types of defects.

In this work, we investigate two approaches for automatic detection of such defects in a single image of a PV cell. The approaches differ in their hardware requirements, which are dictated by their respective application scenarios. The more hardware-efficient approach is based on hand-crafted features that are classified in a Support Vector Machine (SVM). To obtain a strong performance, we investigate and compare various processing variants. The more hardware-demanding approach uses an end-to-end deep Convolutional Neural Network (CNN) that runs on a Graphics Processing Unit (GPU). Both approaches are trained on 1,968 cells extracted from high resolution EL intensity images of mono- and polycrystalline PV modules. The CNN is more accurate, and reaches an average accuracy of 88.42%. The SVM achieves a slightly lower average accuracy of 82.44%, but can run on arbitrary hardware. Both automated approaches make continuous, highly accurate monitoring of PV cells feasible.

Keywords: Deep learning, defect classification, electroluminescence imaging, photovoltaic modules, regression analysis, support vector machines, visual inspection.

1. Introduction

Solar modules are usually protected by an aluminum frame and glass lamination from environmental influences such as rain, wind, and snow. However, these protective measures can not always prevent mechanical damages caused by dropping the PV module during installation, impact from falling tree branches, hail, or thermal stress. Also, manufacturing errors such as faulty soldering or defective wires can also result in damaged PV modules. Defects can in turn decrease the power efficiency of solar modules. Therefore, it is necessary to monitor the

condition of solar modules, and replace or repair defective units in order to ensure maximum efficiency of solar power plants.

Visual identification of defective units is particularly difficult, even for trained experts. Aside from obvious cracks in the glass, many defects that reduce the efficiency of a PV module are not visible to the eye. Conversely, defects that are visible do not necessarily reduce the module efficiency.

To precisely determine the module efficiency, the electrical output of a module must be measured directly. However, such measurements require manual interaction with individual units for diagnosis, and hence they do not scale well to large solar power plants with thousands of PV modules. Additionally, such measurements only capture one point in time,

*Corresponding author

Email address: sergiu.deitsch@fau.de (Sergiu Deitsch)

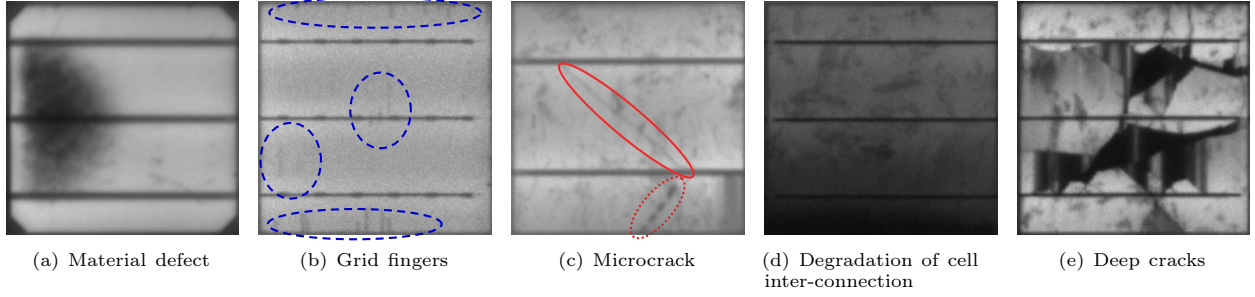


Figure 1: Various intrinsic and extrinsic defects in monocrystalline ((a)–(b)) and polycrystalline ((c)–(e)) solar cells. (a) shows a solar cell with a typical material defect. (b) shows finger interruptions in the encircled areas, which do not necessarily reduce the module efficiency. The solar cell in (c) contains a microcrack that is very subtle in its appearance. While microcracks do not divide the cell completely, they still must be detected because such cracks may grow over time and eventually impair the module efficiency. The spots at the bottom of this cell are likely to indicate cell damage as well. However, such spots can be oftentimes difficult to distinguish from actual material defects. (d) shows a disconnected area due to degradation of the cell interconnection. (e) shows deep cracks, which are usually caused by mechanical damage.

and as such may not reveal certain types of small cracks, which will become an issue over time [1].

Infrared (IR) imaging is a contactless alternative to direct measurements for assessing the quality of solar modules. Damaged solar modules can be easily identified by solar cells which are either partially or completely cut off from the electric circuit. As a result, the solar energy is not converted into electricity anymore, which heats the solar cells up. The emitted infrared radiation can then be imaged by an IR camera. However, IR cameras are limited by their relatively low resolution, which can prohibit detection of small defects such as microcracks not yet affecting the photoelectric conversion efficiency of a solar module.

Electroluminescence (EL) imaging is another established contactless technology for failure analysis of PV modules with the ability to image solar modules at a much higher resolution. In EL images, defective cells appear darker, because disconnected parts do not irradiate. To obtain an EL image, current is applied to a PV module, which induces EL emission at a wavelength of 1,150 nm. The emission can be imaged by a silicon Charge-coupled Device (CCD) sensor [2]. The high spatial image resolution enables the detection of microcracks [3], and EL imaging also does not suffer from blurring due to lateral heat propagation. However, visual inspection of EL images is not only time-consuming and expensive, but also requires trained specialists. In this work, we remove this constraint by proposing an automated method for classifying defects in EL images.

In general, defects in solar modules can be classified into two categories [4]: (1) material-intrinsic defects such as crystal grain boundaries and dislocations that can negatively affect the power efficiency of solar modules, and (2) extrinsic defects such as microcracks and breaks, which reduce the overall module efficiency over time. Finger interruptions are also commonly categorized as extrinsic defects. However, there is no indication that they necessarily influence the module efficiency during the aging process. Instead, the efficiency degradation induced by finger interruptions is a complex interaction between their size, position, and the number of interruptions [5].

Figure 1 shows an example EL image with different types of defects in monocrystalline and polycrystalline solar cells. Figure 1(a) and Fig. 1(b) show general material defects from the production process and finger interruptions, which do not necessarily reduce the lifespan of the affected solar panel. On the other hand, Fig. 1(c)–Fig. 1(e) show microcracks, degradation of cell-interconnections, and deep cracks that are well known to reduce the module efficiency. Particularly the detection of microcracks requires cameras with high spatial resolution.

For the detection of defects during monitoring one can set different goals. Highlighting the exact location of defects within a solar module allows to monitor affected areas with high precision. However, the exact defect location within the solar cell is less important for the quality assessment of a whole PV module. For this task, the overall likelihood indicating a cell defect is more important. This allows to quickly identify defective areas and to predict

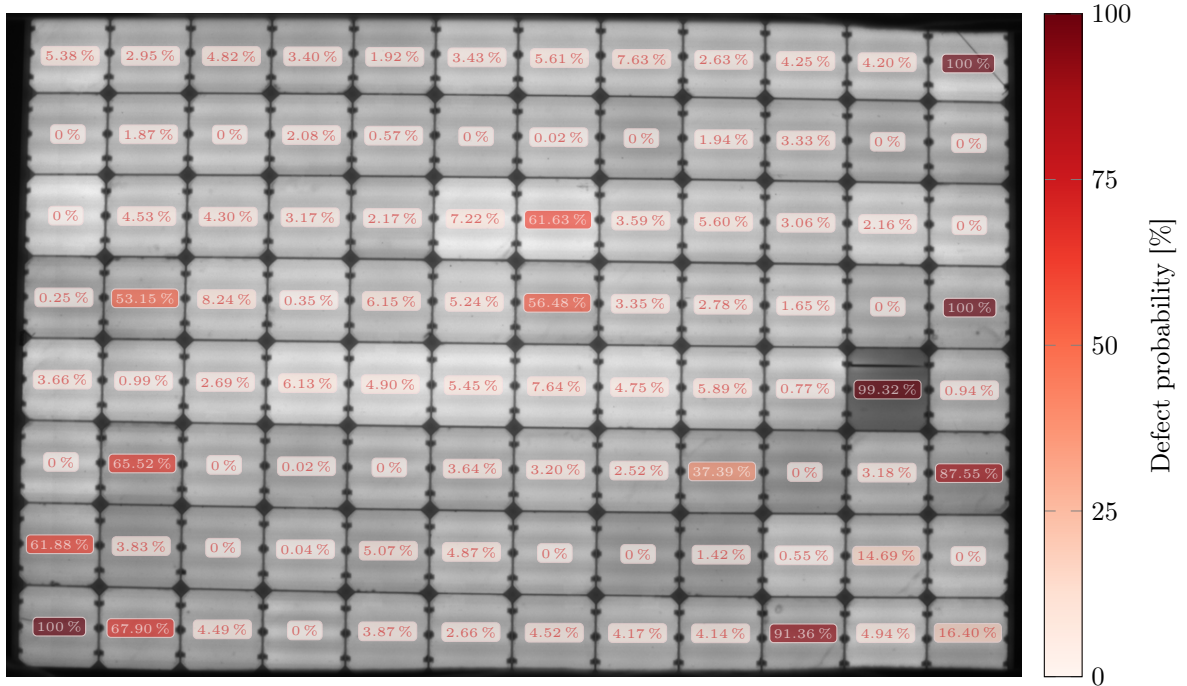


Figure 2: Defect probabilities inferred for each PV module cell by the proposed CNN. A darker shade of red indicates a higher likelihood of a cell defect.

future efficiency loss within a PV module. In this work, we propose two classification pipelines that automatically solve the second task of determining a per-cell defect likelihood that may lead to efficiency loss.

1.1. Contributions

The contribution of this work consists of three parts. First, we present a resource-efficient framework for supervised classification of defective solar cells using hand-crafted features and an SVM classifier that can be used on a wide range of commodity hardware, including tablet computers and drones equipped with low-power single-board computers. The low computational requirements make the on-site evaluation of the EL imagery possible, similar to analysis of low resolution IR images [6]. Second, we present a supervised classification framework using a convolutional neural network that is slightly more accurate, but requires a GPU for efficient training and classification. Third, we contribute an annotated dataset consisting of 2,624 aligned solar cells extracted from high resolution EL images to the community, and we use this dataset to perform an extensive evaluation and comparison of the proposed approaches.

Figure 2 shows the assessment results of a solar panel using the proposed convolutional neural network. Each solar cell in the EL image is overlaid by the likelihood of a defect in the corresponding cell.

1.2. Outline

The remainder of this work is organized as follows. Related work is reviewed in Section 2. Section 3 introduces both proposed classification approaches. In Section 4, we evaluate and compare these approaches, and discuss the results. This work is concluded in Section 5.

2. Related Work

Visual inspection of solar modules via EL imaging is an active research topic. Most of the related work, however, focuses on the detection of specific intrinsic or extrinsic defects, but not on the prediction of defects that eventually lower the power efficiency of solar modules.

For instance, Tsai et al. [7] use Fourier image reconstruction to detect defective solar cells in EL images of polycrystalline PV modules. The targeted extrinsic defects are (small) cracks, breaks, and finger interruptions. Fourier image reconstruction is

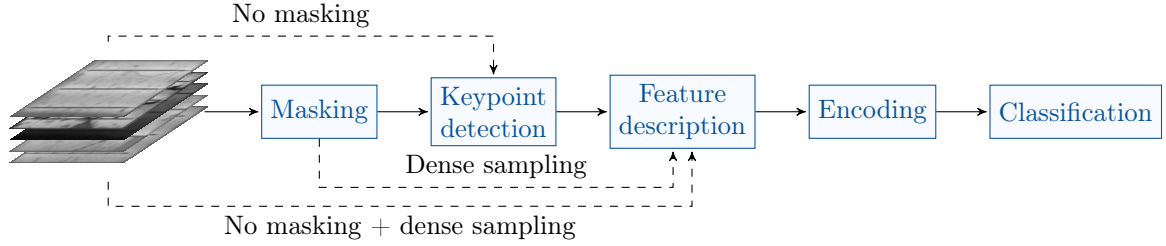


Figure 3: An overview of the SVM classification pipeline with the four proposed variations of the preprocessing and feature extraction process.

applied to remove possible defects by setting high-frequency coefficients associated with line- and bar-shaped artifacts to zero. The spectral representation is then transformed back into the spatial domain. The defects can then be identified as intensity differences between the original and the high-pass filtered image. Due to the shape assumption, the method has difficulties detecting defects with more complex shapes.

Tsai et al. [4] also introduced a supervised learning method for identification of defects using Independent Component Analysis (ICA) basis images. Defect-free solar cell subimages are used to find a set of independent basis images with ICA. The method achieves a high accuracy of 93.40 % with a relatively small training dataset of 300 solar cell subimages. However, material defects such as finger interruptions are treated equally to microcracks and deep cracks. This strategy is only suitable for detection of every abnormality on the surface of solar cells, but not for the prediction of future energy loss.

Anwar and Abdullah [8] developed an algorithm for the detection of microcracks in polycrystalline solar cells. They use anisotropic diffusion filtering followed by shape analysis to localize the defects in solar cells. While the method performs well at detecting microcracks, it does not consider other defect types such as completely disconnected cells, which appear completely dark in EL images.

Tseng et al. [9] proposed a method for automatic detection of finger interruptions in monocrystalline solar cells. The method employs binary clustering of features from candidate regions for the detection of defects. Finger interruptions, however, do not necessarily provide suitable cues for prediction of future power loss.

The success of deep learning led to a gradual replacement of traditional pattern recognition pipelines for optical inspection. However, to our

knowledge, no CNN architecture has been proposed for EL images, but only for other modalities or applications. Most closely related is the work by Mehta et al. [10], who presented a system for predicting the power loss, localization and type of soiling from RGB images of solar modules. Their approach does not require manual localization labels, but instead operates on images with the corresponding power loss as input.

Examples for other applications include the work by Masci et al. [11] who proposed an end-to-end max-pooling CNN for classifying steel defects. Their network performance is compared against multiple hand-crafted feature descriptors that are trained using SVMs. Although their dataset consists of only 2,281 training and 646 test images, the CNN architecture classifies steel defects at least twice as accurately as the SVMs. Zhang et al. [12] proposed a CNN architecture for detection of cracks on roads. To train the CNN, approximately 45,000 hand-labeled image patches were used. They show that CNNs greatly outperform hand-crafted features classified by a combination of an SVM and boosting. Esteva et al. [13] employ deep neural networks to classify different types of skin cancer. They trained the CNN end-to-end on a large dataset consisting of 129,450 clinical images and 2,032 different diseases.

3. Methodology

We subdivide each module into its solar cells, and analyze each cell individually. This breaks down the analysis to the smallest meaningful unit, in the sense that the mechanical design of PV modules interconnects units of cells in series. Also, the breakdown considerably increases the number of available data samples for training. For the segmentation of solar cells, we use a recently developed method [14], which brings every cell into a normal form free of perspective and lens distortions.

The investigated classification approaches in this work are SVM and CNN classifiers.

Support Vector Machines (SVMs) are trained on various features extracted from EL images of solar cells.

Convolutional Neural Network (CNN) is directly fed with image pixels of solar cells and the corresponding labels.

The SVM approach is computationally particularly efficient during training and inference. This allows to operate the method on a wide range of commodity hardware, such as tablet computers or drones, whose usage is dictated by the respective application scenario. Conversely, the prediction accuracy of the CNN is generally higher, while training and inference is much more time-intensive and commonly requires a GPU for an acceptably short runtime.

3.1. Classification Using Support Vector Machines

The general approach for classification using SVMs [15] is as follows. First, local descriptors are extracted from images of segmented PV cells. The features are typically extracted at salient points, also known as *keypoints*, or from a dense pixel grid. For training the classifier and subsequent predictions, a global representation needs to be computed from the set of local descriptors, oftentimes referred to as *encoding*. Finally, this global descriptor for a solar cell is classified into defective or functional. Figure 3 visualizes the classification pipeline, consisting of masking, keypoint detection, feature description, encoding, and classification. We describe these steps in the following subsections.

3.1.1. Masking

We assume that the solar cells were segmented from a PV module, e.g., using the automated algorithm we proposed in earlier work [14]. A binary mask allows then to separate the foreground of every cell from the background. The cell background includes image regions that generally do not belong to the silicon wafer, such as the busbars and the inter-cell borders. This mask can be used to strictly limit feature extraction to the cell interior. In the evaluation, we investigate the usefulness of masking, and find that its effect is minor, i.e., it only slightly improves the performance in a few feature/classifier combinations.

Table 1: Investigated keypoint detectors and feature descriptors. SIFT, SURF, and KAZE (in bold) contain both a detector and a descriptor. We explored also combinations of the keypoint detectors of AGAST and KAZE with other feature descriptors. Note, the keypoints provided by SIFT and SURF were not reliable enough and thus not further evaluated.

Method	Keypoint detector	Feature descriptor
AGAST [16]	✓	✗
KAZE [17]	✓	✓
HOG [18]	✗	✓
PHOW [19]	✗	✓
SIFT [20]	(✓)	✓
SURF [21]	(✓)	✓
VGG [22]	✗	✓

3.1.2. Feature Extraction

In order to train the SVMs, feature descriptors are computed. The locations of these local features are determined using two main sampling strategies: (1) keypoint detection, and (2) dense sampling. These strategies are exemplarily illustrated in Fig. 4.

Keypoint detection automatically determines salient points for feature descriptors. The number of detected keypoints varies greatly with the image content (and parameters), usually depending on the amount of high frequencies in the image. Keypoint detectors typically operate in scale space, allowing feature detection at different scale levels.

Dense sampling subdivides the 300×300 pixels PV cell by overlaying it with a grid consisting of $n \times n$ cells. The center of each grid cell specifies the position at which a feature descriptor will be subsequently extracted. In contrast to keypoint detectors, the number of feature locations does not vary with the image content, but only depends on the grid size.

We employ different popular combinations of keypoint detectors and feature extractors from the literature, as listed in Table 1 and outlined below.

Several algorithms combine keypoint detection and feature description. Probably the most popular of these methods is Scale-invariant Feature Transform (SIFT) [20], which detects and describes features at multiple scales. SIFT is invariant to rotation, translation, and scaling, and partially resilient to varying illumination conditions. Speeded Up Robust Features (SURF) [21] is a faster variant of SIFT, and also consists of a keypoint detector and

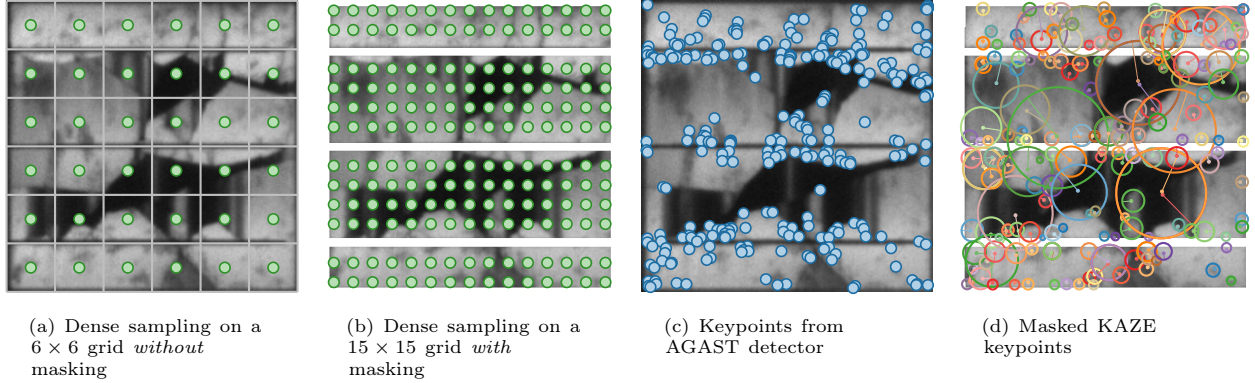


Figure 4: Two different feature extraction strategies applied to the same PV cell with and without masking. In (a), keypoints are sampled at fixed positions specified by the center of a cell in the overlaid grid. (b) uses equally sized and oriented keypoints laid out on a dense grid similar to (a). (c) shows an example for AGAST keypoints (detection threshold slightly increased for visualization). (d) shows KAZE keypoints of various sizes and orientations after masking out the background area.

a local feature descriptor. However, the detector part of SURF is not invariant to affine transformations. In initial experiments, we were not able to successfully use the keypoint detectors of SIFT and SURF, because the keypoint detector at times failed to detect features in relatively homogeneous monocrystalline cell images, and hence we used only the descriptor parts.

KAZE [17] is a multiscale feature detector and descriptor. The keypoint detection algorithm is very similar to SIFT, except that the linear Gaussian scale space used by SIFT is replaced by nonlinear diffusion filtering. For feature description, however, KAZE uses the SURF descriptor.

We also investigated Adaptive and Generic Accelerated Segment Test (AGAST) [16] as a dedicated keypoint detector without descriptor. It is based on a random forest classifier trained on a set of corner features that is known as Features from Accelerated Segment Test (FAST) [23, 24].

Among the dedicated descriptors, Pyramid Histogram of Visual Words (PHOW) [19] is an extension of SIFT that computes SIFT descriptors densely over a uniformly spaced grid. We use the implementation variant from VLFEAT [25]. Similarly, Histogram of Oriented Gradients (HOG) [18] is a gradient-based feature descriptor computed densely over a uniform set of image blocks. Finally, we also used the Visual Geometry Group (VGG) descriptor trained end-to-end using an efficient optimization method [22]. In our implementation, we employ the 120-dimensional real-valued descriptor variant.

We omitted binary descriptors from this selection. Even though binary feature descriptors are typically

very fast to compute, they generally do not perform better than real-valued descriptors [26].

3.1.3. Combinations of Detectors and Extractors

For the purpose of determining the most powerful feature detector/extractor combination, we evaluated all feature detector and feature extractor combinations with few exceptions.

In most cases, we neither tuned the parameters of keypoint detectors nor those of feature extractors but rather used the defaults by OPENCV [27] as of version 3.3.1. One notable exception is Adaptive and Generic Accelerated Segment Test (AGAST), where we lowered the detection threshold to 5 to be able to detect keypoints in monocrystalline PV modules. For SIFT and SURF, similar adjustments were not successful, which is why we only used their descriptors. HOG requires a grid of overlapping image regions, which is incompatible with the keypoint detectors. Instead, we downsampled the 300×300 pixels cell images to 256×256 pixels (the closest power of 2) for feature extraction. Masking was omitted for HOG due to implementation-specific limitations. Given these exceptions, we overall evaluate twelve feature combinations.

3.1.4. Encoding

The computed features are encoded into a global feature descriptor. The purpose of encoding is the formation of a single, fixed-length global descriptor from multiple local descriptors. Encoding is commonly represented as a histogram that draws its statistics from a background model. To this end,

we employ Vectors of Locally Aggregated Descriptors (VLAD) [28], which offers a compact state-of-the-art representation [29]. VLAD encoding is sometimes also used for deep learning based features in classification, identification and retrieval tasks [30, 31, 32, 33].

The VLAD dictionary is created by k -means clustering of a random subset of feature descriptors from the training set. For performance reasons, we use the fast mini-batch variant [34] of k -means. The cluster centroids μ_k correspond to anchor points of the dictionary. Afterwards, first order statistics are aggregated as a sum of residuals of all descriptors $\mathcal{X} := \{\mathbf{x}_t \in \mathbb{R}^d \mid t = 1, \dots, T\}$ extracted from a solar cell image. The residuals are computed with respect to their nearest anchor point μ_k in the dictionary $D := \{\mu_k \in \mathbb{R}^d \mid k = 1, \dots, K\}$ as

$$\nu_k := \sum_{t=1}^T \eta_k(\mathbf{x}_t)(\mathbf{x}_t - \mu_k) \quad (1)$$

where $\eta_k: \mathbb{R}^d \rightarrow \{0, 1\}$ is an indicator function to cluster membership, i.e.:

$$\eta_k(\mathbf{x}) := \begin{cases} 1 & \text{if } k = \underset{j=1, \dots, K}{\operatorname{argmin}} \|\mathbf{x} - \mu_j\|_2 \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

which indicates whether \mathbf{x} is the nearest neighbor of μ_k . The final VLAD representation $\nu \in \mathbb{R}^{Kd}$ corresponds to the concatenation of all residual terms (1) into a Kd -dimensional vector:

$$\nu := (\nu_1^\top, \dots, \nu_K^\top)^\top. \quad (3)$$

Several normalization steps are required to make the VLAD descriptor robust. Power normalization addresses issues when some local descriptors occur more frequently than others. Here, each element of the global descriptor $v_i \in \nu$ is normalized as

$$\hat{v}_i := \operatorname{sign}(v_i)|v_i|^\rho, \quad i = 1, \dots, Kd \quad (4)$$

where we chose $\rho = 0.5$ as a typical value from the literature. After power normalization, the vector is normalized such that its ℓ^2 -norm equals one.

Similarly, an over-counting of *co-occurrences* can occur if at least two descriptors appear together frequently. Jégou and Ondřej [35] showed that Principal Component Analysis (PCA) whitening effectively eliminates such co-occurrences and additionally decorrelates the data.

To enhance the robustness of the codebook D against potentially suboptimal solutions from the

probabilistic k -means clustering, we compute five VLAD representations from different training subsets using different random seeds. Afterwards, the concatenation of the VLAD encodings $\tilde{\nu} := (\tilde{\nu}_1^\top, \dots, \tilde{\nu}_m^\top)^\top \in \mathbb{R}^{mKd}$ is jointly decorrelated and whitened by means of PCA [36]. The transformed representation is again normalized such that its ℓ^2 -norm equals one and the result is eventually passed to the SVM classifier.

3.1.5. Support Vector Machine Training

We trained SVMs both with a linear and a Radial Basis Function (RBF) kernel. For the linear kernel, we use LIBLINEAR [37], which is optimized for linear classification tasks and large datasets. For the non-linear RBF kernel, we use LIBSVM [38].

The SVM hyperparameters are determined by evaluating the average F_1 score in an inner five-fold cross-validation on the training set using a grid search. For the linear SVM, we employ the ℓ^2 penalty on a squared hinge loss. The penalty parameter C is selected from a set of powers of ten, i.e., $C_{\text{linear}} \in \{10^k \mid k = -2, \dots, 6\} \subset \mathbb{R}_{>0}$. For RBF SVMs, the penalty parameter C is determined from a slightly smaller set $C_{\text{RBF}} \in \{10^k \mid k = 2, \dots, 6\}$. The search space of the kernel coefficient γ is constrained to $\gamma \in \{10^{-7}, 10^{-6}, S^{-1}\} \subset [0, 1]$, where S denotes the number of training samples.

3.2. Regression Using a Deep Convolutional Neural Network

We considered several strategies to train the CNN. Given the limited amount of data we had at our disposal, best results were achieved by means of transfer learning. We utilized the VGG19 [39] network architecture originally trained on the IMAGENET dataset [40] using 1.28 million images and 1,000 classes. We then refined the network using our dataset.

We replaced the two fully connected layers of VGG19 by a Global Average Pooling (GAP) [41] and two fully connected layers with 4,096 and 2,048 neurons, respectively. The GAP layer is used to make the VGG19 network input tensor ($224 \times 224 \times 3$) compatible to the resolution of our solar cell image samples ($300 \times 300 \times 3$), in order to avoid additional downsampling of the samples. The output layer consists of a single neuron that outputs the defect probability of a cell. The CNN is refined by minimizing the Mean Squared Error (MSE) loss function. Hereby, we essentially train a deep regression network, which allows us to predict (continuous)

Table 2: Partitioning of the solar cells into functional and defective, with an additional self-assessment on the rater’s confidence after visual inspection. Non-confident decisions obtain a weight lower than 100 % for the evaluation of the classifier performance.

Condition	Confident?	Label p	Weight w
functional	✓	functional	100 %
	✗	defective	33 %
defective	✓	defective	100 %
	✗	defective	67 %

defect probabilities trained using only two defect likelihood categories (functional and defective). By rounding the predicted continuous probability to the nearest neighbor of the four original classes, we can directly compare CNN decisions against the original ground truth labels without binarizing them.

Data augmentation is used to generate additional, slightly perturbed training samples. The augmentation variability, however, is kept moderate, since the segmented cells vary only by few pixels along the translational axes, and few degrees along the axis of rotation. The training samples are scaled by at most 2 % of the original resolution. The rotation range is capped to $\pm 3^\circ$. The translation is limited to $\pm 2\%$ of the cell dimensions. We also use random flips along the vertical and horizontal axes. Since the busbars can be laid out both vertically and horizontally, we additionally include training samples rotated by exactly 90° . The rotated samples are augmented the same way as described above.

We fine-tune the pretrained model in two stages. First, we train only the fully connected layers while keeping the weights of the convolutional blocks fixed. Here, we employ the ADAM optimizer [42] with a learning rate of 10^{-3} , exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and the regularization value $\hat{\epsilon} = 10^{-8}$. In the second step, we refine the weights of all layers. At this stage, we use the Stochastic Gradient Descent (SGD) optimizer with a learning rate of $5 \cdot 10^{-4}$ and a momentum of 0.9.

In both stages, we process the augmented versions of the 1,968 training samples in mini-batches of 16 samples on two NVIDIA GeForce GTX 1080, and run the training procedure for a maximum of 100 epochs. This totals to 196,800 augmented variations of the original 1,968 training samples that are used to refine the network. For the implementation of the deep regression network, we use KERAS ver-

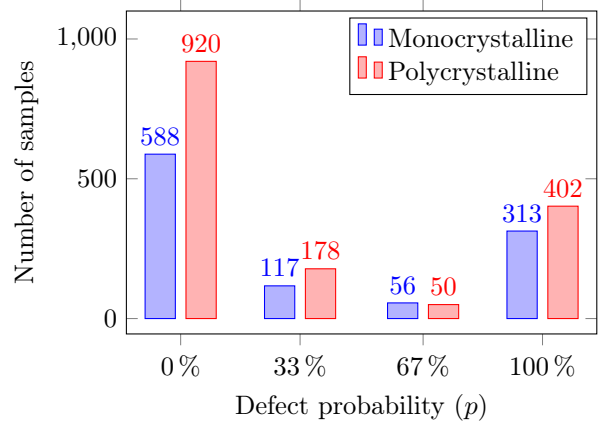


Figure 5: The distribution of the total number of cells in our dataset depending on their label p and the PV module type from which the cells were originally extracted.

sion 2.0 [43] with TENSORFLOW version 1.4 [44] in the backend.

4. Evaluation

For the quantitative evaluation, we first evaluate different feature descriptors extracted densely over a grid. Then, we compare the best configurations against feature descriptors extracted at automatically detected keypoints to determine the best performing variation of the SVM classification pipeline. We then compare it against the proposed deep CNN, and visualize the internal feature mapping of the CNN.

4.1. Dataset

We use a public¹ dataset of solar cells extracted from high resolution EL images [45]. The EL images of monocrystalline and polycrystalline PV modules were captured in a dark room using a camera placed on a tripod. A controlled environment for taking EL images was necessary due to a long exposure time required for image acquisition. This especially stems from the fact that the amount of radiation emitted by PV modules is comparatively low opposed to background radiation on an average day.

In total, we extracted 2,624 cells from the modules at a resolution of 300×300 pixels from 44 different PV modules, where 18 modules are of monocrystalline type, and 26 are of polycrystalline type. The

¹The solar cell dataset is available at <https://github.com/zae-bayern/elpv-dataset>

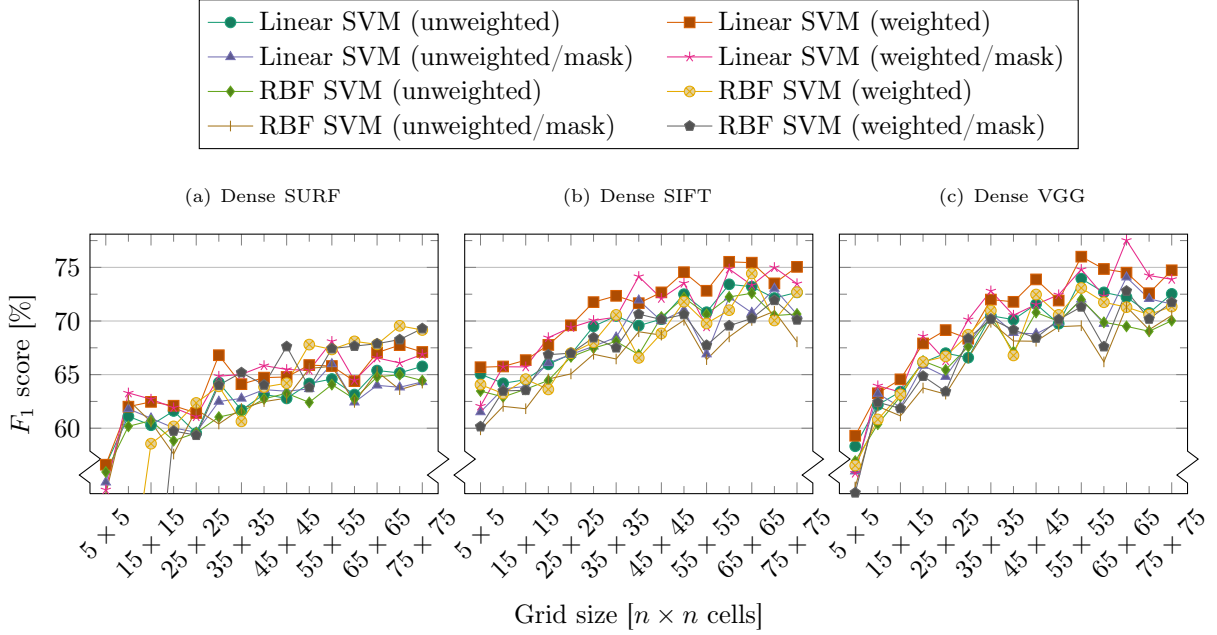


Figure 6: Classification performance for different dense sampling configurations in terms of F_1 score grouped by the feature descriptor, classifier, weighting strategy, and the use of masking. The highest F_1 score is achieved using a linear SVM and the VGG feature descriptor at a grid resolution of 65×65 cells with sample weighting and masking (—*—) (c).

solar cells exhibit intrinsic and extrinsic defects commonly occurring in mono- and polycrystalline solar modules. In particular, the dataset includes micro-cracks and deep cracks, short-circuited cells, open inter-connects, and soldering failures. These cell defects are widely known to negatively influence efficiency, reliability, and durability of solar modules. Finger interruptions are excluded since the power loss caused by such defects is typically negligible.

Measurements of power degradation were not available to provide the ground truth. Instead, the extracted cells were presented in random order to a recognized expert, who is very familiar with intricate details of different defects in EL images. The expert answered the questions (1) is the cell *functional* or *defective*? (2) are you *confident* in your assessment? The assessments into functional and defective cells by a confident rater were directly used as labels. Non-confident assessments of functional and defective cells were all labeled as defective. To reflect the rater’s uncertainty, lower weights are assigned to these assessments, namely a weight of 33 % to a non-confident assessment of functional cell, and a weight of 67 % to a non-confident assessment of defective cell. Table 2 shows this in summary, with the rater assessment on the left, and the associated

classification labels and weights on the right. Figure 5 shows the distribution of ground truth solar cell labels, separated by the type of the source PV module.

We used 25 % of the labeled cells (656 cells) for testing, and the remaining 75 % (1,968 cells) for training. Stratified sampling was used to randomly split the samples while ensuring similar distribution of samples within different classes in the training and the test sets. To further balance the training set, we weight the classes using the inverse proportion heuristic derived from King and Zeng [46]

$$c_j := \frac{S}{2n_j}, \quad (5)$$

where S is the total number of training samples, and n_j is the number of functional ($j = 0$) or defective ($j = 1$) samples.

4.2. Dense Sampling

In this experiment, we evaluate different grid sizes for subdividing a single 300×300 pixels cell image. The number of grid points per cell is varied between 5×5 to 75×75 points. At each grid point, SIFT, SURF, and VGG descriptors are computed. The remaining two descriptors, PHOW and HOG, are

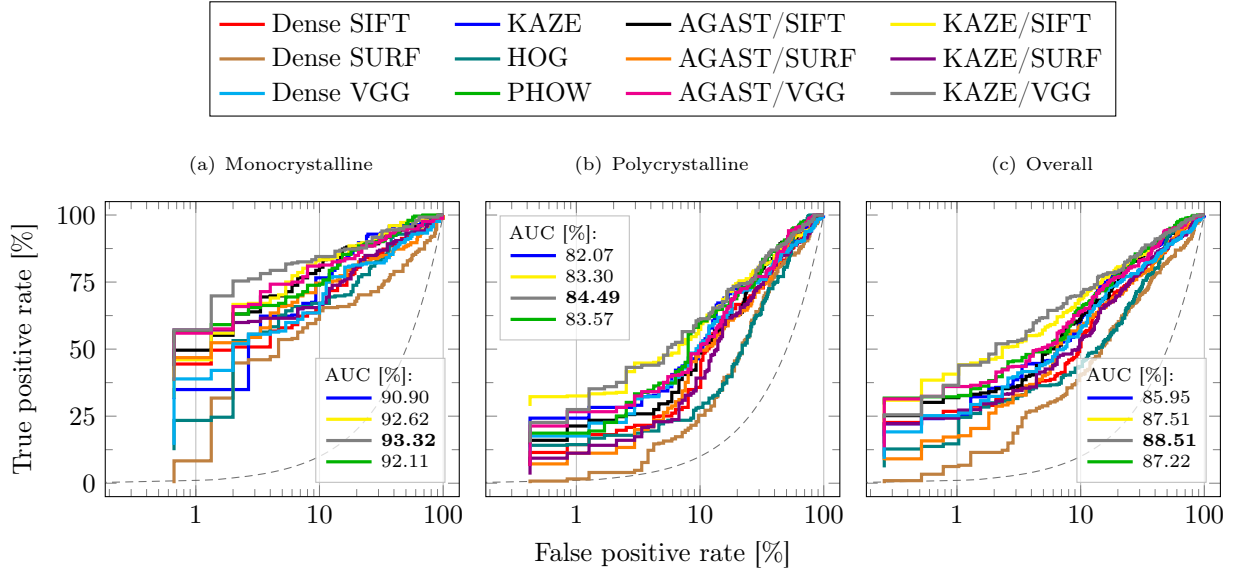


Figure 7: Receiver Operating Characteristic (ROC) for top performing feature detector/extractor combinations grouped by mono-, polycrystalline, and both solar module types combined. Note the logarithmic scale of the false positive rate axis. See text for details.

omitted in this experiment, because they do not allow to arbitrarily specify the position for descriptor computation. Note that at a 75×75 point grid, the distance between two grid points is only 4 pixels, which leads to a significant overlap between neighbored descriptors. Therefore, further increase of the grid resolution cannot be expected to considerably improve the classification results.

The goal of this experiment is to find the best performing combination of grid size and classifier. We trained both linear SVMs and SVMs with the RBF kernel. For each classifier, we also examine two additional options, namely whether the addition of the sample weights w (confer Table 2) or masking out the background region (confer Section 3.1.1) improves the classifiers.

Performance is measured using the F_1 score, which is the harmonic mean of precision and recall. Figure 6 shows the F_1 scores that are averaged over the individual per-class F_1 scores. From left to right, these scores are shown for the SURF descriptor (Fig. 6(a)), SIFT descriptor (Fig. 6(b)) and VGG descriptor (Fig. 6(c)). Here, the VGG descriptor achieves the highest score on a grid of size 65×65 using a linear SVM with weighting and masking. SIFT is the second best performing descriptor with best performance on a 60×60 grid using linear SVM with weighting, but without masking. SURF

achieved the lowest scores, with a peak at a 70×70 grid using an RBF SVM with weighting, but without masking. The results show the trend that more grid points lead to better results. The classification accuracy of SURF increases only slowly and saturates at about 70 %. SIFT and VGG benefit more from denser grids. The use of the weights w leads in most cases to a higher score, because the classifier can stronger rely on samples for which the expert labeler was more confident. Masking also improves the F_1 score for VGG features. However, the improvement by almost two percent is small compared to the overall performance variation over the configurations. One can argue that the cell structure is not substantial for distinguishing different kinds of cell defects given the high density of the feature points and the degree of overlap between image regions evaluated by feature extractors.

4.3. Dense Sampling vs. Keypoint Detection

This experiment aims at comparing the classification performance of dense grid-based features versus keypoint-based features. To this end, the best performing grid-based classifier per descriptor from the previous experiment are compared to combinations of keypoint detectors and feature descriptors.

Figure 7 shows the evaluated detector and extractor combinations for monocrystalline cells, poly-

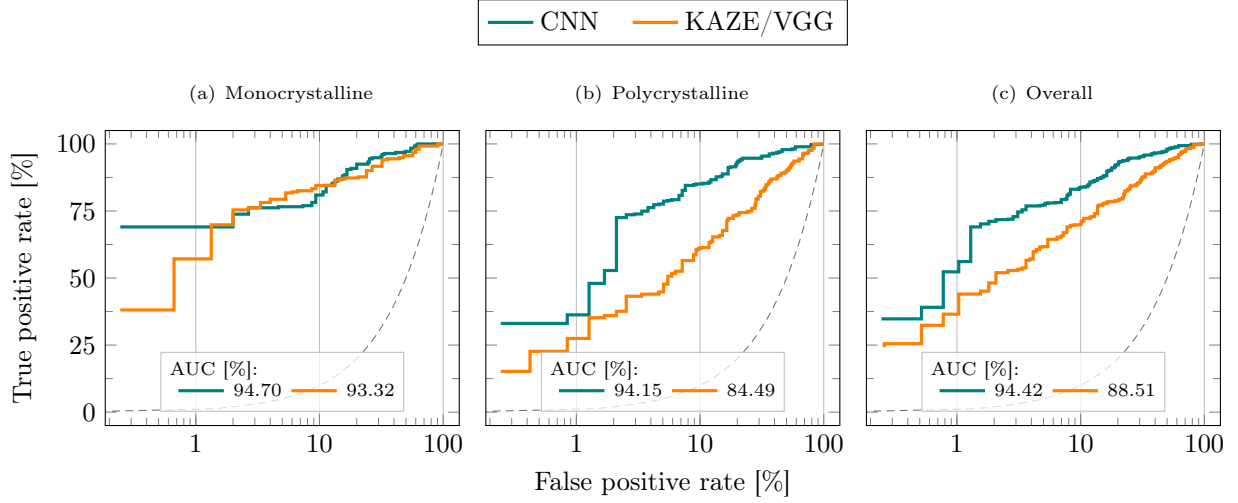


Figure 8: Receiver Operating Characteristic (ROC) curves of the best performing KAZE/VGG feature detector/descriptor combination (—) compared to the ROC of the deep regression network (—). While in the monocrystalline case (a) the classification performance of the CNN is almost on par with the linear SVM. For polycrystalline PV modules (b) the CNN considerably outperforms SVM with the linear kernel trained on KAZE/VGG features. The latter outcome leads to a higher CNN ROC Area Under the Curve (AUC) for both PV modules types combined (c). The dashed curve (---) represents the baseline in terms of a random classifier.

crystalline cells, and both together. Most detector/extractor combinations are specified by a forward slash (*Detector/Descriptor*). Entries without a forward slash, namely KAZE, HOG, and PHOW, denote features which already include both a detector and a descriptor. The three best performing methods on a dense grid are denoted as Dense SIFT 60×60 , Dense SURF 70×70 , and Dense VGG 65×65 , respectively. Unless otherwise specified, the features were trained with sample weighting, without masking, and using a linear SVM.

The performance is shown using ROC curves that indicate the performance of binary classifiers at various false positive rates. Additionally, the plots show the AUC scores for the top-4 features with the highest AUC emphasized in bold. In all three cases, KAZE/VGG outperforms other feature combinations with an AUC of 88.51% on all modules, followed by KAZE/SIFT with an AUC of 87.22%. As an exception, the second best feature combination for polycrystalline solar cells in terms of AUC is PHOW. The gray dashed curve represents the baseline in terms of a random classifier. Overall, the use of keypoints leads to better performance than dense sampling.

4.4. Support Vector Machine vs. Transfer Learning Using Deep Regression Network

Figure 8 shows the performance of the strongest SVM variant, KAZE/VGG, in comparison to the CNN classifier. The ROC curve on the left in Fig. 8(a) contains the results for monocrystalline PV modules. Figure 8(b) in the center provides the classification performance for polycrystalline PV modules. Finally, the overall classification performance of both models is shown in Fig. 8(c) on the right.

Notably, the classification performance of SVM and CNN is very similar for monocrystalline PV modules. The CNN performs on average only slightly better than the SVM. At lower false positive rates around and below 1%, the CNN achieves a higher true positive rate. In the range of roughly 1 to 10% false positive rate, however, the SVM performs better. This shows that KAZE/VGG is able to capture surface abnormalities on homogeneous surfaces almost as accurate as a CNN that is directly trained on the image pixels.

For polycrystalline PV modules, the CNN is able to predict defective solar cells almost 10% more accurately than the SVM in terms of the AUC. This is also clearly a more difficult test due to the large variety of textures among the solar cells.

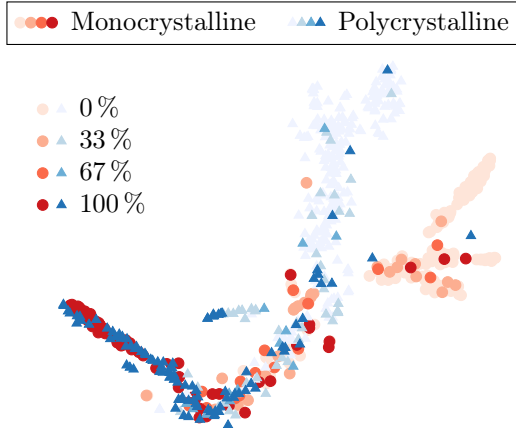


Figure 9: t -SNE visualization of the CNN’s last hidden layer output for the four defect probability classes quantized from predictions of the deep regression network. The 2,048-dimensional output layer is mapped to a two dimensional space for all 656 test images.

Overall, the CNN outperforms the SVM. However, the performances of both classifiers differ in total by only about 6 %. The SVM classifier can therefore also be useful for a quick, on-the-spot assessment of a PV module in situations where specialized hardware for a CNN is not available.

4.5. Analysis of the CNN Feature Space

Here, we analyze the features learned by the CNN using t -distributed Stochastic Neighbor Embedding (t -SNE), a manifold learning technique for dimensionality reduction. The purpose is to examine the criteria for separation of different solar cell clusters. To this end, we use the Barnes-Hut variant [47] of t -SNE which is substantially faster than the standard implementation. For computing the embeddings, we fixed t -SNE’s perplexity parameter to 35. Due to the small size of our test dataset, we avoided an initial dimensionality reduction of the features using PCA in the preprocessing step, but rather used random initialization of embeddings.

The resulting representation for all 656 test images is shown in Fig. 9. Each point corresponds to a feature vector projected from the 2,048-dimensional last layer of the CNN onto two dimensions. The projected feature vectors are grouped into two categories: mono- and polycrystalline solar modules from which the corresponding cells were originally extracted.

An important observation is that the class of definitely defective (100 %) cells forms a single elongated cluster (bottom left) that includes cells irrespective

of the source PV module type. In contrast to this, definitely functional cells (0 %) are separated into different clusters which depend on the type of the source PV module. The overall appearance of the cell (i.e., the number of soldering joints, texture-ness, etc.) additionally generates several branches in the monocrystalline cluster (on the right). These branches include samples grouped by the number of busbar soldering joints within the cell. Here, the branches are more pronounced than the separations in the cluster of functional polycrystalline cells (at the top right) due to the homogeneous (i.e., texture-less) surface of the silicon wafers.

The clusters for the categories of possibly defective (33 %) and likely defective (67 %) cells are mixed. The high confusion between these samples stems from the comparably small size of the corresponding categories compared to the size of the two remaining categories of high confidence samples in our dataset (see Fig. 5). Additionally, the samples from these two categories can stimulate ambiguous decisions due to being at the boundary of clearly distinguishable defects and non-defects.

4.6. Discussion

Several conclusions can be drawn from the evaluation results. First, masking can be useful if the spatial distribution of detected features is rather sparse. However, in most cases masking does not improve the classification accuracy. Secondly, weighting samples proportionally to the confidence of the defect likelihood in a cell does improve the generalization ability of the learned classifiers.

KAZE/VGG features trained using linear SVM is the best performing SVM pipeline variant with an accuracy of 82.44 % and an F_1 score of 82.52 %. The CNN is even more accurate. It distinguishes functional and defective solar cells with an accuracy of 88.42 %. The corresponding F_1 score is 88.39 %. The 2-dimensional visualization of the CNN feature distribution via t -SNE underlines that the network learns the actual structure of the task at hand.

A limitation of the proposed method is that each solar cell is examined independently. In particular, some types of surface abnormalities that do not affect the module efficiency can appear in repetitive patterns across cells. Accurate classification of such larger-scale effects requires to take context into consideration, which is subject to future work.

5. Conclusions

We presented a general framework for training an SVM and a CNN classifier to predict the defect likelihood of solar cells in high resolution EL images. The processing pipeline for the SVM classifier is carefully designed. In a series of experiments, the best performing pipeline is determined as KAZE/VGG features in a linear SVM trained on samples that take the confidence of the labeler into consideration. The CNN network is a fine-tuned regression network based on VGG19, trained on augmented cell images that also consider the labeler confidence.

On monocrystalline solar modules, both classifiers perform similarly well, with only a slight advantage on average for the CNN. However, the CNN classifier outperforms the SVM classifier by about 6 % accuracy on the more inhomogeneous polycrystalline cells. This leads also to the better average accuracy across all cells of 88.42 % for the CNN versus 82.44 % for the SVM. The high accuracies make both classifiers useful for visual inspection. If the application scenario permits the usage of GPUs and higher processing times, the computationally more expensive CNN is preferred. Otherwise, the SVM classifier is a viable alternative for applications that require a low resource footprint.

Acknowledgments

This work was funded by Energie Campus Nürnberg (EnCN) and partially supported by the Research Training Group 1773 “Heterogenous Image Systems” funded by the German Research Foundation (DFG).

References

- [1] S. Kajari-Schröder, I. Kunze, M. Köntges, Criticality of cracks in PV modules, *Energy Procedia* 27 (2012) 658–663. doi: [10.1016/j.egypro.2012.07.125](https://doi.org/10.1016/j.egypro.2012.07.125).
- [2] M. Köntges, S. Kurtz, C. Packard, U. Jahn, K. Berger, K. Kato, T. Friesen, H. Liu, M. Van Iseghem, Review of Failures of Photovoltaic Modules, Technical Report, International Energy Agency, 2014.
- [3] O. Breitenstein, J. Bauer, K. Bothe, D. Hinken, J. Müller, W. Kwapil, M. C. Schubert, W. Warta, Can luminescence imaging replace lock-in thermography on solar cells?, *IEEE Journal of Photovoltaics* 1 (2011) 159–167. doi: [10.1109/JPHOTOV.2011.2169394](https://doi.org/10.1109/JPHOTOV.2011.2169394).
- [4] D.-M. Tsai, S.-C. Wu, W.-Y. Chiu, Defect detection in solar modules using ICA basis images, *IEEE Transactions on Industrial Informatics* 9 (2013) 122–131. doi: [10.1109/TII.2012.2209663](https://doi.org/10.1109/TII.2012.2209663).
- [5] R. De Rose, A. Malomo, P. Magnone, F. Crupi, G. Cellere, M. Martire, D. Tonini, E. Sangiorgi, A methodology to account for the finger interruptions in solar cell performance, *Microelectronics Reliability* 52 (2012) 2500–2503. doi: [10.1016/j.microrel.2012.07.014](https://doi.org/10.1016/j.microrel.2012.07.014).
- [6] S. Dotenco, M. Dalsass, L. Winkler, T. Würzner, C. Brabec, A. Maier, F. Gallwitz, Automatic detection and analysis of photovoltaic modules in aerial infrared imagery, in: *Winter Conference on Applications of Computer Vision (WACV)*, 2016, p. 9. doi: [10.1109/WACV.2016.7477658](https://doi.org/10.1109/WACV.2016.7477658).
- [7] D.-M. Tsai, S.-C. Wu, W.-C. Li, Defect detection of solar cells in electroluminescence images using Fourier image reconstruction, *Solar Energy Materials and Solar Cells* 99 (2012) 250–262. doi: [10.1016/j.solmat.2011.12.007](https://doi.org/10.1016/j.solmat.2011.12.007).
- [8] S. A. Anwar, M. Z. Abdullah, Micro-crack detection of multicrystalline solar cells featuring an improved anisotropic diffusion filter and image segmentation technique, *EURASIP Journal on Image and Video Processing* 2014 (2014) 15. doi: [10.1186/1687-5281-2014-15](https://doi.org/10.1186/1687-5281-2014-15).
- [9] D.-C. Tseng, Y.-S. Liu, C.-M. Chou, Automatic finger interruption detection in electroluminescence images of multicrystalline solar cells, *Mathematical Problems in Engineering* 2015 (2015) 1–12. doi: [10.1155/2015/879675](https://doi.org/10.1155/2015/879675).
- [10] S. Mehta, A. P. Azad, S. A. Chemmengath, V. Raykar, S. Kalyanaraman, Deepsolareye: Power loss prediction and weakly supervised soiling localization via fully convolutional networks for solar panels, in: *Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 333–342. doi: [10.1109/WACV.2018.00043](https://doi.org/10.1109/WACV.2018.00043).
- [11] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, G. Fricout, Steel defect classification with max-pooling convolutional neural networks, in: *International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1–6. doi: [10.1109/IJCNN.2012.6252468](https://doi.org/10.1109/IJCNN.2012.6252468).
- [12] L. Zhang, F. Yang, Y. Daniel Zhang, Y. J. Zhu, Road crack detection using deep convolutional neural network, in: *International Conference on Image Processing (ICIP)*, 2016, pp. 3708–3712. doi: [10.1109/ICIP.2016.7533052](https://doi.org/10.1109/ICIP.2016.7533052).
- [13] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, *Nature* 542 (2017) 115–118. doi: [10.1038/nature21056](https://doi.org/10.1038/nature21056).
- [14] S. Deitsch, C. Buerhop-Lutz, A. Maier, F. Gallwitz, C. Riess, Segmentation of Photovoltaic Module Cells in Electroluminescence Images, e-print, arXiv, 2018. [arXiv:1806.06530](https://arxiv.org/abs/1806.06530).
- [15] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning* 20 (1995) 273–297. doi: [10.1007/BF00994018](https://doi.org/10.1007/BF00994018).
- [16] E. Mair, G. D. Hager, D. Burschka, M. Suppa, G. Hirzinger, Adaptive and generic corner detection based on the accelerated segment test, in: *European Conference on Computer Vision (ECCV)*, volume 6312 of *Lecture Notes in Computer Science*, 2010, pp. 183–196. doi: [10.1007/978-3-642-15552-9_14](https://doi.org/10.1007/978-3-642-15552-9_14).
- [17] P. F. Alcantarilla, A. Bartoli, A. J. Davison, KAZE features, in: *European Conference on Computer Vision (ECCV)*, volume 7577 of *Lecture Notes in Computer Science*, 2012, pp. 214–227. doi: [10.1007/978-3-642-33783-3_16](https://doi.org/10.1007/978-3-642-33783-3_16).
- [18] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: *Conference on Computer Vision*

- and Pattern Recognition (CVPR), volume 1, 2005, pp. 886–893. doi: [10.1109/CVPR.2005.177](https://doi.org/10.1109/CVPR.2005.177).
- [19] A. Bosch, A. Zisserman, X. Munoz, Image classification using random forests and ferns, in: International Conference on Computer Vision (ICCV), 2007, pp. 1–8. doi: [10.1109/ICCV.2007.4409066](https://doi.org/10.1109/ICCV.2007.4409066).
 - [20] D. G. Lowe, Object recognition from local scale-invariant features, in: International Conference on Computer Vision (ICCV), volume 2, 1999, pp. 1150–1157. doi: [10.1109/ICCV.1999.790410](https://doi.org/10.1109/ICCV.1999.790410).
 - [21] H. Bay, A. Essa, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), *Computer Vision and Image Understanding* 110 (2008) 346–359. doi: [10.1016/j.cviu.2007.09.014](https://doi.org/10.1016/j.cviu.2007.09.014).
 - [22] K. Simonyan, A. Vedaldi, A. Zisserman, Learning local feature descriptors using convex optimisation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (2014) 1573–1585. doi: [10.1109/TPAMI.2014.2301163](https://doi.org/10.1109/TPAMI.2014.2301163).
 - [23] E. Rosten, T. Drummond, Fusing points and lines for high performance tracking, in: International Conference on Computer Vision (ICCV), 2005, pp. 1508–1515. doi: [10.1109/ICCV.2005.104](https://doi.org/10.1109/ICCV.2005.104).
 - [24] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: European Conference on Computer Vision (ECCV), volume 3951 of *Lecture Notes in Computer Science*, 2006, pp. 430–443. doi: [10.1007/11744023_34](https://doi.org/10.1007/11744023_34).
 - [25] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, 2008. URL: <http://www.vlfeat.org>.
 - [26] J. Heinly, E. Dunn, J.-M. Frahm, Comparative evaluation of binary features, in: European Conference on Computer Vision (ECCV), volume 7573 of *Lecture Notes in Computer Science*, 2012, pp. 759–773. doi: [10.1007/978-3-642-33709-3_54](https://doi.org/10.1007/978-3-642-33709-3_54).
 - [27] Itseez, Open source computer vision library (OpenCV), 2017. URL: <https://github.com/itseez/opencv>.
 - [28] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012) 1704–1716. doi: [10.1109/TPAMI.2011.235](https://doi.org/10.1109/TPAMI.2011.235).
 - [29] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice, *Computer Vision and Image Understanding* 150 (2015) 109–125. doi: [10.1016/j.cviu.2016.03.013](https://doi.org/10.1016/j.cviu.2016.03.013).
 - [30] Y. Gong, L. Wang, R. Guo, S. Lazebnik, Multi-scale orderless pooling of deep convolutional activation features, in: European Conference on Computer Vision (ECCV), volume 8695, 2014, pp. 392–407. doi: [10.1007/978-3-319-10584-0_26](https://doi.org/10.1007/978-3-319-10584-0_26).
 - [31] J. Y. H. Ng, F. Yang, L. S. Davis, Exploiting local features from deep networks for image retrieval, in: Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 53–61. doi: [10.1109/CVPRW.2015.7301272](https://doi.org/10.1109/CVPRW.2015.7301272).
 - [32] M. Paulin, J. Mairal, M. Douze, Z. Harchaoui, F. Perronnin, C. Schmid, Convolutional patch representations for image retrieval: An unsupervised approach, *International Journal of Computer Vision* 121 (2016) 149–168. doi: [10.1007/s11263-016-0924-3](https://doi.org/10.1007/s11263-016-0924-3).
 - [33] V. Christlein, M. Gropp, S. Fiel, A. K. Maier, Unsupervised feature learning for writer identification and writer retrieval, in: International Conference on Document Analysis and Recognition (ICDAR), volume 1, 2017, pp. 991–997. doi: [10.1109/ICDAR.2017.165](https://doi.org/10.1109/ICDAR.2017.165).
 - [34] D. Sculley, Web-scale k -means clustering, in: International Conference on World Wide Web (WWW), 2010, pp. 1177–1178. doi: [10.1145/1772690.1772862](https://doi.org/10.1145/1772690.1772862).
 - [35] H. Jégou, C. Ondřej, Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening, in: European Conference on Computer Vision (ECCV), volume 7573 of *Lecture Notes in Computer Science*, 2012, pp. 774–787. doi: [10.1007/978-3-642-33709-3_55](https://doi.org/10.1007/978-3-642-33709-3_55).
 - [36] A. Kessy, A. Lewin, K. Strimmer, Optimal Whitening and Decorrelation, e-print, arXiv, 2016. [arXiv:1512.00809](https://arxiv.org/abs/1512.00809).
 - [37] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A library for large linear classification, *The Journal of Machine Learning Research* 9 (2008) 1871–1874.
 - [38] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology* 2 (2011) 27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
 - [39] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, e-print, arXiv, 2014. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
 - [40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255. doi: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
 - [41] M. Lin, Q. Chen, S. Yan, Network In Network, e-print, arXiv, 2013. [arXiv:1312.4400](https://arxiv.org/abs/1312.4400).
 - [42] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, e-print, arXiv, 2014. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
 - [43] F. Chollet, et al., Keras, GitHub, 2015. URL: <https://github.com/keras-team/keras>.
 - [44] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL: <https://www.tensorflow.org>.
 - [45] C. Buerhop-Lutz, S. Deitsch, A. Maier, F. Gallwitz, J. Hauch, C. Camus, C. J. Brabec, A benchmark for visual identification of defective solar cells in electroluminescence imagery, in: 35th European PV Solar Energy Conference and Exhibition, 2018. To appear.
 - [46] G. King, L. Zeng, Logistic regression in rare events data, *Political Analysis* 9 (2001) 137–163. doi: [10.1093/oxfordjournals.pan.a004868](https://doi.org/10.1093/oxfordjournals.pan.a004868).
 - [47] L. van der Maaten, Barnes-Hut-SNE, e-print, arXiv, 2013. [arXiv:1301.3342](https://arxiv.org/abs/1301.3342).