Solar Energy 185 (2019) 455–468

Contents lists available at ScienceDirect

Solar Energy

journal homepage: www.elsevier.com/locate/solener

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]

[a] Energy Campus Nuremberg, Fürther Str. 250, 90429 Nuremberg, Germany
[b] Nuremberg Institute of Technology, Department of Computer Science, Hohfederstr. 40, 90489 Nuremberg, Germany
[c] Bavarian Center for Applied Energy Research, Immerwahrstr. 2, 91058 Erlangen, Germany
[d] Pattern Recognition Lab, University of Erlangen-Nuremberg, Martensstr. 3, 91058 Erlangen, Germany

ARTICLE INFO

Keywords:
Deep learning
Defect classification
Electroluminescence imaging
Photovoltaic modules
Regression analysis
Support vector machines
Visual inspection

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 1968 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.

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 cannot 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, and as such may not reveal certain types of small cracks, which will become an issue over time (Kajari-Schröder et al., 2012).

Infrared (IR) imaging is a non-destructive, 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.

* Corresponding author at: Energy Campus Nuremberg, Fürther Str. 250, 90429 Nuremberg, Germany.
E-mail address: sergiu.deitsch@fau.de (S. Deitsch).

https://doi.org/10.1016/j.solener.2019.02.067
Received 8 July 2018; Received in revised form 9 February 2019; Accepted 26 February 2019
Available online 02 May 2019
0038-092X/ © 2019 International Solar Energy Society. Published by Elsevier Ltd. All rights reserved.

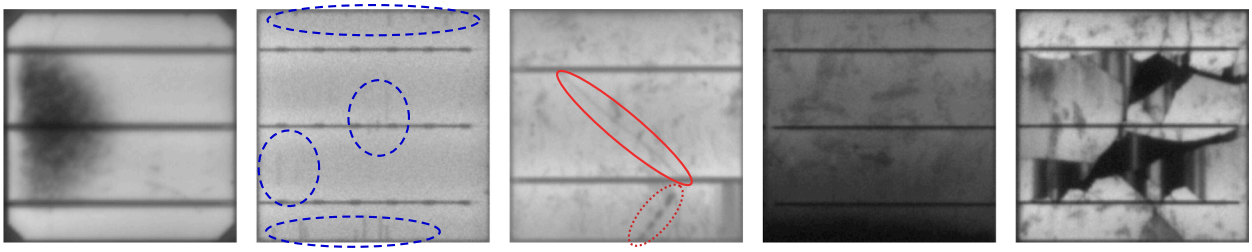


Fig. 1. Various intrinsic and extrinsic defects in monocrystalline ((a)–(b)) and polycrystalline ((c)–(e)) solar cells. (a) shows a solar cell with a typical material defect. (b) shows finger interruptions in the encircled areas, which do not necessarily reduce the module efficiency. The solar cell in (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 a cell with electrically separated or degraded parts, which are usually caused by mechanical damage.

Electroluminescence (EL) imaging (Fuyuki et al., 2005; Fuyuki and Kitiyanan, 2009) is another established non-destructive 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 1150 nm. The emission can be imaged by a silicon Charge-coupled Device (CCD) sensor. The high spatial image resolution enables the detection of microcracks (Breitenstein et al., 2011), 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 (Fuyuki and Kitiyanan, 2009): (1) intrinsic deficiencies due to material properties such as crystal grain boundaries and dislocations, and (2) process-induced extrinsic defects such as microcracks and breaks, which reduce the overall module efficiency over time.

Fig. 1 shows an example EL image with different types of defects in monocrystalline and polycrystalline solar cells. Fig. 1(a) and (b) show general material defects from the production process such as finger interruptions which do not necessarily reduce the lifespan of the affected solar panel unless caused by high strain at the solder joints (Köntges et al., 2014). Specifically, the efficiency degradation induced by finger interruptions is a complex interaction between their size, position, and the number of interruptions (De Rose et al., 2012; Köntges et al., 2014). Fig. 1(c) to (e) show microcracks, degradation of cell-interconnections, and cells with electrically separated or degraded parts 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 enables a quick identification of defective areas and can potentially complement the prediction of future efficiency loss within a PV module. In this work, we propose two classification pipelines that automatically solve the second task, *i.e.*, to determine a per-cell defect likelihood that may lead to efficiency loss.

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. Particularly for aerial imagery, however, additional issues may arise and will need to be solved. Kang and Cha (2018) highlight several challenges that need to be addressed before applying our approach outside of a manufacturing setting.

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 (Dotenco et al., 2016). 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. In particular, we show how uncertainty can be incorporated into both frameworks to improve the classification accuracy. Third, we contribute an annotated dataset consisting of 2624 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.

Fig. 2 shows the assessment results of a solar panel using the proposed convolutional neural network. Each solar cell in the EL image is

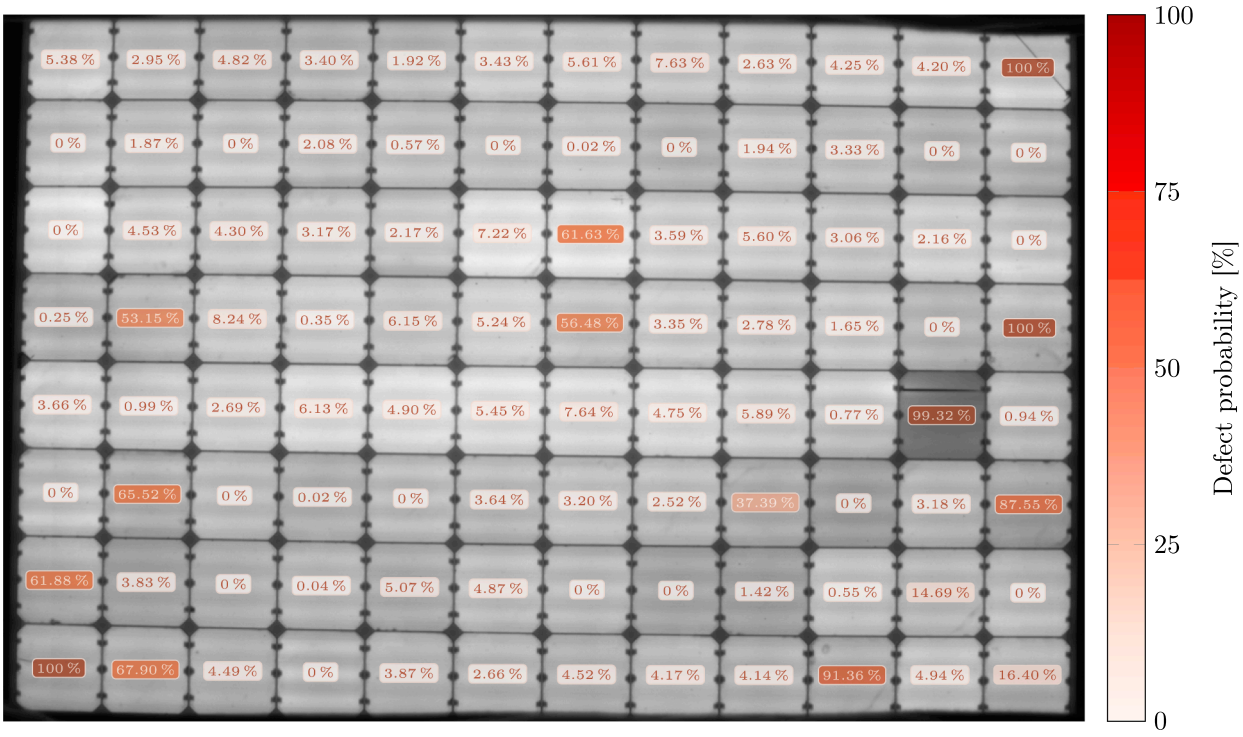


Fig. 2. Defect probabilities inferred for each PV module cell by the proposed CNN. A darker shade of red indicates a higher likelihood of a cell defect. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

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. Detection of surface abnormalities in EL images of solar cells is related to structural health monitoring. However, it is important to note that certain defects in solar cells are only specific to EL imaging of PV modules. For instance, fully disconnected solar cells simply appear as dark image regions (similar to Fig. 1(d)) and thus have no comparable equivalent in terms of structural defects. Additionally, surface irregularities in solar wafers (such as finger interruptions) are easily confused with cell cracks, even though they do not significantly affect the power loss.

In the context of visual inspection of solar modules, Tsai et al. (2012) 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 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. (2013) 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 cell cracks. This strategy is therefore 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 (2014) 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. (2015) 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. (2018), 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. Masci et al. (2012) 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 2281 training and 646 test images, the CNN architecture classifies steel defects at least twice as accurately as the SVMs. Zhang et al. (2016) 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. Cha et al. (2017) use a very similar approach to detect concrete cracks in a broad range of images taken under various environmental and illumination conditions. Kang and Cha (2018) employ deep learning for structural health monitoring on aerial imagery. Cha et al. (2018) additionally investigated defect localization using the modern learning-based segmentation approaches for region proposals based on the Faster R-CNN framework which can perform in real-time. Lee et al. (2019) also use semantic segmentation to detect cracks in concrete.

In medical context, Esteva et al. (2017) 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 2032 different diseases making it possible to achieve a high degree of accuracy.

3. Methodology

We subdivide each module into its solar cells, and analyze each cell individually to eventually infer the defect likelihood. 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 (Deitsch et al., 2018), which brings every cell into a normal form free of perspective and lens distortions.

Unless otherwise stated, the proposed methods operate on size-normalized EL images of solar cells with a resolution of 300×300 pixels. This image resolution was derived from the median dimensions of image regions corresponding to individual solar cells in the original EL images of PV modules. The solar cell images are used directly as pipeline input. The image resolution of solar cells in the wild will generally deviate from the required resolution and therefore must be adjusted accordingly. The CNN architecture sets a minimum image resolution, which typically equals the CNN's receptive field (*e.g.*, the original VGG-19 architecture uses 224×224). If the resolution is lower than this minimum resolution, then the image must be upscaled. For higher resolutions, the network can be applied in a strided window manner and afterwards the outputs are pooled together (typically using average or maximum pooling). We followed an alternative approach in which the CNN architecture encodes this process inherently. In case of the SVM pipeline, the resolution requirement is less stringent. Given local features that are scale-invariant, the image resolution of the classified solar cells does not need to be adjusted and may vary from image to image.

3.1. Classification using support vector machines

The general approach for classification using SVMs (Cortes and Vapnik, 1995) 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. Fig. 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 (Deitsch et al., 2018). 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.

3.1.2. Feature extraction

In order to train the SVMs, feature descriptors are extracted first. 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. Both strategies produce different sets of features that can be better suitable for specific types of solar wafers than others. Dense sampling disregards the image content and instead uses a fixed configuration of feature points. Keypoint detectors, on the other hand, rely on the textureness in the image and therefore the number of keypoints is proportional to the amount of high-frequency elements, such as edges and corners (as can be seen in Fig. 4(c) and (d)). Keypoint detectors typically operate in scale space, allowing feature detection at different scale levels and also at different orientations. Fig. 4(d) shows keypoints detected by KAZE. Here, each keypoint has a different scale (visualized by the radius of corresponding circles) and also a specific orientation exemplified by the line drawn from the center to the circle border. Keypoints that capture both the scale and the rotation are invariant to changes in image resolution and to in-plane rotations, which makes them very robust.

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. The number of feature locations only depends on the grid size. Dense sampling can be useful if computational resources are very limited, or if the purpose is to identify defects only in monocrystalline PV modules.

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

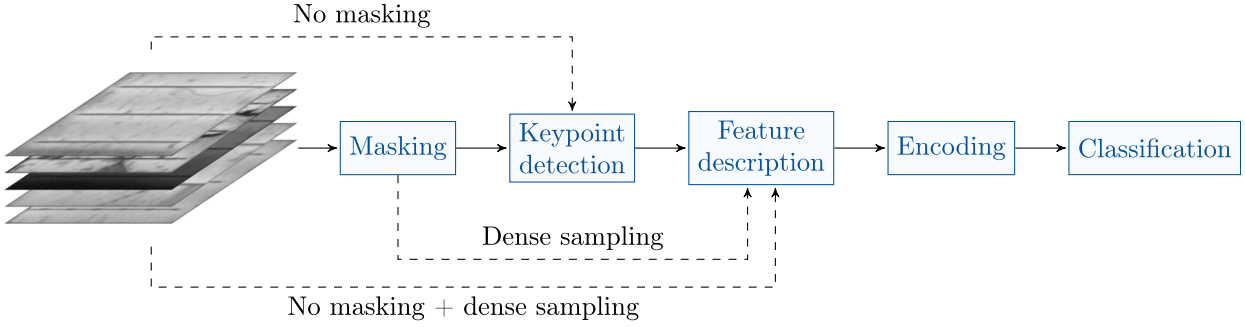
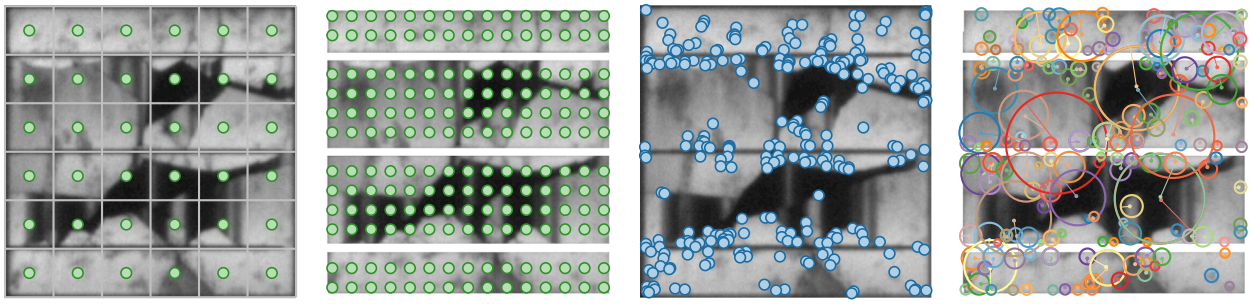


Fig. 3. An overview of the SVM classification pipeline with the four proposed variations of the preprocessing and feature extraction process.



(a) Dense sampling on a 6×6 grid *without* masking (b) Dense sampling on a 15×15 grid *with* masking (c) Keypoints from AGAST detector (d) Masked KAZE keypoints

Fig. 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.

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 (Mair et al., 2010)	✓	✗
KAZE (Alcantarilla et al., 2012)	✓	✓
HOG (Dalal and Triggs, 2005)	✗	✓
PHOW (Bosch et al., 2007)	✗	✓
SIFT (Lowe, 1999)	(✓)	✓
SURF (Bay et al., 2008)	(✓)	✓
VGG (Simonyan et al., 2014)	✗	✓

Feature Transform (SIFT) (Lowe, 1999), 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) (Bay et al., 2008) is a faster variant of SIFT, and also consists of a keypoint detector and 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 (Alcantarilla et al., 2012) 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) (Mair et al., 2010) 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) (Rosten and Drummond, 2005, 2006).

Among the dedicated descriptors, Pyramid Histogram of Visual Words (PHOW) (Bosch et al., 2007) is an extension of SIFT that computes SIFT descriptors densely over a uniformly spaced grid. We use the implementation variant from VLFEAT (Vedaldi and Fulkerson, 2008). Similarly, Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005) 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 (Simonyan et al., 2014). 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 (Heinly et al., 2012).

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 (Itseez, 2017) as of version 3.3.1. One notable exception is 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) (Jégou et al., 2012), which offers a compact state-of-the-art representation (Peng et al., 2015). VLAD encoding is sometimes also used for deep learning based features in classification, identification and retrieval tasks (Gong et al., 2014; Ng et al., 2015; Paulin et al., 2016; Christlein et al., 2017).

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 (Sculley, 2010) of k-means. The cluster centroids $\boldsymbol{\mu}_k$ correspond to anchor points of the dictionary. Afterwards, first order statistics are aggregated as a sum of residuals of all descriptors $\mathscr{X} := \{\boldsymbol{x}_t \in \mathbb{R}^d | t = 1, \ldots, T\}$ extracted from a solar cell image. The residuals are computed with respect to their nearest anchor point $\boldsymbol{\mu}_k$ in the dictionary $D := \{\boldsymbol{\mu}_k \in \mathbb{R}^d | k = 1, \ldots, K\}$ as

$$\boldsymbol{\nu}_k := \sum_{t=1}^{T} \eta_k(\boldsymbol{x}_t)(\boldsymbol{x}_t - \boldsymbol{\mu}_k) \tag{1}$$

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

$$\eta_k(\boldsymbol{x}) \coloneqq \begin{cases} 1 & \text{if } k = \underset{j=1,\ldots,K}{\arg\min} \|\boldsymbol{x} - \boldsymbol{\mu}_j\|_2 \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

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

$$\boldsymbol{v} \coloneqq (\boldsymbol{v}_1^\top, \ldots, \boldsymbol{v}_K^\top)^\top \ . \tag{3}$$

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 $v_i \in \boldsymbol{v}$ is normalized as

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

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 (2012) 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 $\tilde{\boldsymbol{v}} \coloneqq (\hat{\boldsymbol{v}}_1^\top, \ldots, \hat{\boldsymbol{v}}_m^\top)^\top \in \mathbb{R}^{mKd}$ is jointly decorrelated and whitened by means of PCA (Kessy et al., 2016). 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 (Fan et al., 2008), which is optimized for linear classification tasks and large datasets. For the non-linear RBF kernel, we use Libsvm (Chang and Lin, 2011).

The SVM hyperparameters are determined by evaluating the average F_1 score (van Rijsbergen, 1979) in an inner fivefold 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.*, $C_{\text{linear}} \in \{10^k | k = -2, \ldots, 6\} \subset \mathbb{R}_{>0}$. For RBF SVMs, the penalty parameter C is determined from a slightly smaller set $C_{\text{RBF}} \in \{10^k | k = 2, \ldots, 6\}$. 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 Vgg-19 network architecture (Simonyan and Zisserman, 2015) originally trained on the Imagenet dataset (Deng et al., 2009) using 1.28 million images and 1000 classes. We then refined the network using our dataset.

We replaced the two fully connected layers of Vgg-19 by a Global Average Pooling (GAP) (Lin et al., 2013) and two fully connected layers with 4096 and 2048 neurons, respectively (*cf.*, Fig. 5). The GAP layer is used to make the Vgg-19 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) 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 ± 3°. The translation is limited to ± 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 Imagenet model on our data to adapt the CNN to the new task, similar to Girshick et al. (2014). We, however, do

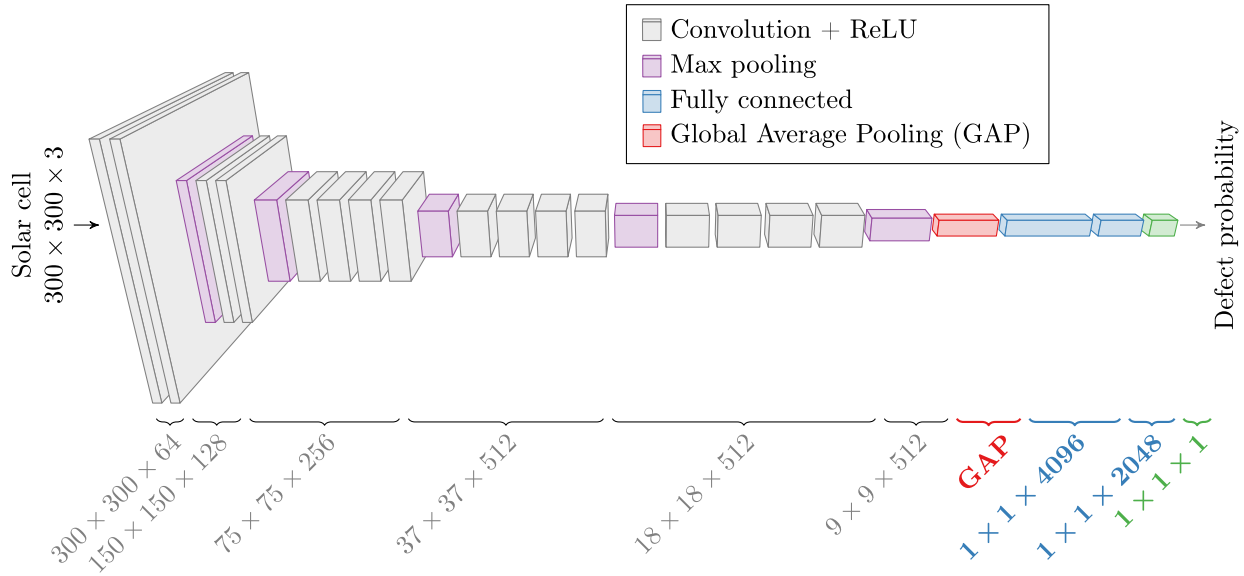


Fig. 5. Architecture of the modified Vgg-19 network used for prediction of defect probability in 300 × 300 pixels EL images of solar cells. Boldface denotes layers that deviate from Vgg-19.

this in two stages. First, we train only the fully connected layers with randomly initialized weights while keeping the weights of the convolutional blocks fixed. Here, we employ the Adam optimizer (Kingma and Ba, 2014) 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. We observed that fine-tuning the CNN in several stages by subsequently increasing the number of hyperparameters slightly improves the generalization ability of the resulting model compared to a single refinement step.

In both stages, we process the augmented versions of the 1968 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 196800 augmented variations of the original 1968 training samples that are used to refine the network. For the implementation of the deep regression network, we use Keras version 2.0 (Chollet et al., 2015) with Tensorflow version 1.4 (Abadi et al., 2015) 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. Finally, we compare the latter against the proposed deep CNN, and visualize the internal feature mapping of the CNN.

4.1. Dataset

We propose a public dataset[1] of solar cells extracted from high resolution EL images of monocrystalline and polycrystalline PV modules (Buerhop-Lutz et al., 2018). The dataset consists of 2624 solar cell images at a resolution of 300 × 300 pixels originally extracted from 44 different PV modules, where 18 modules are of monocrystalline type, and 26 are of polycrystalline type.

The images of PV modules used to extract the individual solar cell samples were taken in a manufacturing setting. Such controlled conditions enable a certain degree of control on quality of imaged panels and allow to minimize negative effects on image quality, such as overexposure. Controlled conditions are also required particularly because background irradiation can predominate EL irradiation. Given PV modules emit the only light during acquisition performed in a dark room, it can be ensured the images are uniformly illuminated. This is opposed to image acquisition in general structural health monitoring, which introduces additional degrees of freedom where images can suffer from shadows or spot lighting (Cha et al., 2017). An important issue in EL imaging, however, can be considered blurry (*i.e.*, out-of-focus) EL images due to incorrectly focused lens which can be at times challenging to attain. Therefore, we ensured to include such images in the proposed dataset (*cf.*, Fig. 1 for an example).

The solar cells exhibit intrinsic and extrinsic defects commonly occurring in mono- and polycrystalline solar modules. In particular, the dataset includes microcracks and cells with electrically separated and degraded parts, 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 familiar with intricate details of different defects in EL images. The criteria for such failures are summarized by Köntges et al. (2014). In their failure categorization, the expert focused specifically on defects with known power loss above 3% from the initial power output. The expert answered the questions (2) 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. Table 3 shows the distribution of ground truth solar cell labels, separated by the type of the source PV module.

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%

We used 25% of the labeled cells (656 cells) for testing, and the remaining 75% (1968 cells) for training. Stratified sampling was used to randomly split the samples while retaining the 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 (2001)

$$c_j := \frac{S}{2n_j}, \tag{5}$$

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 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 (*cf.* Table 2) or masking out the background region (*cf.* Section 3.1.1) improves the classifiers.

Performance is measured using the F_1 score, which is the harmonic mean of precision and recall. Fig. 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

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

Table 3
The distribution of the total number of solar cell images in the dataset depending on sample label p and the PV module type from which the solar cells were originally extracted. The numbers of solar cell images are given for the 75%/25% training/test split.

Solar water	Train				Test				Σ
	0%	33%	67%	100%	0%	33%	67%	100%	
Monocrystalline	438	87	41	249	150	30	15	64	1,074
Polycrystalline	683	132	37	301	237	46	13	101	1,550
Σ	1,121	219	78	550	387	76	28	165	2,624

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.

Fig. 7 shows the evaluated detector and extractor combinations for monocrystalline cells, polycrystalline 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 (Fawcett, 2006). 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

Fig. 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. Fig. 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

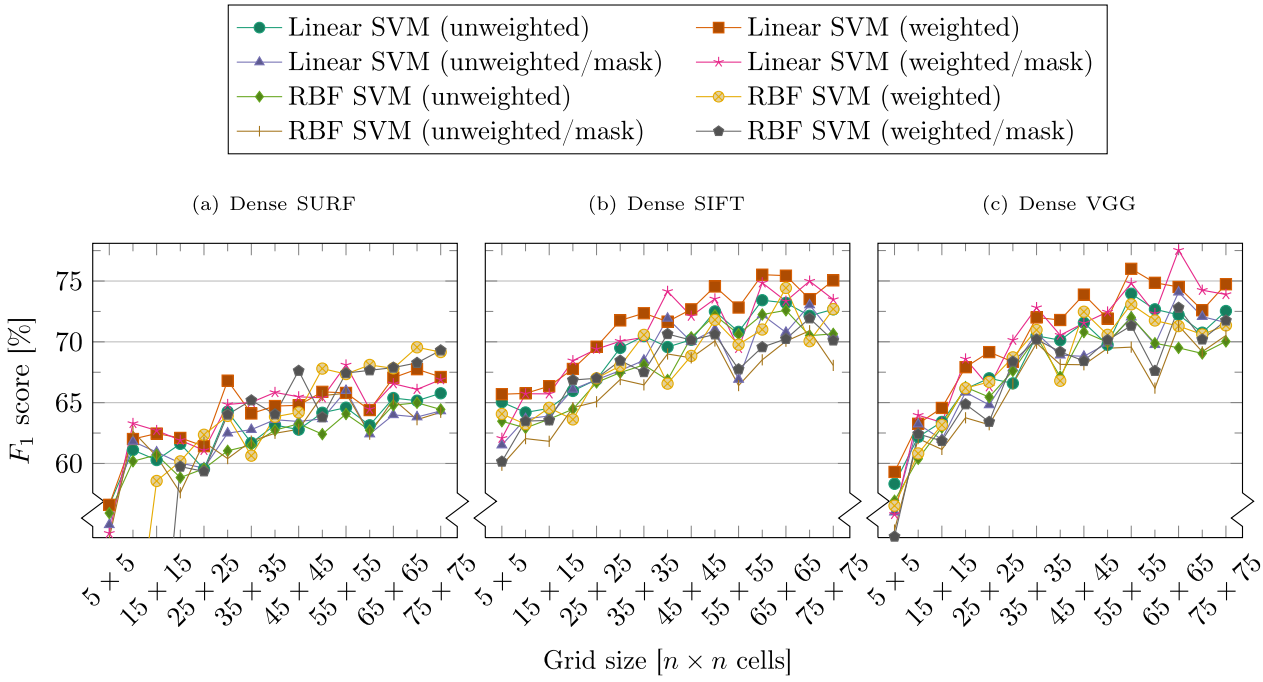


Fig. 6. Classification performance for different dense sampling configurations in terms of F_1 score grouped by the feature descriptor, classifier, weighting strategy, and the use of masking. The highest F_1 score is achieved using a linear SVM and the VGG feature descriptor at a grid resolution of 65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

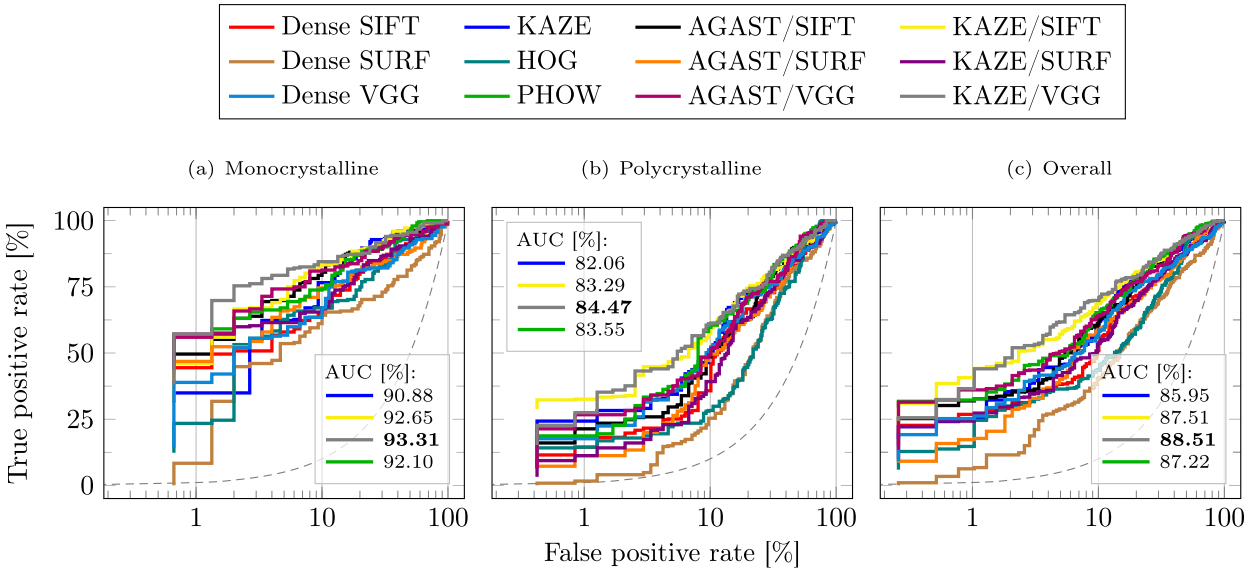


Fig. 7. Receiver Operating Characteristic (ROC) for top performing feature detector/extractor combinations grouped by mono-, polycrystalline, and both solar module types combined. The dashed curve (—) represents the baseline in terms of a random classifier. Note the logarithmic scale of the false positive rate axis. Refer to the text for details.

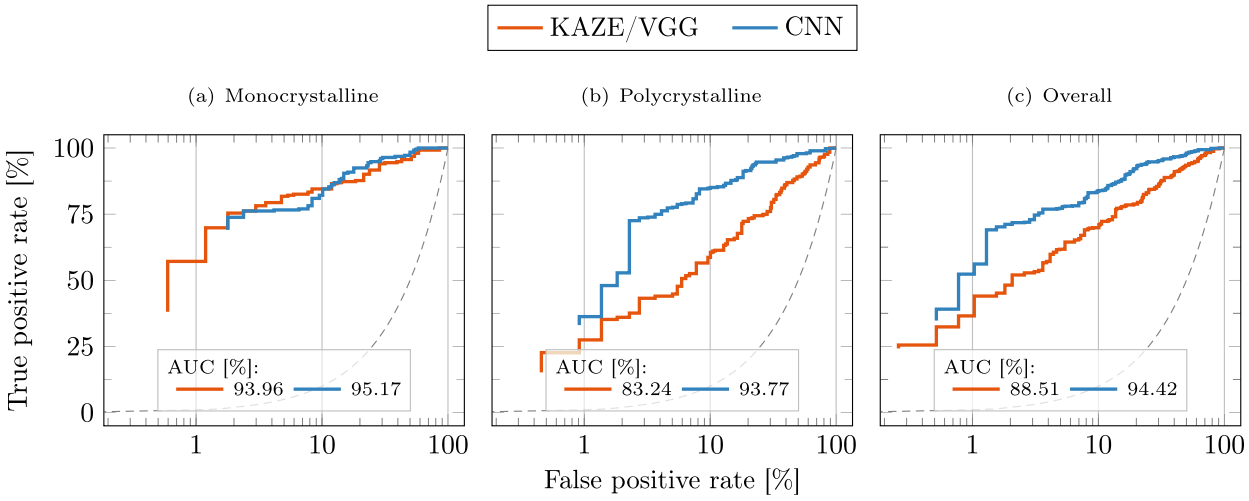


Fig. 8. ROC curves of the best performing KAZE/VGG feature detector/descriptor combination (——) compared to the ROC of the deep regression network (——). While in the monocrystalline case (a) the classification performance of the CNN is almost on par with the linear SVM. For polycrystalline PV modules (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 ra.ndom classifier.

abnormalities on homogeneous surfaces almost as accurate as a CNN trained on image pixels directly.

For polycrystalline PV modules, the CNN is able to predict defective solar cells almost 11% 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.

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. Model performance per defect category

Here, we provide a detailed report of the performance of proposed models with respect to individual categories of solar cells (*i.e.*, defective and functional) in terms of confusion matrices. The two dimensional confusion matrix stores the proportion of correctly identified cells (true negatives and true positives) in each category on its primary diagonal. The secondary diagonal provides the proportion of incorrectly identified solar cells (false negatives and false positives) with respect to the other category.

Fig. 9 shows the confusion matrices for the proposed models. The confusion matrices are given for each type of solar wafers, and their combination. The vertical axis of a confusion matrices specifies the expected (*i.e.*, ground truth) labels, whereas the horizontal one the labels predicted by the corresponding model. Here, the predictions of the CNN were thresholded at 50% to produce the two categories of functional (0%) and defective (100%) solar cells.

In regard to monocrystalline PV modules, the confusion matrices in Fig. 9(a) and (d) underline that both models provide comparable classification results. The linear SVM, however, is able to identify more defective cells correctly than the CNN at the expense of functional cells being identified as defective (false negatives). To this end, the linear SVM makes also less errors at identifying defective solar cells as being intact (false positives).

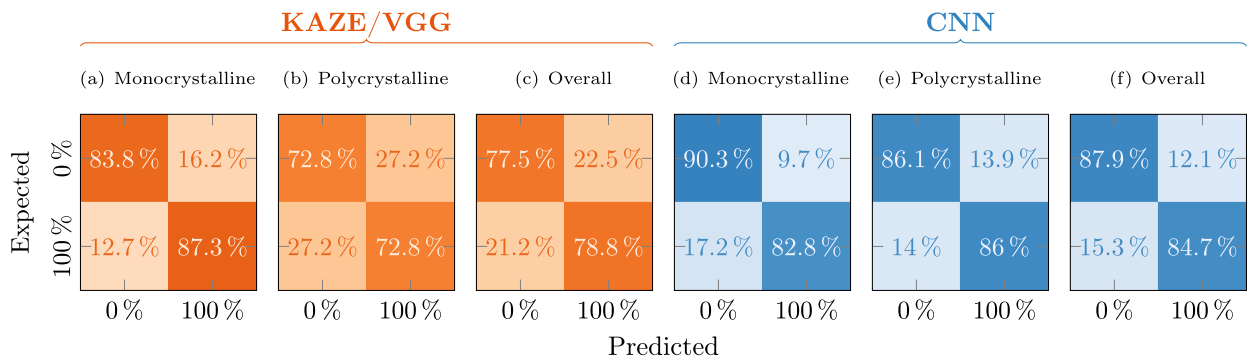


Fig. 9. Confusion matrices for the proposed classification models. Each row of confusion matrices stores the relative frequency of instances in the expected defect likelihood categories. The columns, on the other hand, contain the relative frequency of instances of predictions made by the classification models. Ideally, only the diagonals of confusion matrices would contain non-zero entries which corresponds to perfect agreement in all categories between the ground truth and the classification model. The CNN generally makes less prediction errors than an SVM trained on KAZE/VGG features.

In polycrystalline case given by Fig. 9(b) and (e), the CNN clearly outperforms the linear SVM in every category. This also leads to overall better performance of the CNN in both cases, as evidenced in Fig. 9(c) and (f).

4.6. Impact of training dataset size on model performance

For training both the linear SVM and the CNN a relatively small dataset of unique solar cell images was used. Given that typical PV module production lines have an output of 1500 modules per day containing around 90,000 solar cells, models can be expected to greatly benefit from additional training data. In order to examine how the proposed models improve if more training samples are used, we evaluate their performance on subsets of original training samples since no additional training samples are available.

To a infer the performance trend, we evaluate the models on three differently sized subsets of original trainings samples. We used 25%, 50% and 75% of original training samples. To avoid a bias in the obtained metrics, we not only sample the subsets randomly but also sample each subset 50 times to obtain the samples used to train the models. We additionally use stratified sampling to retain the distribution of labels from the original set of training samples. To evaluate the

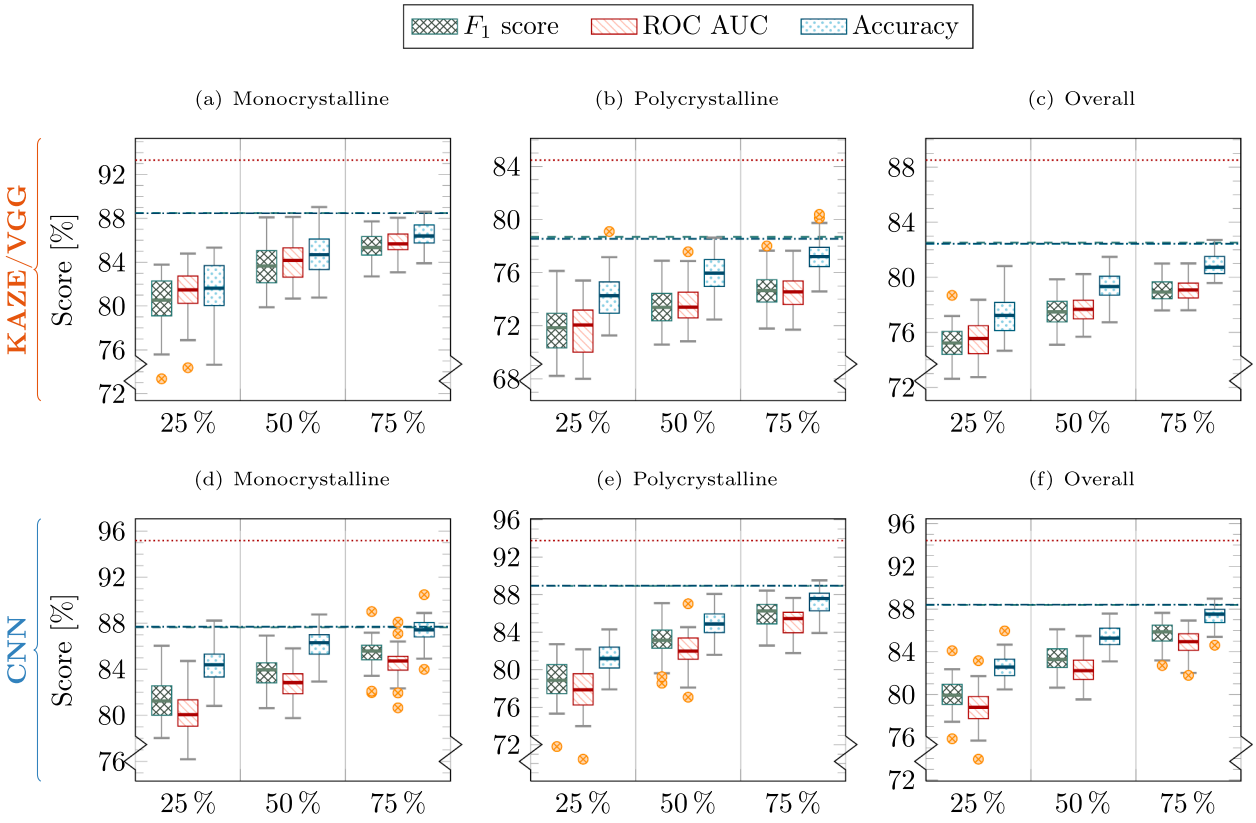


Fig. 10. Performance of the proposed models trained on subsets of original training samples. The results are grouped by the solar wafer type (left two columns) and the combination of both wafer types (last column). The first three plots in the top row show the distribution of evaluated metrics as boxplots for the linear SVM trained using KAZE/VGG features. The bottom row shows the results for the CNN. The horizontal lines specify the reference scores with respect to the F_1 measure (– – –), ROC AUC (·····), and the accuracy (–·–·–) of the proposed models trained on 100% of training samples. The circles (⊗) denote outliers in the distribution of evaluated metrics given by each boxplot. Increasing the number of training samples directly improves the performance of both models. The improvement is approximately logarithmic with respect to the number of t.raining samples.

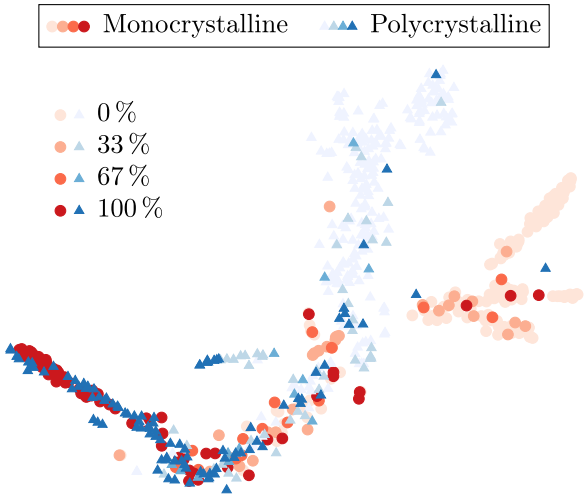


Fig. 11. *t*-SNE visualization of the CNN's last hidden layer output for the four defect probability classes quantized from predictions of the deep regression network. The 2048-dimensional output layer is mapped to a 2-D space for all 656 test images. This structure preserving 2-D projection of embeddings shows that similar cells defects are grouped together allowing the CNN to discern between various defects.

performance, we use the original test samples and provide the results for three metrics: F_1 score, ROC AUC, and accuracy.

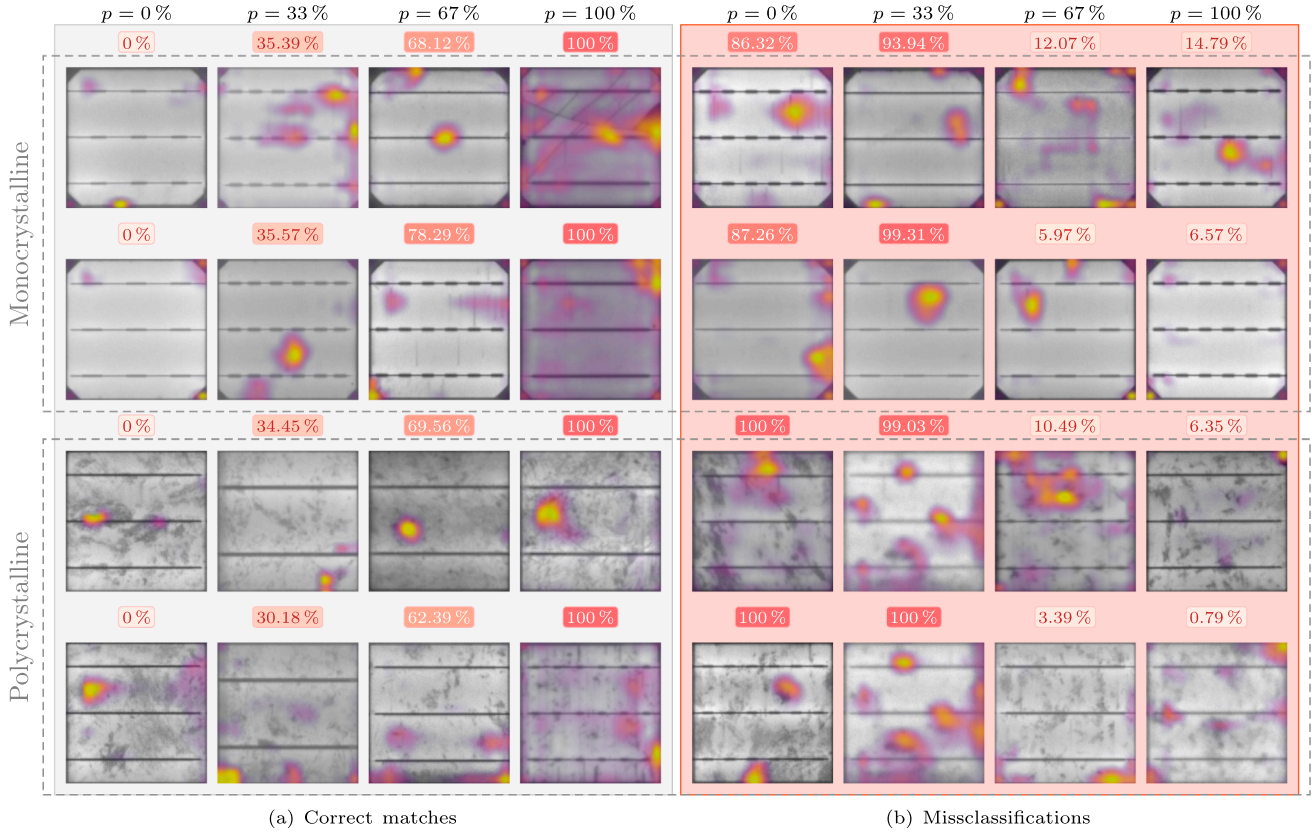
Fig. 10 shows the distribution of evaluated scores on all samples of the three differently sized subsets of training samples used to train the proposed models. The distribution of all 50 scores for each of the three subsets is summarized in a boxplot. The results clearly show that the performance of the proposed models improves roughly logarithmically with respect to the number of training samples which is typically observed in vision tasks (Sun et al., 2017).

4.7. Analysis of the CNN feature space

Here, we analyze the features learned by the CNN using *t*-distributed Stochastic Neighbor Embedding (*t*-SNE) (van der Maaten and Hinton, 2008), 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 (van der Maaten, 2014) 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. 11. Each point corresponds to a feature vector projected from the 2048-dimensional last layer of the CNN onto two dimensions. Projected feature vectors that were extracted from mono- and polycrystalline modules are color-coded in red and blue, respectively. The defect probabilities are encoded by saturation. The two dimensional representation preserves the structure of the high-dimensional feature space and shows that similar defect probabilities are in most cases co-located in the features space. This allows the CNN classifier to distinguish between EL images of defective and functional solar cells.

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



(a) Correct matches (b) Missclassifications

Fig. 12. Qualitative results of predictions made by the proposed CNN with correctly classified solar cell images (a) and missclassifications (b). Each column is labeled using the ground truth label. Red shaded probabilities above each solar cell image correspond to predictions made by the CNN. The upper two rows correspond to monocrystalline solar cells and bottom two rows to polycrystalline solar cell images.

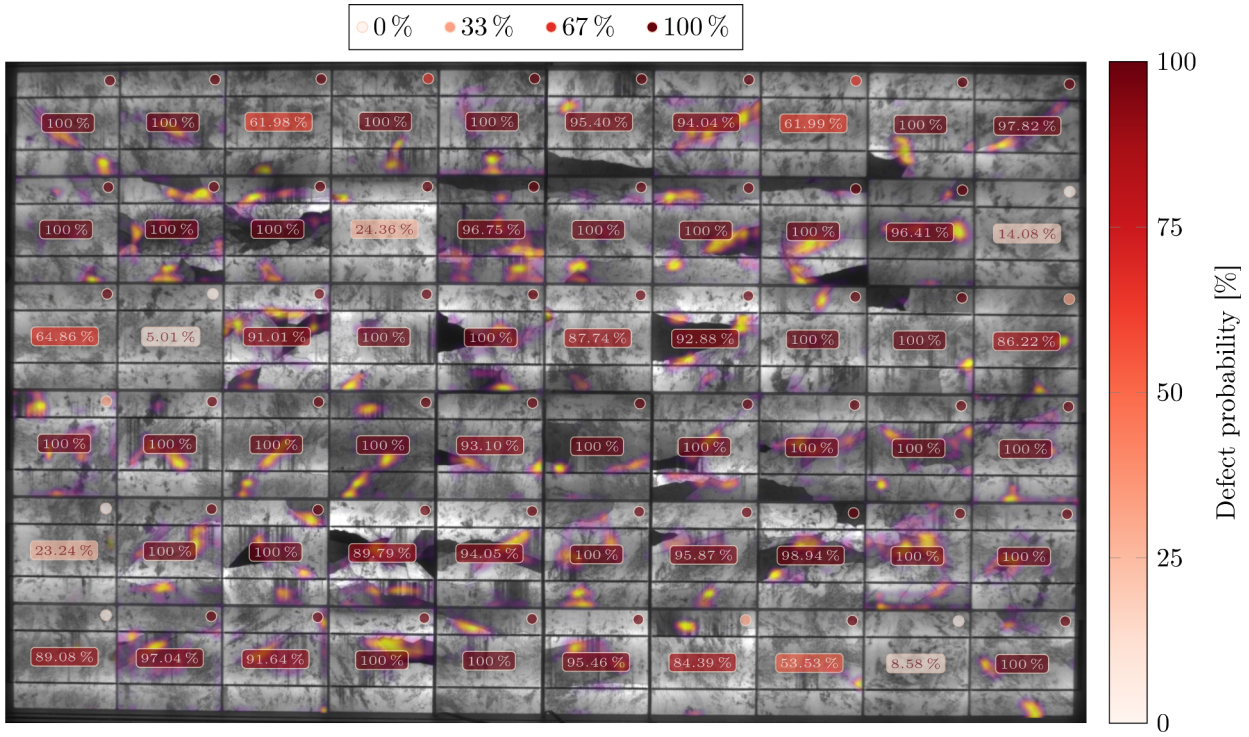


Fig. 13. Qualitative defect classification results in a PV module previously not seen by the deep regression network. The red shaded circles in the top right corner of each solar cell specify the ground truth labels. The solar cells are additionally overlaid by CAMs determined using Grad-CAM++ (Chattopadhay et al., 2018). The CAM for individual solar cells was additionally weighted by network's predictions to reduce the clutter. Notably, the network pays attention to very specific defects (such as fine cell cracks) that are harder to identify than cell cracks which a.re more obvious.

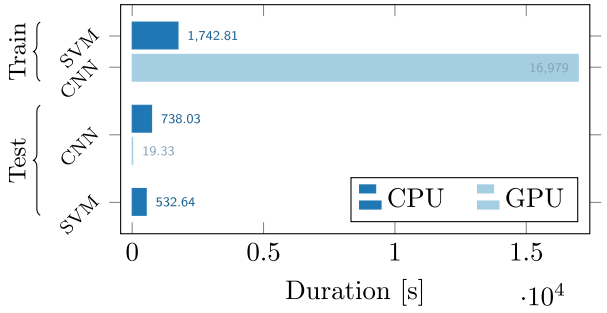


Fig. 14. Runtime of training and test phases for the proposed models. Training the SVM takes around 30 min, whereas training the CNN takes almost 5 h. The CNN is overall more efficient at inference (*i.e.*, testing) when running on the GPU requiring just slightly less than 20 s compared to over 12 min on the CPU. Our unoptimized implementation of the SVM pipeline completes in only 8 min.

which depend on the type of the source PV module. The overall appearance of the cell (*i.e.*, the number of soldering joints, textureness, *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.*, textureless) 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 Table 3). Additionally, the samples from these two categories can stimulate ambiguous decisions due to being at the boundary of clearly distinguishable defects and non-defects.

4.8. Qualitative results

Fig. 12 provides qualitative results for a selection of monocrystalline and polycrystalline solar cells with the corresponding defect likelihoods inferred by the proposed CNN. To allow an easy comparison to ground truth labels, the CNN defect probabilities are quantized into four categories corresponding to original labels by rounding the probabilities to nearest category. The selection contains both correctly as well as incorrectly classified solar cells having the smallest and largest squared distance, respectively, between the predicted probability and the ground truth label.

In order to highlight class-specific discriminative regions in solar cell images, Class Activation Maps (CAMs) (Zhou et al., 2016; Selvaraju et al., 2017; Chattopadhay et al., 2018) can be employed. While CAMs are not directly suitable for precise segmentation of defective regions particularly due to their coarse resolution. CAMs can still provide cues that explain why the convolutional network infers a specific defect probability. To this end, the solar cells in Fig. 12 are additionally overlaid by CAMs extracted from the last convolutional block ($18 \times 18 \times 512$) of the modified VGG-19 network and upscaled to the original resolution of 300×300 of solar cell images using the methodology by Chattopadhay et al. (2018).

Interestingly, even if the CNN incorrectly classifies a defective solar cell to be functional (*cf.*, last column in Fig. 12(b)), the CAM can still highlight image regions which are potentially defective. CAMs can therefore complement the fully automatic assessment process and provide decision support in complicated situations during visual inspection. One particular problem that can be witnessed from inspection of CAMs is that finger interruptions are not always clearly discerned from actual defects. This, however, can be managed by including corresponding samples to train the CNN.

In Fig. 13 we show the predictions of the CNN for a complete polycrystalline solar module. The ground truth labels are given as red shaded circles in the top right corner of each solar cell. Again, the solar

cells are overlaid by CAMs and additionally weighted by network's predictions to reduce the amount of visual clutter. By inspecting the CAMs it can be observed that the CNN focuses on particularly unique defects within solar cells that are harder to identify than more obvious defects such as degraded or electrically insulated cell parts (appearing as dark regions) in the same cell.

4.9. Runtime evaluation

Here, we evaluate the time taken by each step of the SVM pipeline and by the CNN, both during training and testing. The runtime is evaluated on a system running an Intel i7-3770 K CPU clocked at 3.50 GHz with 32 GB of RAM. The results are summarized in Fig. 14.

Unsurprisingly, training takes most of the time for both models. While training the SVM takes in total around 30 min. Refining the CNN is almost ten times slower and takes around 5 h. However, inference using CNN is much faster than that of the SVM pipeline and takes just under 20 s over 8 min of the SVM. It is, however, important to note that the SVM pipeline inference duration is reported for the execution on the CPU, whereas the duration of the much faster CNN inference is obtained on the GPU only. Additionally, only a part of the SVM pipeline performs the processing in parallel. When running the highly parallel CNN inference on the CPU, the test time increases considerably to over 12 min. Consequently, training the CNN on the CPU becomes intractable and we therefore refrained from measuring the corresponding runtime.

Considering the relative contributions of individual SVM pipeline steps, feature extraction is most time-consuming, followed by encoding of local features and clustering (*cf.*, Fig. 15). Preprocessing of features and hyperparameter optimization require the least.

In applications that require not only a low resource footprint but also must run fast, the total execution time of the SVM pipeline can be reduced by replacing the VGG feature descriptor either by SIFT or PHOW. Both feature descriptors substantially reduce the time taken for feature extraction during inference from originally 8 min to around 23 s and 12 s, respectively while maintaining a classification performance similar to the VGG descriptor.

4.10. Discussion

Several conclusions can be drawn from the evaluation results. First, masking can be useful if the spatial distribution of keypoints 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.

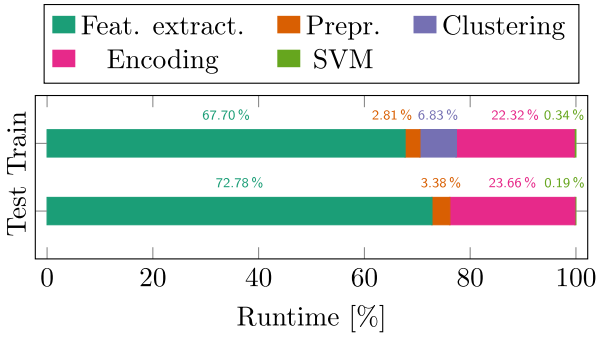


Fig. 15. Relative runtime contributions to training and test phases of the SVM pipeline. The most time-demanding step during SVM training and inference is the detection and extraction of KAZE/VGG features.

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.

Instead of predicting the defect likelihood one may want to predict specific defect types. Given additional training data, the methodology presented in this work can be applied without major changes (*e.g.*, by fine-tuning to the new defect categories) given additional training data with appropriate labels. Fine-tuning the network to multiple defect categories with the goal of predicting defect types instead of their probabilities, however, will generally affect the choice of the loss function and consequently the number of neurons in the last activation layer. A common choice for the loss function for such tasks is the (categorical) cross entropy loss with softmax activation (Goodfellow et al., 2016).

5. Conclusions

We presented a general framework for training an SVM and a CNN that can be employed for identifying defective 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 VGG-19, 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 Energy Campus Nuremberg (EnCN) and partially supported by the Research Training Group 1773 "Heterogeneous Image Systems" funded by the German Research Foundation (DFG).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL: https://www.tensorflow.org.

Alcantarilla, P.F., Bartoli, A., Davison, A.J., 2012. KAZE features. In: European Conference on Computer Vision (ECCV). Lect. Notes Comput. Sci., vol. 7577. pp. 214–227. https://doi.org/10.1007/978-3-642-33783-3_16.

Anwar, S.A., Abdullah, M.Z., 2014. Micro-crack detection of multicrystalline solar cells featuring an improved anisotropic diffusion filter and image segmentation technique. EURASIP J. Image Video Process. 2014, 15. https://doi.org/10.1186/1687-5281-2014-15.

Bay, H., Essa, A., Tuytelaarsb, T., Van Goola, L., 2008. Speeded-up robust features (SURF). Comput. Vis. Image Underst. 110, 346–359. https://doi.org/10.1016/j.cviu.

2007.09.014.
Bosch, A., Zisserman, A., Munoz, X., 2007. Image classification using random forests and ferns. In: International Conference on Computer Vision (ICCV), pp. 1–8. https://doi.org/10.1109/ICCV.2007.4409066.
Breitenstein, O., Bauer, J., Bothe, K., Hinken, D., Müller, J., Kwapil, W., Schubert, M.C., Warta, W., 2011. Can luminescence imaging replace lock-in thermography on solar cells? IEEE J. Photovolt. 1, 159–167. https://doi.org/10.1109/JPHOTOV.2011.2169394.
Buerhop-Lutz, C., Deitsch, S., Maier, A., Gallwitz, F., Berger, S., Doll, B., Hauch, J., Camus, C., Brabec, C.J., 2018. A benchmark for visual identification of defective solar cells in electroluminescence imagery. In: 35th European PV Solar Energy Conference and Exhibition, pp. 1287–1289. https://doi.org/10.4229/35thEUPVSEC20182018-5CV.3.15.
Cha, Y.-J., Choi, W., Büyüköztürk, O., 2017. Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil Infrastruct. Eng. 32, 361–378. https://doi.org/10.1111/mice.12263.
Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., Büyüköztürk, O., 2018. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Computer-Aided Civil Infrastruct. Eng. 33, 731–747. https://doi.org/10.1111/mice.12334.
Chang, C.-C., Lin, C.-J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 27:1–27:27. Software available at. http://www.csie.ntu.edu.tw/~cjlin/libsvm.
Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N., 2018. Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In: Winter Conference on Applications of Computer Vision (WACV), pp. 839–847. https://doi.org/10.1109/WACV.2018.00097.
Chollet, F., et al., 2015. Keras. GitHub URL: https://github.com/keras-team/keras.
Christlein, V., Gropp, M., Fiel, S., Maier, A.K., 2017. Unsupervised feature learning for writer identification and writer retrieval. In: International Conference on Document Analysis and Recognition (ICDAR), vol. 1. pp. 991–997. https://doi.org/10.1109/ICDAR.2017.165.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learn. 20, 273–297. https://doi.org/10.1007/BF00994018.
Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1. pp. 886–893. https://doi.org/10.1109/CVPR.2005.177.
Deitsch, S., Buerhop-Lutz, C., Maier, A., Gallwitz, F., Riess, C., 2018. Segmentation of Photovoltaic Module Cells in Electroluminescence Images, e-print. arXiv:1806.06530.
Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848.
De Rose, R., Malomo, A., Magnone, P., Crupi, F., Cellere, G., Martire, M., Tonini, D., Sangiorgi, E., 2012. A methodology to account for the finger interruptions in solar cell performance. Microelectron. Reliab. 52, 2500–2503. https://doi.org/10.1016/j.microrel.2012.07.014.
Dotenco, S., Dalsass, M., Winkler, L., Würzner, T., Brabec, C., Maier, A., Gallwitz, F., 2016. Automatic detection and analysis of photovoltaic modules in aerial infrared imagery. In: Winter Conference on Applications of Computer Vision (WACV). Springer, pp. 9. https://doi.org/10.1109/WACV.2016.7477658.
Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S., 2017. Dermatologist-level classification of skin cancer with deep neural networks. Nature 542, 115–118. https://doi.org/10.1038/nature21056.
Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J., 2008. LIBLINEAR: a library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874.
Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recogn. Lett. 27, 861–874. https://doi.org/10.1016/j.patrec.2005.10.010.
Fuyuki, T., Kitiyanan, A., 2009. Photographic diagnosis of crystalline silicon solar cells utilizing electroluminescence. Appl. Phys. A 96, 189–196. https://doi.org/10.1007/s00339-008-4986-0.
Fuyuki, T., Kondo, H., Yamazaki, T., Takahashi, Y., Uraoka, Y., 2005. Photographic surveying of minority carrier diffusion length in polycrystalline silicon solar cells by electroluminescence. Appl. Phys. Lett. 86. https://doi.org/10.1063/1.1978979.
Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587. https://doi.org/10.1109/CVPR.2014.81.
Gong, Y., Wang, L., Guo, R., Lazebnik, S., 2014. Multi-scale orderless pooling of deep convolutional activation features. In: European Conference on Computer Vision (ECCV), vol. 8695. pp. 392–407. https://doi.org/10.1007/978-3-319-10584-0_26.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.
Heinly, J., Dunn, E., Frahm, J.-M., 2012. Comparative evaluation of binary features. In: European Conference on Computer Vision (ECCV). Lect. Notes Comput. Sci., vol. 7573. pp. 759–773. https://doi.org/10.1007/978-3-642-33709-3_54.
Itseez, Open source computer vision library (OpenCV), 2017. URL: https://github.com/itseez/opencv.
Jégou, H., Ondřej, C., 2012. Negative evidences and co-occurences in image retrieval: the benefit of PCA and whitening. In: European Conference on Computer Vision (ECCV). Lect. Notes Comput. Sci., vol. 7573. pp. 774–787. https://doi.org/10.1007/978-3-642-33709-3_55.
Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C., 2012. Aggregating local image descriptors into compact codes. IEEE Trans. Pattern Anal. Mach. Intell. 34, 1704–1716. https://doi.org/10.1109/TPAMI.2011.235.
Kajari-Schröder, S., Kunze, I., Köntges, M., 2012. Criticality of cracks in PV modules. Energy Proc. 27, 658–663. https://doi.org/10.1016/j.egypro.2012.07.125.
Kang, D., Cha, Y.-J., 2018. Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging. Computer-Aided Civil Infrastruct. Eng. 33, 885–902. https://doi.org/10.1111/mice.12375.
Kessy, A., Lewin, A., Strimmer, K., 2016. Optimal Whitening and Decorrelation, e-print. arXiv:1512.00809.
King, G., Zeng, L., 2001. Logistic regression in rare events data. Polit. Anal. 9, 137–163. https://doi.org/10.1093/oxfordjournals.pan.a004868.
Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization, e-print. arXiv:1412.6980.
Köntges, M., Kurtz, S., Packard, C., Jahn, U., Berger, K., Kato, K., Friesen, T., Liu, H., Van Iseghem, M., 2014. Review of Failures of Photovoltaic Modules, Technical Report.
Lee, D., Kim, J., Lee, D., 2019. Robust concrete crack detection using deep learning-based semantic segmentation. Int. J. Aeronaut. Space Sci. https://doi.org/10.1007/s42405-018-0120-5.
Lin, M., Chen, Q., Yan, S., 2014. Network In Network, e-print. arXiv:1312.4400.
Lowe, D.G., 1999. Object recognition from local scale-invariant features. In: International Conference on Computer Vision (ICCV), vol. 2. pp. 1150–1157. https://doi.org/10.1109/ICCV.1999.790410.
Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G., 2010. Adaptive and generic corner detection based on the accelerated segment test. In: European Conference on Computer Vision (ECCV). Lect. Notes Comput. Sci., vol. 63. pp. 183–196. https://doi.org/10.1007/978-3-642-15552-9_14.
Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., Fricout, G., 2012. Steel defect classification with max-pooling convolutional neural networks. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–6. https://doi.org/10.1109/IJCNN.2012.6252468.
Mehta, S., Azad, A.P., Chemmengath, S.A., Raykar, V., Kalyanaraman, S., 2018. 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), pp. 333–342. https://doi.org/10.1109/WACV.2018.00043.
Ng, J.Y.H., Yang, F., Davis, L.S., 2015. Exploiting local features from deep networks for image retrieval. In: Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 53–61. https://doi.org/10.1109/CVPRW.2015.7301272.
Paulin, M., Mairal, J., Douze, M., Harchaoui, Z., Perronnin, F., Schmid, C., 2016. Convolutional patch representations for image retrieval: an unsupervised approach. Int. J. Comput. Vision 121, 149–168. https://doi.org/10.1007/s11263-016-0924-3.
Peng, X., Wang, L., Wang, X., Qiao, Y., 2015. Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. Comput. Vis. Image Underst. 150, 109–125. https://doi.org/10.1016/j.cviu.2016.03.013.
Rosten, E., Drummond, T., 2005. Fusing points and lines for high performance tracking. In: International Conference on Computer Vision (ICCV), pp. 1508–1515. https://doi.org/10.1109/ICCV.2005.104.
Rosten, E., Drummond, T., 2006. Machine learning for high-speed corner detection. In: European Conference on Computer Vision (ECCV). Lect. Notes Comput. Sci., vol. 3951. pp. 430–443. https://doi.org/10.1007/11744023_34.
Sculley, D., 2010. Web-scale k-means clustering. In: International Conference on World Wide Web (WWW), pp. 1177–1178. https://doi.org/10.1145/1772690.1772862.
Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: IEEE International Conference on Computer Vision (ICCV), pp. 618–626. https://doi.org/10.1109/ICCV.2017.74.
Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations.
Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Learning local feature descriptors using convex optimisation. IEEE Trans. Pattern Anal. Mach. Intell. 36, 1573–1585. https://doi.org/10.1109/TPAMI.2014.2301163.
Sun, C., Shrivastava, A., Singh, S., Gupta, A., 2017. Revisiting unreasonable effectiveness of data in deep learning era. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 843–852. https://doi.org/10.1109/ICCV.2017.97.
Tsai, D.-M., Wu, S.-C., Li, W.-C., 2012. Defect detection of solar cells in electroluminescence images using Fourier image reconstruction. Sol. Energy Mater. Sol. Cells 99, 250–262. https://doi.org/10.1016/j.solmat.2011.12.007.
Tsai, D.-M., Wu, S.-C., Chiu, W.-Y., 2013. Defect detection in solar modules using ICA basis images. IEEE Trans. Industr. Inf. 9, 122–131. https://doi.org/10.1109/TII.2012.2209663.
Tseng, D.-C., Liu, Y.-S., Chou, C.-M., 2015. Automatic finger interruption detection in electroluminescence images of multicrystalline solar cells. Math. Probl. Eng. 2015, 1–12. https://doi.org/10.1155/2015/879675.
van der Maaten, L., 2014. Accelerating t-SNE using tree-based algorithms. J. Mach. Learn. Res. 15, 3221–3245.
van der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605.
van Rijsbergen, C.J., 1979. Information Retrieval, 2nd ed. Butterworth-Heinemann.
Vedaldi, A., Fulkerson, B., 2008. VLFeat: An open and portable library of computer vision algorithms. URL: http://www.vlfeat.org.
Zhang, L., Yang, F., Daniel Zhang, Y., Zhu, Y.J., 2016. Road crack detection using deep convolutional neural network. In: International Conference on Image Processing (ICIP), pp. 3708–3712. https://doi.org/10.1109/ICIP.2016.7533052.
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016. Learning deep features for discriminative localization. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2921–2929. https://doi.org/10.1109/CVPR.2016.319.