PV-AIDED: Photovoltaic Artificial Intelligence Defect Identification. Multichannel encoder-decoder ensemble models for electroluminescence images of thin-film photovoltaic modules, PEARL TF-PV.

Evgenii Sovetkin*[1], Bart Pieters[1], Thomas Weber[2], Elbert Jan Achterberg[3], Arthur Weeber[4], Björn Rau[5], Marcus Rennhofer[6], and Mirjam Theelen[7]

[1]IEK5-Photovoltaics, Forschungszentrum Jülich, 52425 Jülich, Germany
[2]PI Photovoltaik-Institut Berlin AG, Wrangelstraße 100, 10997 Berlin, Germany
[3]Solar Tester, Vonderstraat 33A, 6365CR Schinnen, Netherlands
[4]Delft University of Technology, PVMD group, Mekelweg 4, 2628 CD Delft, Netherlands
[5]Helmholtz-Zentrum Berlin für Materialien und Energie GmbH/PVcomB, Schwarzschildstr. 3, 12489 Berlin, Germany
[6]AIT, Austrian Institute of Technology, Donau-City-Strasse 1, 1220 Vienna, Austria
[7]TNO, High Tech Campus 21, 5656AE Eindhoven, Netherlands

Topic: 5.3 PV Systems and Storage — Modelling, Design, Operation and Performance

Abstract

The Solar-Era.net project PEARL TF-PV, [1], aims to reduce the uncertainties in the operation of thin-film solar power plants. To this end, one of the main parts of the project is the gathering of performance data and electroluminescence (EL) images of different types of thin-film solar cells and modules (see abstract of Mirjam Theelen et al, this conference). Detailed, local information on the module performance is obtained using EL imaging, which may provide early warning signs of degradation. A large number of samples (over 6000 modules) are analyzed, ranging from cells and modules produced in the different laboratories of the project partners to industrially produced modules used in power plants. Measurements are performed in laboratories as well as outdoor directly at the power plants location. All gathered data is stored in a database that in turn is used to develop a failure catalogue for thin-film modules that describes typical defects, visible with EL in various technologies, and their influence on the solar modules reliability and lifetime.

In this work we present a novel image segmentation approach, aiming to identify commonly occurring defects in thin-film modules. We are building on top of the encoder-decoder neural networks framework, that have established itself as a standard tool in many other image processing applications.

We demonstrate our software, PV-AIDED, is capable of fully automatic and fast EL image processing of full-sizes modules. We are able to reliably identify frequently occurring defects in thin-film modules, such as shunts and so called “droplets”. The framework is general and applicable to other types of defects, other types of PV images, as well as other types of PV technology.

Keywords: encoder-decoder neural networks, thin-film, electroluminescence imaging

1 Introduction

Recently there has been an increasing interest in automated image analysis of spatially resolved characterization methods for photovoltaic (PV) modules such as electroluminescence (EL) [2, 3, 4, 5, 6, 7, 8]. Such automated image analysis aims at quality control of modules and is thus of great interest for manufacturers, PV system owners, and insurance companies, as it allows for a systematic inspection of a large number of modules, both prior and after installation.

An automated image analysis allows the systematic analysis of a large number of module images. Thus, it can greatly contribute to classify degradation modes of modules and to the identification of early warning signs or degradation.

In this work we discuss one of the segmentation methods, that allows to assign a label to every pixel in an image. For example, this allows to identify pixels in EL images that correspond to some defect in a module. This work extends our previous work, [8], and we present here the so-called ensemble model framework for image segmentation.

We demonstrate our software, PV-AIDED, is capable to reliably detects various defect types. Furthermore, a combined statistical evaluation of the EL image database reveals hidden features in EL images that are not observed in individual images of the modules. Our methods can be easily adapted for other types of defects, as well as other types of technology.

The paper is organized as follows. Section 2 reviews literature on the subject of automatic image analysis. The available data used in this study and its preprocessing is discussed in Section 3. Section 4 elaborates on our methodology. Section 5 and 6 discusses the results and technical details of our implementation. Lastly, this work is concluded in Section 7.

2 Literature

In photovoltaics automated image analysis methods aim to solve different tasks. Although very different aims are pursued, there are some common methods applied. For this reason we briefly review related works on image analysis which can be roughly categorized as follows; detecting and locating defects and other structures of interest, forecasting module performance, and image collection itself.

Most works on locating and identifying structures of interest revolves around cracks in crystalline silicon solar modules. Works like [9, 10, 11, 12, 13] apply methods like anisotropic diffusion, Fourier transform or independent component analysis to filter images and identify areas with cracks-like structures.

Deitsch et al. [4] train an SVM classifier based on the extracted feature descriptors (SURF, KAZE, FAST), and a VGG-net based neural network in order to identify defective cells that have an impact on power reduction of the whole module.

Demant et al. [2] proposes a Convolution Neural Network (CNN) architecture to forecast IV characteristics from a PL image as a production process control procedure.

When it comes to an application of these automatic methods to the real data, several practical problems arise. It is often the case that images taken in field conditions suffer from various distortions due to the position of a module in front of a camera, lens distortions, blurring due to wind and shocks. Such distortions introduce complications in automatic image processing. Therefore, a certain amount of work has been done in the direction of EL image preprocessing analysis methods [3, 6, 7].

The encoder-decoder deep neural networks is being applied very often in applications other than photovoltaics. Waldner and Diakogiannis [14] use encoder-decoder neural networks to extract agriculture field boundaries from satellite images. Iglovikov and Shvets [15] use U-net with VGG11 encoder network to segment satellite images in the Inria Aerial Image Labeling Dataset [16]. Havaei et al. [17] performs semantic segmentation of the brain tumours using MRI imaging. They explore the possibility to combine a simple CNN in a cascaded fashion. Esteva et al. [18] use deep neural network to classify different types of skin cancer. Attia et al. [19] use combination of CNN and Recurrent Neural Networks (RNN) to identify a surgical tool location in medical imaging. Kayalibay, Jensen, and Smagt [20] adapt the U-net encoder-decoder style architecture for the 3-dimensional input signal of the MRI images.

Recently the encoder-decoder networks we have applied to thin-film modules, [8]. In this work a series of encoder-decoder networks have been trained and a method to select the best model has been proposed. To this end we used a multi-objective optimisation as a first step of the procedure where we select all models lying on a Pareto frontier. After this we select the model the highest average Jaccard index.

In this paper we extend our approach in [8] by combining multiple trained models into a single ensemble model.

3 Data

The data was acquired within the framework of the PEARL-TF project. The website [1] contains detailed information about the project and the involved partners. In this project, the data from several solar parks with thin-film modules was collected. In addition to EL images, also performance characteristics of the modules were measured.

The EL images are taken at predefined conditions (selected fixed applied current and/or fixed applied voltage). A silicon CCD sensor camera is used to measure subsequently several parts of the module, with the images being stitched afterwards. The applied voltage and the applied current together with the temperature of the module are being recorded. The I/V characteristics are also measured and the solar cell performance parameters determined.

The database contains 6000 EL images of the co-evaporated Copper Indium Gallium Diselenide (CIGS) modules from the same manufacturer. Every image is supplied with a measured performance data. A typical EL image of a thin-film CIGS module from our database is depicted in Figure 1. The module consists of 150 connected cells in series (in Figure 1 the cells are recognized as horizontal stripes). The cells are separated by interconnection lines (horizontal dark lines in Figure 1). In addition, the module is separated in 5 parallel sub-modules by vertical isolation lines (dark vertical lines).

As mentioned before, every EL image consists of several stitched images. Different stitched parts of the image have different overall intensities (see Figure 1). This is attributed to the metastable behaviour of CIGS solar cells, where the electrical properties of the cell can change during the measurement.

In order to obtain a labelled dataset we segment images manually. This work is done using two different image editor programs. Firstly, we use the GNU[1] Image Manipulation Program (GIMP) [21] to create binary masks of various defects locations, where the de-

---

[1] GNU is a recursive acronym for GNU's Not Unix!

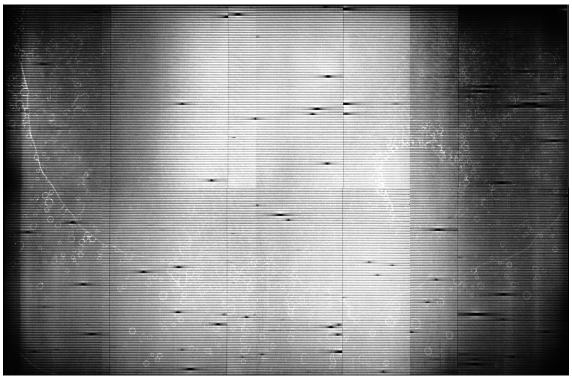

Figure 1: Thin-film module EL image. A module consists of 150 cells (positioned horizontally) connected in series. The cells are separated by interconnection lines (horizontal dark lines). The module consist of several submodules separated by vertical isolation lines, which appear dark in the EL image. The EL image is stitched (there are 1 horizontal and 3 vertical stitch lines); overall intensities of different patches of images are different. These intensity differences are attributed to metastable changes during the measurement.

fect pixels are manually marked using a drawing pad and a digital stylus. Alternatively, we use the ThinFia [22] program that is designed to identify defects in EL images by introducing a grid-mesh. The ThinFia program was developed within the PEARL-TF project. A general image processing program such as GIMP requires more time to segment an image, comparing to the ThinFia, however, smaller defects are segmented more accurately in GIMP.

In the image database we focus on the segmentation of “shunts”. Shunts are characterized by a more conductive connection between the front and back electrodes than the normal solar cell structure (i.e. the solar cell structure is damaged or missing). There are many causes for shunts. Commonly shunts originate from debris of the copper evaporation source or pinholes in the CIGS absorber [23, 24]. Figure 2 (right) depicts a shunt defect. Shunts generally appear as dark areas with a gradient in intensity away from the actual defect location. The dark area is confined to the area of one cell. Severe shunts may also completely darken a cell stripe, in which case often the neighboring cells exhibit bright areas in the vicinity of the shunt [25]. Shunts are generally relevant to the solar module performance, in particular under low light conditions.

In addition to shunts we noticed the CIGS modules often exhibit “droplets” in the EL images. Figure 2 (left) shows a detail of droplets. The appearance of droplets resembles water stains and thus we speculate these structures originate from the chemical bath deposition. At this point it is unknown what the impact of droplets is on the module performance, however, the bright appearance imply a local change in quantum efficiency according to the reciprocity relations between luminescence and quantum efficiency [26].

In total, we have about 6000 unlabelled, 142 labelled module images with shunts, and 14 labelled module images with droplets. The manual segmentation of droplets in an image is particular laborious, hence only few images are available.

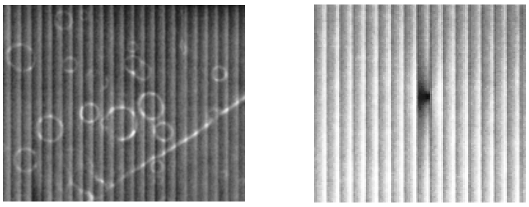

Figure 2: Example of droplets (left) and a shunted area (right). Here the cells are shown vertically.

All labelled images are split randomly onto a training and a testing datasets. The training dataset consists 106 labelled shunts and 8 labelled droplets images. The testing dataset contains 24 labelled shunt and 3 droplets images. In addition we evaluate the final model using the remaining 12 images with shunts and 3 images with droplets.

4 Methods

In this section we review methods that we use to build segmentation models. The *encoder-decoder* deep neural network architectures, [27], are commonly used in the semantic image segmentation problems, [20, 19, 15, 14]. We build multiple models by combining different encoder and decoder parts in the neural network architecture.

The encoder-decoder neural networks architecture consists of two parts: the contraction part (or encoder) and the symmetric expansion part (or decoder). The encoder compresses information content of an arbitrarily high-dimensional image into a feature vector. The decoder gradually upscales the encoded features back to the original resolution.

For the encoder part of our networks we use Mobilenet [28], ResNet [29], VGG-net [30] and U-net [31]. For the decoder part of our network we use U-net [31], FCN-net [32], PSP-net [33] and SegNet [34].

Different encoder can be mixed and matched with different decoders. Combinations of encoders and decoders provide us with different segmentation networks and we call these models as basis models. We train all the combinations of encoder-decoder network and create an ensemble model using a selection of them.

The ensemble model is constructed by combining outputs of basis encoder-decoder models together with the original image into a single multichannel image, and training another encoder-decoder network on the resulting image. This approach is similar to the so-called cascaded neural networks discussed in [17].

It should be noted here that a special care should be taken with the choice of the normalization transformation for the ensemble model input. In our case we work with binary segmentation images, and hence normalisation is performed per channel. This allows to avoid unnecessary interactions that can happen if normalisation is performed for all channels.

Figure 3 schematically demonstrates a structure of our ensemble encoder-decoder architecture. The input (on the left side) consist of a multichannel image, where each channel is either an image or the output of one of the encoder decoder networks. Each block in the middle represents an output of the convolutional layer, with the data flow going from left to right. The arrows represent the skipping links, where the input for the layer is copied from the encoder to the decoder parts. Different networks may have different number of layers, skipping connections and different activation functions.

To evaluate the quality of the model we use various performance metrics: Jaccard index, as well as connected component based precision and recall metrics. Such metrics take values in the interval $[0, 1]$, where 1 indicates the best results. For more details and discussion on the choice of metrics we refer to [8].

The ensemble model has several advantages over basis encoder-decoder networks. Firstly, there is always a trade-off between precision and recall metrics in models. Basis encoder-decoder networks can either identify defects accurately, but miss many of them (precision is high, recall is low), or identify many defects, but having a big ratio of false-positive identifications (precision is low, recall is high). Ensemble model allows to combine multiple models together and improves this trade-off balance. Therefore, we choose our models in ensemble from a pareto frontier of the basis models. Secondly, our basis encoder-decoder networks usually work with small image patches of $\approx 256 \times 256$. This imposes a restrictions on the size of a defect that can be detected, as well as amount information incorporated in decision making. Ensemble models allows to apply models on a larger input image patch, that provides a more global outlook of a surroundings of a defect. Thirdly, one may combine models that identify different defects that allows to learn the interaction between effects for defects occurring in the same region. And lastly, our implementation of ensemble model is general, and thus allows to use many types of images as channels, such as features computed from the input image, other segmentation methods (e.g. [35, 36]), or other measuremens (under different condistions or combining different imaging methods). The latter option is particularly interesting as it may introduce additional information not present in a single image.

5 Results

To compute evaluation of a full sized module image we compute evaluation on a set of overlapping subimages, where the size of each subimage equals to the selected model input image size.

We select a set of best performing basis models from the computed Pareto frontier, and use this set in the ensemble model. For the droplet model we select the following 7 models: ResNet-SegNet, U-net, VGG-net-SegNet, VGG-net-U-net, FCN-net-ResNet. For the shunts model we select the following 3 models: PSP-net, SegNet, U-net.

Table 1: Comparison of Jaccard index values for the best basis and ensemble models for droplets and shunts

	Basis model	Ensemble model
Droplets	0.27	0.31
Shunts	0.24	0.26

We train ensemble model on input patches with size 512 pixels. The basis encoder-decoder models that are used ensemble all have input image patch of 256 pixels. Large input image allows to incorporate more global information about defects and their neighbourhood.

For the ensemble model we select the multichannel version of the VGG-net-U-net network. We train the model for 100 epoch with 512 gradient descent steps in each epoch.

Figures 4 show comparison of the Pareto frontiers for basis encoder-decoder models (red line) and the Pareto frontier for the ensemble model (blue line). Each dots on the plots correspond to a single model, with x-axis showing a precision and y-axis showing a recall metrics, as defined in [8]. The Pareto frontier for the ensemble models consists of training checkpoints.

Figures 4 demonstrate that ensemble model improve upon the basis models, as the Pareto frontier for ensemble models is located above the Pareto frontier for the basis models.

Table 1 compares values of the Jaccard index between the best basis model and the best ensemble model. Those points are indicated with a red circle on Figure 4.

Figure 5 shows an example of model segmentation. This compares segmentation of the best basis model and the ensemble model. The top row shows the segmentation with the best basis model, and the bottom row shows the segmentation with the ensemble model. It is clear that differences are rather subtle, however the proposed metrics are able to capture it.

We apply the segmentation model for droplets and shunts for each EL image in our database. By evaluating an average of the computed binary segmented image, we obtain so-called heat maps. Figures 6 and 7 depict heat maps based on 6000 EL images for droplet and shunts locations. The brighter areas correspond to locations where defects have a higher probability of occurrence. The heat maps are given with a scale that maps pixel value in heat map images to a probability of shunts or droplets in that pixel. Note, that the scale is logarithmic in Figure 7, with the 1% of the brightest observations being squeezed in the white color.

The droplets in Figure 6 expresses a clear structure where droplets are distributed along a broad arc along three edges of the module. In the brighter areas of the arc, the probability that a pixel is marked as a droplet is about 2.5% (148 times in 6000 images). The droplets do not occur in the center of the modules. Furthermore, there are several dark lines where fewer droplets are detected. The vertical and horizontal dark

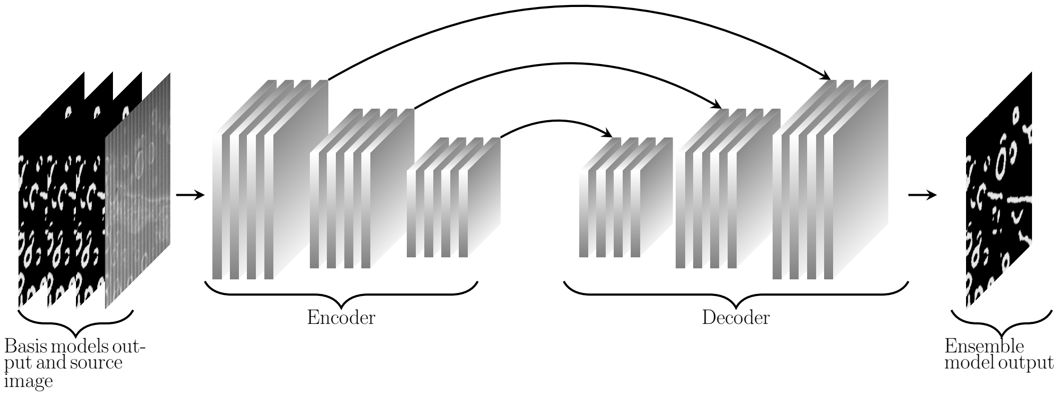

Figure 3: Schematic representation of ensemble encoder-decoder architecture. The left-hand side image is an input image patch together with the segmentation output of basis models that are passed to a series of the computational layers. The right-hand side image is an output binary image. The arrows are skipping connection layers, where input is being copied directly from encoder to a decoder.

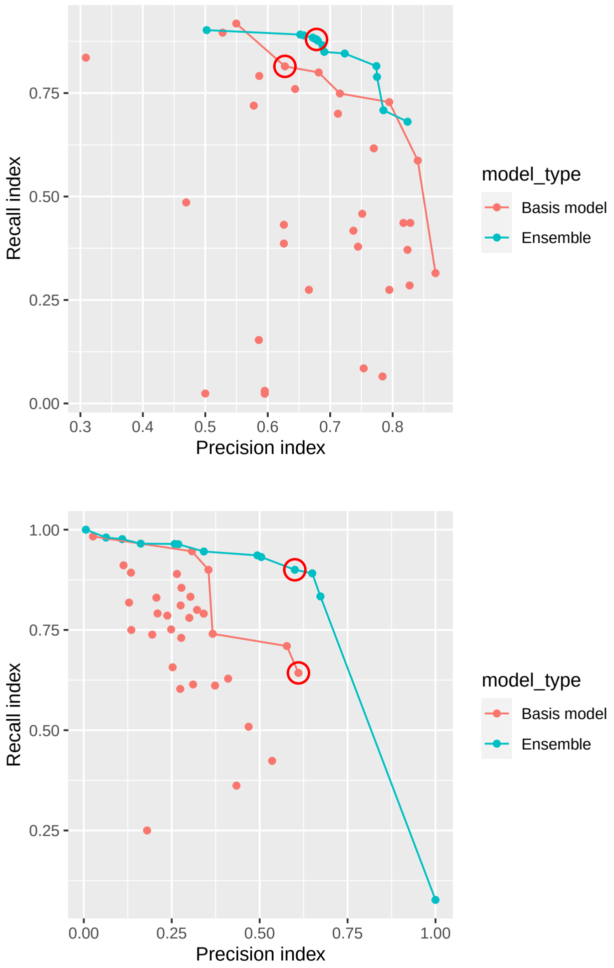

Figure 4: Comparison of the Pareto frontier for basis models (red line) and ensemble models (blue line) for droplets (top) and shunts (bottom). Individual dots correspond to individual models.

lines appear to be interference of vertical isolation lines in the modules as well as stitching lines in the image. However, the diagonal lines do not correspond to any obvious structure in the images that may interfere with droplet detection. We infer the diagonal lines have a physical origin.

The heat map of shunts is shown in Figure 7. There is a great number of features to be seen, such as a clear banded structure, high concentrations at certain edges and locations, such as at the bottom edge where at the isolation lines high concentrations of shunts are detected. Two cell stripes in the bottom half of the module are more often shunted. The stitching lines do not show up and thus do not seem to interfere with shunt detection. The isolation lines are associated with more detected shunts. However, only parts of the isolation lines exhibit a larger concentration of shunts and not all isolation lines are equally affected. For this reason we believe the higher shunt probability around the isolation lines is no artifact. There is a bright vertical line in the center. This line does not correspond to the position of an isolation line or stitching line. Note that in Figure 7 a slightly darker vertical line is visible at the same position. However, in this example no shunts are detected along this line. In other EL images this darker line is not present (e.g. in Figure 1). The origin of this line is unclear.

We would like to note that many features in Figures 6 and 7 are rather subtle in that these features only become visible when analyzing a large number of images. Furthermore, some of these features are quite certainly performance and reliability relevant (e.g. positions where shunts are likely to occur). We thus believe the extraction of such features can give manufacturers a better insight in their production process and thus contribute to process optimization and quality control.

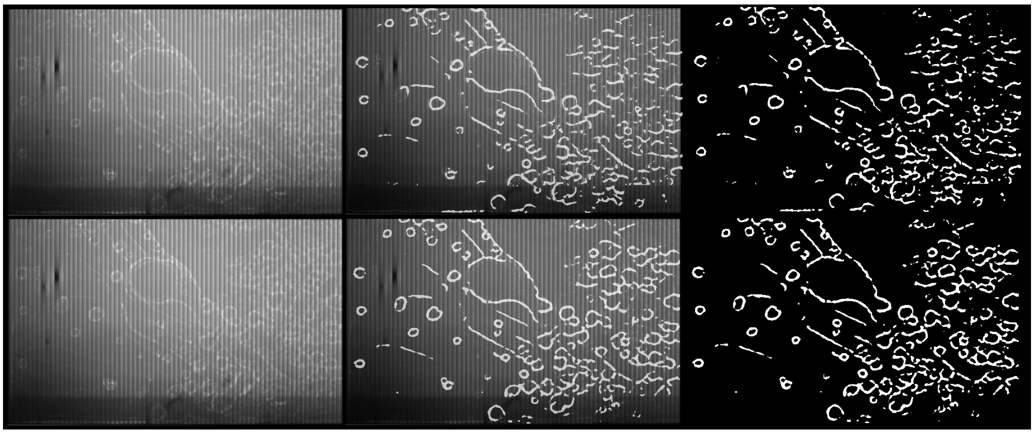

Figure 5: Visual comparison of the best basis model (top row), and the ensemble model (bottom row). The left column shows the original image, the right column shows the segmentation image, the middle column shows overlayed original image with the segmentation image.

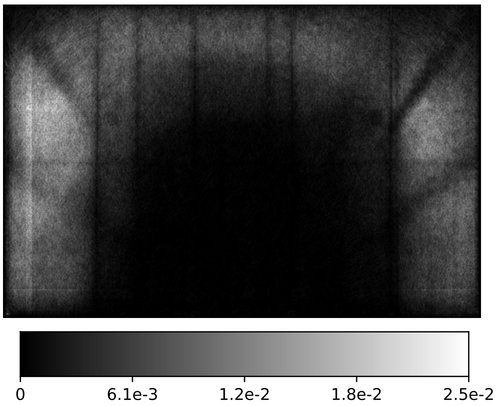

Figure 6: Heat map of droplets locations. The intensity scale indicates the probablity a pixel is marked as droplet.

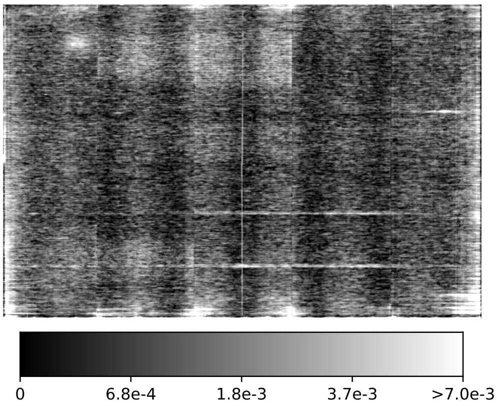

Figure 7: Heat map of shunts locations. The intensity scale indicates the probability a pixel is marked as shunt.

Table 2: Time (seconds) and memory (gigabytes) requirements for model evaluations

Model	Time	Maximum Memory
Shunts	28s	3.5GB
Droplets	50s	3.2GB

6 Implementation, timing and memory usage

Our ensemble models require computation of basis encoder-decoder networks before the ensemble model can be computed. In our current implementation due to the memory restriction the basis encoder-decoder networks are in parallel computed on a Intel Xeon CPU with 88 threads, with the ensemble model is computed on a NVIDIA Quadro M5000 graphic card. Table 2 shows average time and memory required to segment a single EL-image. The computations are performed on 100 test images.

Our implementation uses Keras, [37], library with combination of Tensorflow, [38]. We incorporate in our code publicly available implementations of several networks implemented in [39]. Furthermore, for the computation of several metrics we use the scikit-image library, [40].

The data analysis and plots have been done using R [41] and ggplot2 package [42]. The simulation script uses the GNU-parallel bash utility [43].

7 Conclusions and outlook

In this paper we proposed an ensemble model to combine several encoder-decoder networks together in order to perform image segmentation of EL images of thin-film modules. Our results indicate that the combined model performs better that the basis encoder-decoder networks.

The multichannel encoder-decoder framework is not restricted to ensemble of encoder-decoder models only, but allows to incorporate arbitrary segmentation models together. Generally speaking, the multiple-channel input encoder-decoder networks can be used also for other purposes. We conjecture that such networks can be used to improve segmentation on images where multiple EL image with different measurement conditions are used. Furthermore, different types of measurements can be used simultaneously, e.g. thermography and EL images.

Our image segmentation framework is general and can be adapted for other types of defects in thin-film modules, as well as different type of PV technology (e.g. crystalline technology).

We apply our ensemble models in order to identify droplets and shunts in a database with 6000 images of CIGS modules, all of one module type and one manufacturer. We show heat maps depicting the probability of a shunt or droplet occurring at a certain location in the solar module. The results show that the systematic segmentation of a large volume of images can reveal subtle features which cannot be inferred from studying individual images. Thus, we argue this type of segmentation models may aid process optimization and quality control by manufacturers.

Acknowledgements

This work is supported by the Solar-era.net framework in the project "PEARL TF-PV" (Förderkennzeichen: 0324193A).

References

[1] *PEARL-TF project website.* Accessed: 2019-12-01. URL: `https://pearltf.eu/`.

[2] M. Demant, P. Virtue, A. S. Kovvali, S. X. Yu, and S. Rein. "Deep learning approach to inline quality rating and mapping of multi-crystalline Si-wafers". In: *35th European Photovoltaic Solar Energy Conference and Exhibition.* 2018, pp. 814–818.

[3] S. Deitsch, C. Buerhop-Lutz, A. Maier, F. Gallwitz, and C. Riess. "Segmentation of Photovoltaic Module Cells in Electroluminescence Images". In: *arXiv preprint arXiv:1806.06530* (2018).

[4] S. Deitsch, V. Christlein, S. Berger, C. Buerhop-Lutz, A. Maier, F. Gallwitz, and C. Riess. "Automatic classification of defective photovoltaic module cells in electroluminescence images". In: *Solar Energy* 185 (2019), pp. 455–468.

[5] A. K. V. de Oliveira, M. Aghaei, and R. Rüther. "Automatic Fault Detection of Photovoltaic Array by Convolutional Neural Networks During Aerial Infrared Thermography". In: *36th European Photovoltaic Solar Energy Conference and Exhibition.* 2019.

[6] A. M. Karimi, J. S. Fada, M. A. Hossain, S. Yang, T. J. Peshek, J. L. Braid, and R. H. French. "Automated Pipeline for Photovoltaic Module Electroluminescence Image Processing and Degradation Feature Classification". In: *IEEE Journal of Photovoltaics* 9.5 (2019), pp. 1324–1335.

[7] E. Sovetkin and A. Steland. "Automatic processing and solar cell detection in photovoltaic electroluminescence images". In: *Integrated Computer-Aided Engineering* Preprint (2019), pp. 1–15.

[8] E. Sovetkin, T. Weber, E. J. Achterberg, and B. E. Pieters. "Encoder-decoder semantic segmentation models for electroluminescence images of thin-film modules". In: *IEEE Journal of Photovoltaics* (2020). submitted.

[9] D.-M. Tsai, C.-C. Chang, and S.-M. Chao. "Micro-crack inspection in heterogeneously textured solar wafers using anisotropic diffusion". In: *Image and Vision Computing* 28.3 (2010), pp. 491–501.

[10] S. A. Anwar and M. Z. Abdullah. "Micro-crack detection of multicrystalline solar cells featuring shape analysis and support vector machines". In: *IEEE International Conference on Control System, Computing and Engineering.* IEEE. 2012, pp. 143–148.

[11] D.-M. Tsai, S.-C. Wu, and W.-C. Li. "Defect detection of solar cells in electroluminescence images using Fourier image reconstruction". In: *Solar Energy Materials and Solar Cells* 99 (2012), pp. 250–262.

[12] D.-M. Tsai, S.-C. Wu, and W.-Y. Chiu. "Defect detection in solar modules using ICA basis images". In: *IEEE Transactions on Industrial Informatics* 9.1 (2012), pp. 122–131.

[13] S. A. Anwar and M. Z. Abdullah. "Micro-crack detection of multicrystalline solar cells featuring an improved anisotropic diffusion filter and image segmentation technique". In: *EURASIP Journal on Image and Video Processing* 2014.1 (2014), p. 15.

[14] F. Waldner and F. Diakogiannis. "Deep learning on edge: extracting field boundaries from satellite images with a convolutional neural network". In: *arXiv preprint arXiv:1910.12023* (2019).

[15] V. Iglovikov and A. Shvets. "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation". In: *arXiv preprint arXiv:1801.05746* (2018).

[16] *Inria Aerial Image Labeling Dataset.* Accessed: 2019-12-17. URL: `project.inria.fr/aerialimagelabeling`.

[17] M. Havaei, F. Dutil, C. Pal, H. Larochelle, and P.-M. Jodoin. "A convolutional neural network approach to brain tumor segmentation". In: *BrainLes.* Springer. 2015, pp. 195–208.

[18] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639 (2017), p. 115.

[19] M. Attia, M. Hossny, S. Nahavandi, and H. Asadi. "Surgical tool segmentation using a hybrid deep cnn-rnn auto encoder-decoder". In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC).* IEEE. 2017, pp. 3373–3378.

[20] B. Kayalibay, G. Jensen, and P. van der Smagt. "CNN-based segmentation of medical imaging data". In: *arXiv preprint arXiv:1701.03056* (2017).

[21] The GIMP Development Team. *GIMP.* Version 2.10.12. June 12, 2019.

[22] PI-Berlin. *ThinFiA - Software Version provided and made by PI-Berlin.* 2018.

[23] B. Misic, B. E. Pieters, U. Schweitzer, A. Gerber, and U. Rau. "Defect Diagnostics of Scribing Failures and Cu-Rich Debris in Cu(In,Ga)Se_2 Thin-Film Solar Modules With Electroluminescence and Thermography". In: *IEEE Journal of Photovoltaics* 5.4 (July 2015), pp. 1179–1187.

[24] B. Misic. "Analysis and Simulation of Macroscopic Defects in Cu(In,Ga)Se2 Photovoltaic thin film modules". In: vol. 372. Schriften des Forschungszentrums Jülich. Reihe Energie und Umwelt / Energy and Environment. Forschungszentrum Jülich GmbH, 2017, pp. 17–36.

[25] T. Tran, B. Pieters, M Siegloch, A Gerber, C Ulbrich, T Kirchartz, R Schäffler, and U Rau. "Characterization of shunts in Cu (In, Ga) Se2 solar modules via combined electroluminescence and dark lock-in thermography analysis". In: 2011, pp. 2981–2985.

[26] U. Rau. "Reciprocity relation between photovoltaic quantum efficiency and electroluminescent emission of solar cells". In: *Phys. Rev. B* 76 (8 Aug. 2007), p. 085303.

[27] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning.* MIT press, 2016.

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[29] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2016, pp. 770–778.

[30] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[31] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer. 2015, pp. 234–241.

[32] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2015, pp. 3431–3440.

[33] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. "Pyramid scene parsing network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2017, pp. 2881–2890.

[34] V. Badrinarayanan, A. Kendall, and R. Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495.

[35] P. Getreuer. "Chan-vese segmentation". In: *Image Processing On Line* 2 (2012), pp. 214–224.

[36] P. F. Felzenszwalb and D. P. Huttenlocher. "Efficient graph-based image segmentation". In: *International journal of computer vision* 59.2 (2004), pp. 167–181.

[37] F. Chollet et al. *Keras.* `https://keras.io`. 2015.

[38] 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, and X. Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org. 2015.

[39] *Image Segmentation Keras : Implementation of Segnet, FCN, UNet, PSPNet and other models in Keras.* Accessed: 2019-12-01. URL: `https://github.com/divamgupta/image-segmentation-keras`.

[40] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. "scikit-image: image processing in Python". In: *PeerJ* 2 (June 2014), e453.

[41] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria, 2019.

[42] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2016.

[43] O. Tange. *GNU Parallel 2018.* Tange, Ole, Mar. 2018.