

Cloud Segmentation and Matching Using Deep Learning in All-Sky Images

Niklas Theis¹, Grit Behrens¹ , Andreas Boschert^{2,3} , and Mike Zehner² 

¹ University of Applied Sciences Bielefeld, Solar Computing Lab, Germany

² Technical University of Applied Sciences Rosenheim, Rosenheim Technology Centre for Energy and Buildings, Laboratory for Solar Energy Systems, Germany

³ Technical University of Munich, Munich Institute of Integrated Materials, Energy and Process Engineering (MEP), Combined Smart Energy Systems (CoSES), Germany

Abstract. In this paper, we focus on the segmentation of clouds in All Sky Images using a U-Net-based Deep Learning model and the subsequent recognition of the same cloud in different images. This research lays the foundation for the development of solar radiation forecasts with All-Sky Imagers. The implemented model initially extracts relevant features from the input image using convolutions, thereby reducing the resolution. In the subsequent step, the resolution is restored to its original level using transposed convolutions. Contours are then created from all segmented clouds. Using these contours as references, the same cloud is identified in images from different All-Sky Imagers through template and contour matching. We demonstrate that this segmentation approach yields good results on a small test dataset. Additionally, the recognition of clouds in images from different cameras show promising results, with 75 % of clouds being correctly matched.

Keywords: All-Sky Imagers, Deep Learning, U-Net, Cloud Segmentation

1. Introduction

Irradiance Volatility (IV) and Irradiance Enhancement (IE) are primarily influenced by clouds and their characteristics [1], [2]. The volatility has a major impact on the power output of photovoltaic systems (PV) and is therefore crucial for energy production. To accurately model the output of PV systems, a temporally and spatially high-resolution prediction of solar radiation is required. All-Sky Imagers (ASI) play an important role in this process. By following a sequence of procedures, it is possible to forecast the solar radiation and thus the yields of PV systems using a network of ASI. A critical step in this process is recognizing cloud formations in images taken by two All-Sky Imagers to determine the future positions and heights of the clouds. This information is essential for the forecasting model, which is why this paper focuses on cloud segmentation and matching in All-Sky Images.

2. Initial Situation

The data used consists of recordings from two ASIs (ASI-16/51 Advanced, distributed by EKO Instruments Europe B.V.), each with a field of view of 180°. Both All-Sky Imagers are mounted vertically, looking into the sky, oriented south, and are separated by 942 meters. The ASIs are part of a test setup for researching solar radiation at the Technical University of Applied Sciences Rosenheim and are installed on the main campus and at the Rosenheim Technology Centre for Energy and Buildings. Every 20 seconds, new images are captured. A sample of

these images from both All-Sky Imagers is displayed in Fig. 1. To simplify naming, the cameras will be referred to as Camera A and Camera B, as indicated in the captions of Fig. 1.

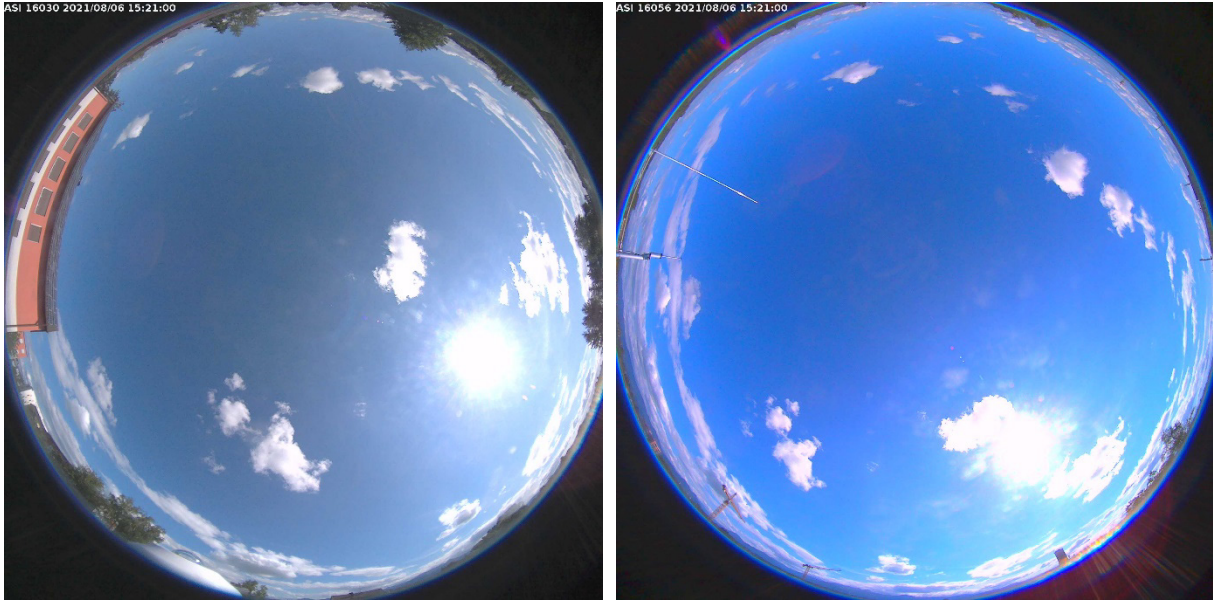


Figure 1. Original All-Sky Images captured on 06-08-2021 at 14:21:00 (UTC). The left image is from Camera A, and the right image is from Camera B.

For the training of the deep learning model the fully labeled dataset WSISEG was used [3]. The input images were captured with similar, but not identical ASIs. The three labels in the dataset are the sky, the clouds, and all other objects, such as trees, houses, and the sun.

3. Methods and Technologies

3.1 Image Preprocessing



Figure 2. Preprocessed images: The left image displays the distortion-corrected Image A, and the right image shows the rotation- and distortion-corrected Image B.

The preprocessing of the images to prepare them for the deep learning model consists of multiple steps. The images of Camera B are rotated 36° counterclockwise compared to the

images from Camera A, to orientate the images towards the south. This rotation is corrected as an initial processing step. To be able to match clouds between both cameras the distortion must be corrected. Otherwise, the shape of a cloud would be different depending on the position within an image. The distortion correction is done with a polynomial model for correcting radial distortion [4], [5], [6]. The first correction was done with manually determined parameters. This method was used for the images presented above. For a second correction, pictures containing calibration pattern placed around the camera were used to create a camera specific calibration. With this, a highly effective correction was achieved, with the tradeoff of reducing the field of view (FOV). Therefore, in this paper, the manual calibration with a larger field of view is used. The corrected images from those in Fig. 1 are displayed in Fig. 2. To maintain a manageable training and inference time for the Deep Learning model, the images are scaled down to a size of 224x224. Additionally, the pixel values are normalized to a range of 0 to 1.

3.2 Cloud Segmentation

The segmentation of the clouds from the sky and other objects, such as trees and buildings at the edges of the All-Sky Image, is achieved with a deep learning model based on the U-Net architecture, as described by Ronneberger et al. [7].

Such a model consists of a down-sampling stack and an up-sampling stack. In each step of the down-sampling stack, the resolution is halved, and the number of feature maps is doubled. In the model used, each step comprises two convolution layers to extract the features, a max-pooling layer to reduce resolution, and a dropout layer to prevent overfitting. The up-sampling stack mirrors the structure of the down-sampling stack but with transposed convolution instead of max-pooling to double the resolution of the input. Before passing the feature maps to the dropout layer, the feature maps from corresponding resolution in the down-sampling stack are appended to them. Two convolution and one dropout layer are used to connect the down- and up-sampling stacks.

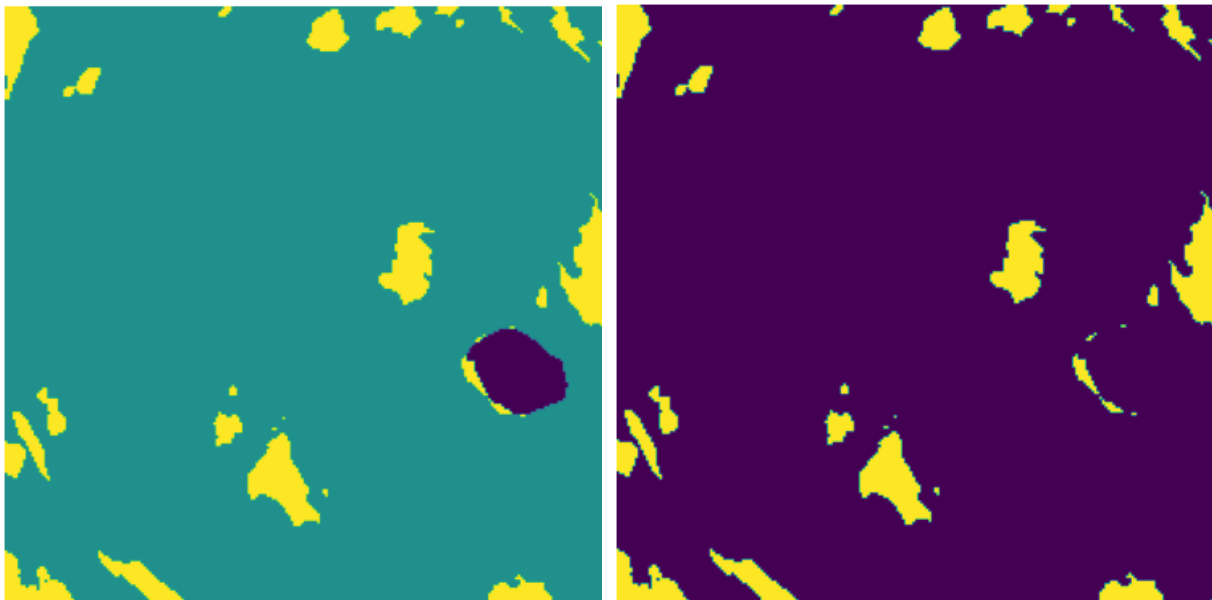


Figure 3. Image with the steps of the cloud segmentation. The left image shows the prediction by the deep learning model for the distortion-corrected image A from Figure 2. The right image shows only the clouds from the prediction.

The model was trained on the WSISEG dataset for a total of 40 epochs. A sample prediction for the distortion corrected image from Fig. 2 is displayed in Fig. 3. The label 0 (purple) denotes the sun and all objects like trees and buildings. The labels 1 (green) and 2 (yellow) stand for the sky and the clouds, respectively.

3.3 Cloud Detection

To isolate a specific cloud within an image, the number of labels is first reduced from three to two combining labels 0 and 1. This simplification is feasible since only the clouds are of importance, and distinguishing between the sun and the sky is not relevant. The result for the left image from Fig. 3 is shown on the right in Fig. 3. This process creates a binary image, which is then used to retrieve the contours of all clouds using an algorithm by Suzuki et al. [8], as implemented in OpenCV. To remove small errors from the segmentation, an area threshold of 80 pixels is used to discard clouds that are too small.

3.4 Cloud Recognition

As a first solution for recognizing a cloud in the second image, template matching is used. Based on the bounding boxes of the found contours, a rectangular template is cut out of the input image. To ensure that the entire cloud is included in the template, an additional margin of 5 pixels is added around the boundaries of the contour. For the image from Fig. 2, this is shown in Fig. 4. To make the template matching more reliable, parts of the input image are removed if it can be guaranteed that the cloud is not in that part. For example, for cloud number 5 from Fig. 4 it is guaranteed that the cloud is also in the right half of the image from Camera B. Therefore, only the right half is used for template matching. With this logic, an attempt is made to find the relevant half of the image for each cloud. If the cloud is in the center, the whole image is used. The result of the template matching is shown in Fig. 4 on the right.

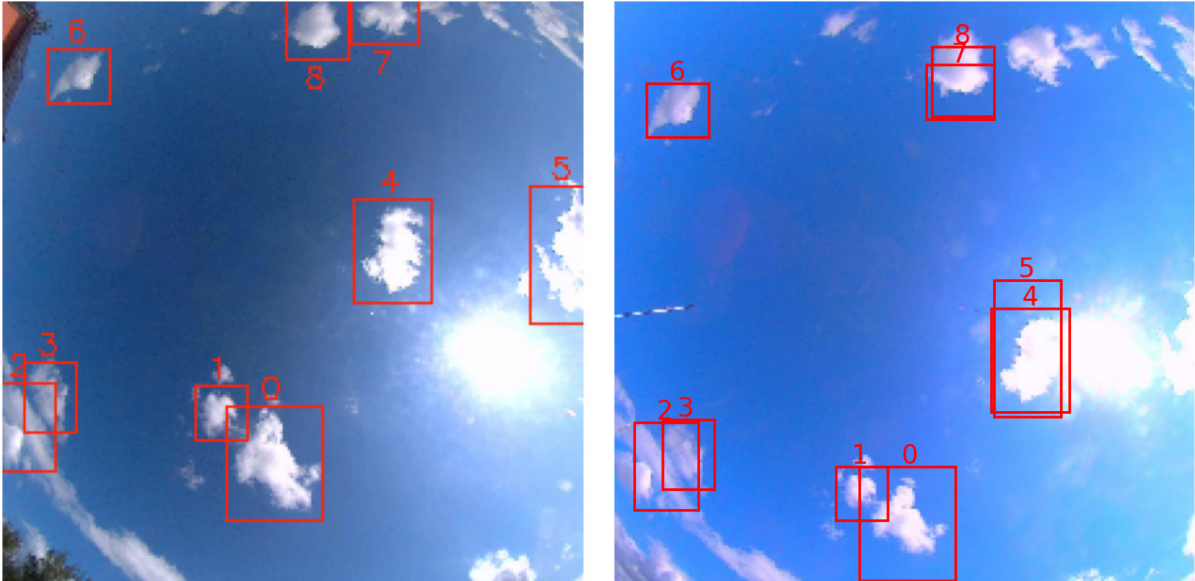


Figure 4. Examples of the template matching. The left image shows the bounding boxes used to cut out the templates from image A, and the right image shows the resulting bounding boxes after the template matching on image B.

The second solution to achieve cloud recognition is based on the contours. Instead of running the cloud segmentation on only one image, both images are segmented. Binarization is done for both, and the contours are determined.

The extracted contours are then matched with an algorithm based on Hu-Moments [9] as implemented in OpenCV. The matching method titled I_3 was used, which is based on Equation (1). Where m_i are the Hu-Moments for the shapes A and B, respectively.

$$I_3(A, B) = \max_{i=1 \dots 7} \frac{|m_i^A - m_i^B|}{|m_i^A|} \quad (1)$$

The resulting scores for the contours from Fig. 5 are shown in Table 1.



Figure 5. Examples of contour matching. The left image shows the contours for image A, and the right image shows the contours for image B.

Table 1. Similarity of the contours based on Hu-Moments; lower is better. The columns represent the contours from image A, and the rows represent the contours from image B. The best match for each contour is highlighted in bold text. The correct combinations are: B0-A1, B1-A6, B2-A7, and B3-A8

	A0	A1	A2	A3	A4	A5	A6	A7	A8
B0	1.45	0.25	0.31	0.26	0.99	0.29	0.62	0.27	0.30
B1	0.88	0.48	0.47	0.77	1.11	0.20	0.55	0.16	0.37
B2	0.91	0.28	0.46	0.53	0.91	0.07	0.41	0.15	0.37
B3	1.57	0.31	0.28	0.36	1.09	0.36	0.70	0.14	0.18

To improve the matching further, two more metrics are included. For the first one, the area within each contour is determined in pixels. For each possible match, the absolute difference between the contour areas is calculated. The second metric consists of the distance between the centers of two matched contours. To determine the center, the bounding box for a given contour is used. By multiplying these metrics with the shape matching scores, the values from Table 2 are received. The values can also be interpreted as a confidence score, where a lower value is better.

Table 2. Scores for matching the shapes, including the distance between centers and area of the contours. Lower is better. Highlighting is done as described in the caption of Table 1. The correct combinations are: B0-A1, B1-A6, B2-A7, and B3-A8

	A0	A1	A2	A3	A4	A5	A6	A7	A8
B0	35	61	92	36	642	208	720	177	109
B1	991	1292	56	529	96	367	4	60	171
B2	633	646	174	239	449	56	144	1	26
B3	30	521	214	62	1055	179	431	9	6

4 Results

Currently, the evaluation of the results has been done preliminarily due to the absence of dedicated datasets. The evaluation was performed visually based on a small test dataset. These results indicate that the cloud segmentation using a deep-learning-based approach is effective.

For matching clouds in two images, template matching shows good performance for most clouds. However, the matching sometimes fails, particularly at the edge of an image and around the sun. The approach based on contour shapes, area differences, and distances performs well for contours found by segmentation. Some clouds at the image edges and around the sun are not detected in both images and therefore cannot be matched.

5 Conclusion and Outlook

The algorithm created based on a deep learning model achieves promising results, suggesting that segmentation and recognition are possible with the approach used.

The preprocessing can be further improved by determining the position of the sun within the images and either mask it before inputting the image into the model or providing the model with information about the sun in another way.

For the algorithm based on contours better results might be achieved by improving the initial cloud segmentation. This could be accomplished by adjusting the model architecture and optimizing training. To achieve better accuracy, a labeled dataset for the cameras used should be created, which preferably not only distinguishes the clouds from the sky but also differentiates individual clouds from each other. Currently, using the distance between contour centers does not account for the expected shift due to the kilometer distance between the cameras. Including this consideration could further optimize the algorithm.

Author contributions

Niklas Theis performed writing – original draft, conceptualization, and software development. Grit Behrens performed conceptualization, writing – review & editing, and supervision. Andreas Boschert performed conceptualization, data curation, writing – review & editing. Mike Zehner performed conceptualization.

Competing interests

The authors declare that they have no competing interests.

References

- [1] M. Zehner, T. Weigl, M. Hartmann, S. Thaler, O. Schrank, B. Mayer, T. Betts, R. Gottschalg, K. Behrens, G. König-Langlo, B. Giesler, G. Becker und O. Mayer, „Energy Loss Due to Irradiance Enhancement“, 2011.
- [2] N. Stut, A. Boschert, F. Kaiser, M. Zehner, B. Mayer und O. Mayer, „Analyse von Einstrahlungsvolatilität und -überhöhungen in hochaufgelösten Datensätzen des DWD und MIM zur Untersuchung von Korrelationen zu meteorologischen Messdaten“.
- [3] W. Xie, D. Liu, M. Yang, S. Chen, B. Wang, Z. Wang, Y. Xia, Y. Liu, Y. Wang und C. Zhang, „SegCloud: a novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation,“ *Atmospheric Measurement Techniques*, Bd. 13, p. 1953–1961, 2020.
- [4] D. Scaramuzza, A. Martinelli und R. Siegwart, „A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion,“ in *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, New York, NY, USA, IEEE, 2006, pp. 45.
- [5] D. Scaramuzza, A. Martinelli und R. Siegwart, „A Toolbox for Easily Calibrating Omnidirectional Cameras,“ in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, IEEE, 2006, pp. 5695-5701.
- [6] M. Rufli, D. Scaramuzza und R. Siegwart, „Automatic detection of checkerboards on blurred and distorted images,“ *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3121-3126, 2008.
- [7] O. Ronneberger, P. Fischer und T. Brox, „U-Net: Convolutional Networks for Biomedical Image Segmentation,“ *CoRR*, Bd. abs/1505.04597, 2015.
- [8] S. Suzuki und K. be, „Topological structural analysis of digitized binary images by border following,“ *Computer Vision, Graphics, and Image Processing*, Bd. 30, Nr. 1, pp. 32-46, 1985.
- [9] M.-K. Hu, „Visual pattern recognition by moment invariants,“ *IRE Transactions on Information Theory*, Bd. 8, Nr. 2, pp. 179-187, 1962.