PROCEEDINGS OF ECOS 2020 - THE 33RD INTERNATIONAL CONFERENCE ON
EFFICIENCY, COST, OPTIMIZATION, SIMULATION AND ENVIRONMENTAL IMPACT OF ENERGY SYSTEMS
JUNE 29-JULY 3, 2020, OSAKA, JAPAN

A supervised deep learning model for cloud/sky image segmentation

Youssef Karout [a,b], *Stéphane Thil* [a,b], *Julien Eynard* [a,b] *and Stéphane Grieu* [a,b]

[a] *PROMES-CNRS, Rambla de la thermodynamique, Tecnosud, 66100 Perpignan, France*
[b] *University of Perpignan Via Domitia, 52 Avenue Paul Alduy, 66860 Perpignan, France*
youssef.karout@promes.cnrs.fr, stephane.thil@promes.cnrs.fr, julien.eynard@promes.cnrs.fr, stephane.grieu@promes.cnrs.fr

Abstract:
Sky images captured by ground-based cameras are increasingly used nowadays because of their applications in a number of fields, in particular assessment and forecasting of the solar resource. Direct normal irradiance (DNI) forecasts, which can be used to improve the efficiency of concentrated solar power systems, are highly correlated with cloud cover evolution. In this context, accuracy is a key feature; taking into consideration the nature of the clouds and changes in the atmospheric conditions, achieving such high accuracy is problematic and challenging. In this paper, we present two segmentation tools to address the complicated nature of the clouds, attempting to present models free from tunable thresholds and completely based on supervised learning. The first one is a multilayer (deep) artificial neural network-based model and the second one is a k-nearest-neighbors-based model. Using high-dynamic-range (HDR) sky images collected in Perpignan (southern France), we compare these two models with state-of-the-art segmentation algorithms.

Keywords:
Sky-imaging data, Direct normal irradiance, Image segmentation, Machine and deep learning, Artificial neural networks, k-nearest neighbors.

1. Introduction

Solar energy is a high-trending field of research nowadays and research centers are more and more interested in developing algorithms to increase as much as possible the efficiency of solar energy processes and systems. As a result, short-term forecasting of the solar resource is critical, especially for the development of efficient model-based predictive controllers. We are interested in the intrahour forecasting of DNI using HDR sky images given by a ground-based camera [1], which provides high spatial-temporal resolution unlike satellite-based imaging [2]. Since this task is deeply influenced by both cloud cover and cloud motion [3], a segmentation method which enables to identify the cloud pixels with high accuracy is needed. That is why a machine learning-based algorithm has been developed, which permits also to solve the problems posed by the threshold-based algorithms that will be discussed in the coming sections of the paper. Our goal is to perform probabilistic segmentation with no manual interference.

1.1. Related work

- *Threshold-based algorithms.* The cloud luminosity, thickness, position in the sky, and clear-sky radiance anisotropy effect complicate the cloud detection task. As a result, algorithms based on only one fixed threshold are rejected and an adaptive threshold featuring the red to blue channel [4], which uses the hybrid thresholding algorithm (HYTA), is introduced. Note that the possibility of enhancing the minimum cross entropy (MCE) by saturating the threshold and then defining different regions of interest in the image, which permits to calculate a local threshold in each region, is highlighted in [5] and [6]. As in [7], the difference between the blue channel and the red one proved to be a good candidate to perform the segmentation. More sophisticated methods were introduced in [8], where a fuzzy c-mean clustering is performed on the normalized difference between the red channel and the blue one to gather some seeds, and then the graph cut method [9–11] is performed to classify the pixels. Threshold calculation based on the Otsu's method [12], which treats the gray scale feature of an image, is also used to segment the cloud/sky pixels. In [13], a method, where the red-to-blue ratio histogram is used and then the average of the two highest classes is averaged with the mean of the red/blue values holding the highest local standard deviation of the gray scale image, is presented.

- *Machine learning-based algorithms.* A support vector machine (SVM) classifier is used in [3] to perform a supervised learning based on the RGB (red, green and blue) channels, both the local standard deviation and Laplacian of Gaussian being used as predictors (features) to construct a binary mask, achieving notable accuracy. The images used during the training phase are manually labeled. An artificial neural network (ANN) enables detecting the clouds. In addition, the normalized red-to-blue channel, normalized saturation-value ratio (NSV), red-to-blue ratio, gray-scale values and RGB channels can serve as good features for the network used [14]. Another approach including the following features was introduced in [15]: hue, saturation and brightness (HSV), RGB channels, R/B, red-blue difference, the pixel's estimated movement from the previous image, the pixel's distance from the Sun and the Sun's azimuth and zenith angles. The artificial neural network used, which consists of two hidden layers of ten neurons each, also achieved notable accuracy. Furthermore, in [16] the cloud identification network uses the variance of the red channel, the mean of that channel and the mean of the blue channel on a given neighbor. The network is designed with one hidden layer with the same number of neurons as the input layer. The approach presented in [17] provides a two-step identification method. First, the image is grouped to one of three clusters based on some coarse grain characteristic, and then a specific network is used to identify the pixels.

1.2. Proposed method

The above methods perform accurately in simple cases where clouds intensities are uniform and cloud color is homogeneous; but in the case where different types of clouds (cirrostratus, cirrocumulus, cirrus, altostratus, etc.) are to be identified, the thresholding-based methods pose the problem of over- or under-identification depending on the aggressivity of the threshold and the tolerance on the details. In these cases, these methods lose accuracy on the boundaries of the image (note that we are applying these methods on a camera with a fisheye lens). The machine learning-based models can easily overcome this problem in the learning process. In addition, the Sun affects these methods drastically due to the glare. Our goal is to overcome the Sun glare problem, preserve as much information as possible and present a model with no predefined parameters. Furthermore, we are interested in constructing a probabilistic cloud mask describing the membership of each pixel, which is not possible with thresholding methods. This probabilistic map is of great importance when it comes to forecasting DNI as it provides better understanding of the clouds and their types.
To achieve the required performance we developed two models issued from the supervised machine learning approach. The first model is an artificial neural network with multiple layers (deep architecture to be explained in the coming sections) whereas the second model is a k-nearest neighbors model (k-NN) trained with different features. The proposed models are compared with some of the methods mentioned above, the evaluation is done both visually and numerically. Then, key observations are provided to evaluate the performance and prove the accuracy of our models compared to threshold-based methods.

1.3. Database

In this paper, we use sky images [1] collected at PROMES-CNRS, in Perpignan (southern France). The weather is moderate and the cloud cover varies continuously, providing different sky situations

such as completely/partially covered sky (with different kinds of clouds) and clear sky.
The ground-based camera used is a new generation sky imager developed by PROMES-CNRS, with the help of PROMECA (`https://www.promecaweb.fr/`). It is equipped with a CMOS sensor, a fisheye lens (185°@ 0.15°), a neutral density (ND) filter and an embedded computer. The operating system is Debian GNU/Linux. Temperature control is performed through active cooling and heating. Each HDR image is generated on the fly using a batch of 30 images taken with different exposure times within a second: cloud motion is negligible. HDR imaging allows details in both the circumsolar area and the darkest parts of the sky to be preserved. A fully automated geometrical calibration is performed by comparing the Sun's position in the sky and the one detected on the image.
The database used for this study is composed of 5959 HDR images over four consecutive days, along with the Sun's position. The size of those 64-bit images is 1300×1216 pixels. The images used for both the training and validation phases present around 215 different features dealing with sky situations (clear sky, overcast, multiple cloud types, etc.). Note that for visualizing goals we perform tone-mapping on the images to be able to show them as in Fig. 1.

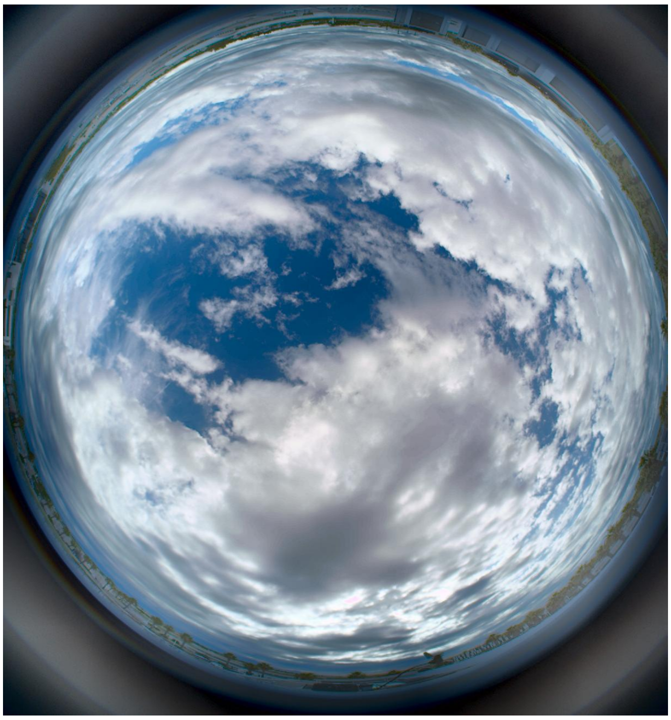


Figure 1: Tone-mapped image from the PROMES-CNRS database (Perpignan, southern France).

2. Proposed models

In this section, we present the two proposed models, emphasizing their implementation techniques and tools.

2.1. Multilayer artificial neural networks

Artificial neural networks are very powerful artificial intelligence tools. However, to achieve high accuracy and low data loss, we have to pick wisely the features (predictors), in order to learn to classify.

2.1.1. Tools and features

An artificial neural network, as any machine learning model, requires specific features that will help match the input to the output (observation). As seen in the related work, the most used features are the following:

- The RGB (red, green and blue) channels.

- The red-to-blue ratio (RBR) channel, defined as follows:

$$RBR = \frac{R}{B} \tag{1}$$

- The normalized red/blue ratio (NRBR) channel, defined as follows:

$$NRBR = \frac{R - B}{R + B} \tag{2}$$

- The local standard deviation on a specified neighborhood.
- The Laplacian of Gaussian (used for edge detection).
- The gray scale.

An ANN-based model can be expressed as the following equation:

$$Y = F(X) \tag{3}$$

where X is a vector of pixel features and Y is the pixel class (label) decision, where F can be seen as a black box holding the weights and biases of each unit in the network. The mapping is performed by applying these trained parameters on the normalized input vector X.
More specifically, such a model can be defined as follows:

$$y(x, w) = \sum_{j=1}^{m} w_j f\left(\sum_{i=1}^{d} w_{ji} x^{\star} + b_1\right) + b_2 \tag{4}$$

where $w = (w_j, w_{ji}, b_1, b_2)$, which is estimated during the training phase, contains the set of weights w_j, w_{ji} and two bias b_1, b_2, and f is the activation function. Artificial neural networks use a variety of learning techniques and optimizers, but the most popular is the backpropagation algorithm [18].

2.1.2. Network layers and units

The network has the following types of layers:

- Up-sampling layer, where the number of units is doubled compared to the previous layer in the network.
- Down-sampling layer, where the number of units is divided by two compared to the previous layer in the network.
- Bottleneck layer, where the number of units is preserved compared to the previous layer in the network.
- Dropout layer (dropout is used to avoid overfitting by randomly dropping units from the network during the training phase).

Varying the number of layers will affect both the accuracy and execution time accordingly; for example, increasing the number of layers will increase the execution time where, with sufficient data size, it might fight against overfitting. As for activation functions, we chose rectified linear units (ReLU) for the intermediate layers and sigmoid functions for the output layer, respectively (see Fig. 2).

2.1.3. Generation of training data

First of all, as proposed in [3], cloud labeling has been manually done, carefully, using the *Image Segmenter Matlab Toolbox* on four full images of different natures (clear sky, overcast and heterogeneous situation). Eliminating the physical mask and potential NaN values from the data, we are left with 4 425 990 observations (vector of features). This permits us to build a first NN, which is used to generate 11 cloud masks of another 11 images; these 11 masks are then refined manually. In the end, a training database of 15 images is obtained (4 clear sky images, 5 overcast images, 6 mixed images), representing 12 193 471 observations.

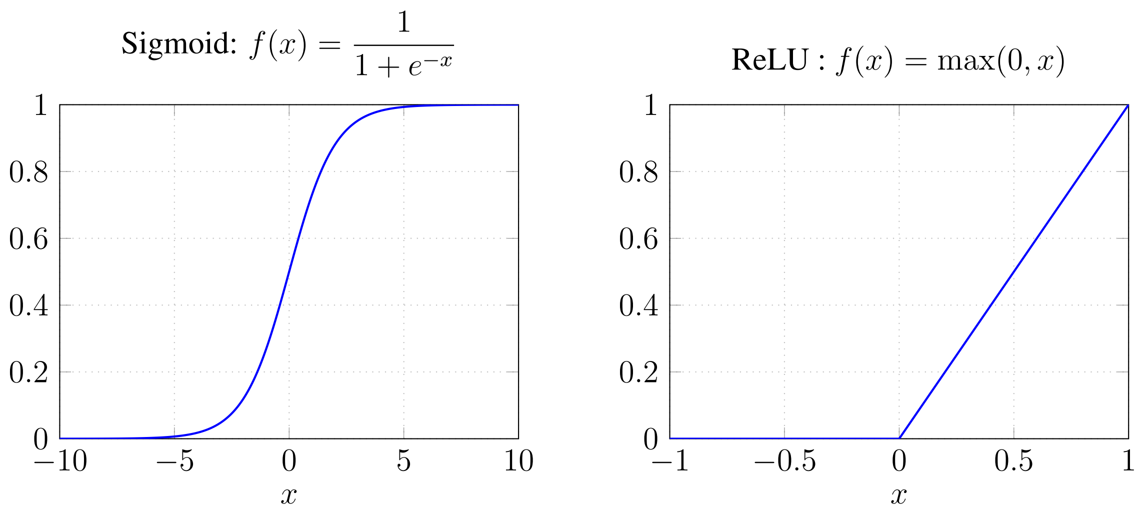


Figure 2: Activation functions.

2.2. k-NN-based model

The second model proposed in the present paper is described in this section.

2.2.1. Definition

The k-nearest neighbors algorithm is a simple algorithm that stores the dataset and classifies new cases based on a similarity measure. It provides the possibility of finding the k closest points in the data set to a set of points.

2.2.2. Tools and features

Several k-NN variants are possible, each one is characterized by specific parameters such as:

- Distance metric [19], such as city block, chebychev distance or Euclidean distance.
- Number of nearest neighbors, which ranges from 1 to 100.
- Nearest neighbor search method [20], such as Kd-tree searching method.

Each combination of these parameters affects the performance, which varies their prediction speed, learning speed, flexibility, interpretability, implementation behaviour...

3. Implementation and comparative study

In this section we will be mentioning specific and technical facts regarding the proposed models. Then, we are going to compare them with several state-of-the-art methods and algorithms; allowing us to show their robustness and accuracy versus threshold-based algorithms.

3.1. Implementationl

3.1.1. ANN-based model

After performing several tests, we ended up with a neural network having 97$\%$ validation accuracy and 11$\%$ of validation loss (calculated using the binary cross-entropy method). This network uses the RGB channels, red to blue ratio, the local standard deviation (11×11 neighborhood) and the normalized difference of red and blue channels. For such predictors we construct a deep artificial neural network having the following layers (for a total of 19 layers):

- 4 up-sampling layers.
- 1 bottleneck layer.
- 4 down-sampling layers.
- 1 dropout layer (40$\%$) after each of the above layers.

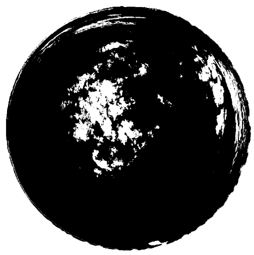
- 1 output layer.

Note that the choice of the features used and the mentioned architecture has been derived by trial and error. The architecture is inspired from the U-net segmentation network [21].
We trained the neural network (three epochs) using the TensorFlow/Keras GPU aided version [22]. Each epoch lasted 30 min to 40 min using ADAM (a well known adaptive learning rate optimizer) on a machine having Intel I7-4510U CPU @ 2.00-2.60 GHz as the CPU and Nvidia Geforce 840M as the GPU. Once trained, the model provides the probabilistic segmentation map of a sky image (1300×1216 pixels) in 40 s to 50 s.
A fixed threshold is applied on the neighborhood of the Sun to overcome the "Sun glare" problem [23]. Tests showed that 0.83 is a good value. This approach is impossible to apply on a binary mask, which is another advantage of having a probabilistic map.

3.1.2. k-NN-based model

For this model, we use the RGB channels, local standard deviation on a specified neighborhood (11×11 neighbors) and NRBR channel. We used the *Matlab Classification Learner* to perform the learning process, which took 40 minutes achieving an accuracy of 97 %. The distance metric is the Euclidean distance, the number of neighbors is set to 100, and the model uses the Kd-tree searching method.

(a) Validation image: case 1



(b) Ground truth: case 1

(c) Validation image: case 2

(d) Ground truth: case 2

Figure 3: Validation images.

3.2. Comparative study

In this section, we present the methods and algorithms commonly used to perform cloud/sky segmentation and then we apply them to our database.

3.2.1. State-of-the-art methods and algorithms

As explained in Subsection 1.1., there are many existing methods, mainly based on thresholding developed to perform cloud/sky segmentation. The most common methods are described below:

- The updated MCE algorithm, which is a part of the hybrid thresholding algorithm (HYTA), uses an adaptive threshold on the red to blue ratio and is one of the most used threshold-based algorithm [4]. In this paper, the enhanced version of this algorithm [6] is used.

- The graph cut method can be considered as a more advanced method used to segment HDR images where seeding is done with the help of c-fuzzy clustering. The feature used is the normalized difference of the red and blue channels.
- The Otsu's method, a well-known threshold calculation method using the gray-scale feature.
- The method proposed by Huo and Lu [13], in which the threshold is calculated by averaging the highest two classes in the red to blue ratio histogram, and then averaging this value with the average of the 20 red to blue ratio holding the highest standard deviation values. This method is proposed for heterogeneous cases only.

These methods are to be compared with our ANN-based and k-NN-based models. The obtained results are discussed in the following section.

3.2.2. Assessment

In this section, we compare our models with some of the state-of-the-art methods and algorithms. Performance of the models is evaluated by calculating the root mean square error (RMSE). Note that the ground truth masks are manually labeled. Two cases are presented (see Fig. 3): the first case is a whole image with apparent Sun, clear sky and some shattered clouds; the aim is to test the robustness of the algorithms to the Sun glare. The second case is a local part of a cloudy image, used to evaluate the performance of the proposed approaches in regions where threshold-based algorithms perform very well. Results are presented in Table 1. For better understanding of the models' performance, we present the average execution time of the models used in the comparison (see Table 2).

Table 1: RMSE obtained on the two validation images.

Approach	Otsu's method	MCE	Huo and Lu	k-NN-based model	ANN-based model
Case 1	0.55	0.47	0.26	0.21	0.15
Case 2	0.13	0.11	0.27	0.13	0.10

Table 2: Average execution time on 20 sky images.

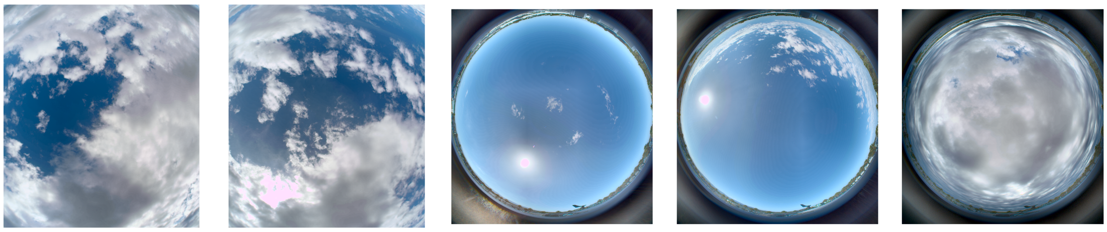
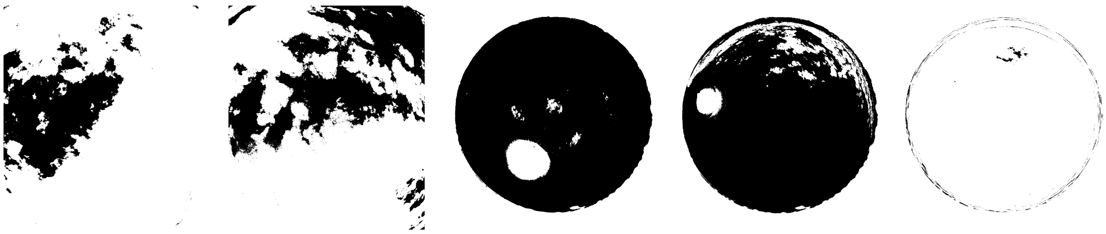
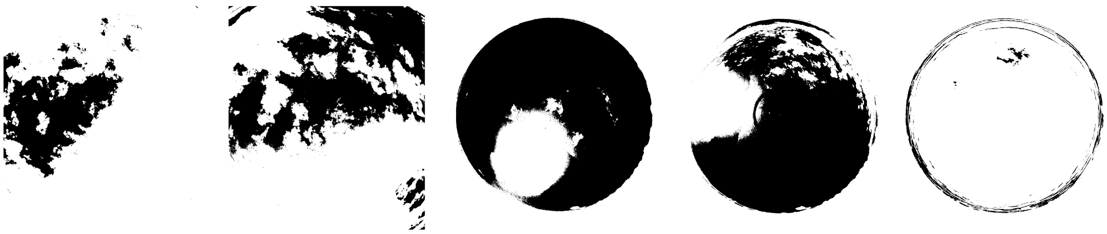
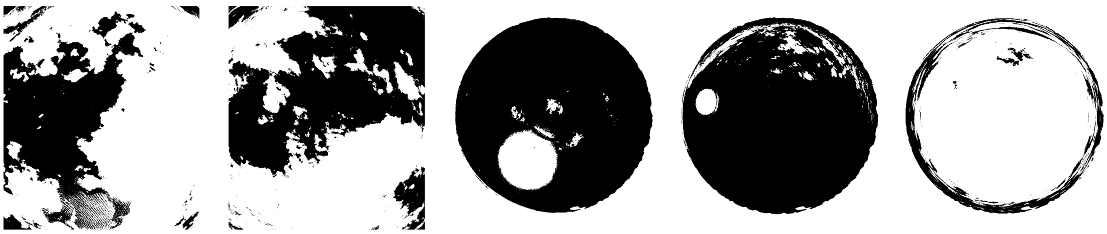
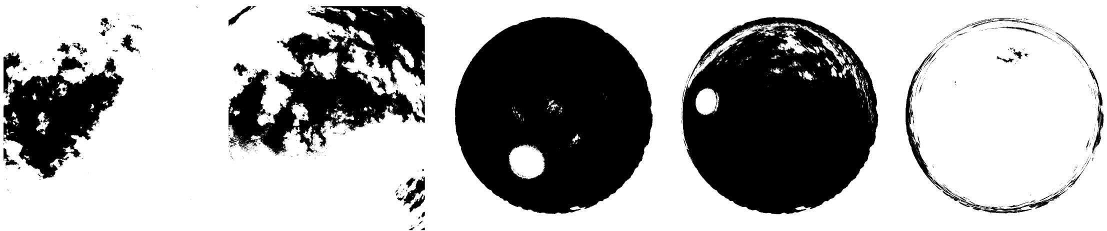
Approach	Otsu's method	MCE	Huo and Lu	k-NN-based model	ANN-based model
Time (Sec)	0.22	0.55	0.74	0.43	52.63

4. Observation and discussion

Five sky images selected from the database are shown in Table 3: two heterogeneous and regional images, two whole and sunny images and a fully clouded image. These images have been selected to show that the proposed approaches can overcome the problems generally faced by cloud detection algorithms such as the "Sun glare" problem (columns 3 and 4), thin/thick clouds (columns 1, 2 and 5), and clear-sky radiance anisotropy (columns 3 and 4). In addition, we intend to prove that our approaches are not regional, contrary to most threshold-based methods. Indeed, for these methods, it is advised to perform local thresholding to overcome the problem posed by the fish-eye lens distortion, especially on images' boundaries, where false detection is common in threshold-based algorithms. In machine learning-based models, this problem is not usually encountered and there is no need to separate images into regions of interest.

As can be seen in Table 3, the state-of-the-art methods and algorithms suffer from over- or under-identification due to the way the threshold is calculated (either soft or hard), while for machine

Table 3: Comparison between the tested approaches on five images selected in the database.

Original image	
ANN mask	
k-NN mask	
Updated MCE mask	
Huo and Lu mask	
Otsu mask	
Graph cut mask	

learning-based models the probabilistic nature of the approach is sufficient to confront the problem as shown in the first two columns.

In addition the threshold-based algorithms need some additional support and special treatment in case of dark clouds; for example in the fixed threshold approach the threshold might be too high for some situations; which implies that additional post-processing has to be proposed to solve this problem, as in e.g. [24]. Meanwhile, our model learned to distinguish this type of clouds without additional sophisticated post-processing, which serves our purpose of finding a robust algorithm.

Another observation to be mentioned is that the learning-based models are sensitive to details but robust to the Sun glare (column 3 and 4); the updated MCE algorithm is sensitive to small clouds/details but it will drastically fail on the neighborhood of the Sun, since the threshold will not treat the glare specially; this will yield false detection as seen in e.g. the third and fourth columns of Table 3. The other methods are more robust to the Sun glare, but fail to detect small details as well as thin clouds, like e.g. the method proposed by Huo and Lu. This leads to the second strong point of the proposed approaches, that mix high sensitivity and robustness to the Sun glare.

We can see that the graph cut method based on the fuzzy clustering failed on the whole image, and even on the regional image the results were not as satisfying and detailed as intended. This might be caused by the fact that this method is based on seeds, then the accuracy is highly dependent on the quality of the chosen seeds. The Otsu's method is the best performing threshold-based method, but the proposed approaches are slightly better or same in terms of detection. Needless to say, the Otsu's method yields a binary mask while our models yield a probabilistic map. As for the method proposed by Huo and Lu, it is good for overall look but not for deep/small details and its counter-glare ability is limited. Furthermore, when the sky is completely covered by clouds (last column of Table 3), we can see that most algorithms lost their accuracy on the boundaries, while the proposed models do not face any problem.

The results provided in Table 1 describe the performance of the proposed models versus the other algorithms. Both the ANN-based and k-NN-based models outperformed the state-of-the-art methods and algorithms in the two cases, which proves that those models are robust to the Sun glare (case 1) and accurate enough in regional situation (case 2). The ANN-based model high accuracy comes at the cost of long execution time as shown in Table 2, while the other algorithms and models share similar execution time.

Performance of the ANN-based model is generally better than the one of the k-NN-based model, in terms of accuracy they performed well versus other algorithms. Furthermore, the ANN-based model provides probabilistic map and performs better than the k-NN-based model around the Sun, but the ANN-based model's computation time is noticeably high. The k-NN-based model provides an accurate binary cloud mask and it is much faster.

5. Conclusion

Based on the presented observations in Section 4. of the paper, the ANN-based model provides high accuracy and multiple advantages such as probabilistic map, accuracy on circumsolar region, accuracy on boundaries, etc. The k-NN-based model performed well in the validation step where it provides the advantage of high accuracy and short execution time. In addition, the proposed models were able to deal with the problems posed by the threshold-based algorithms, such as the effect of the lens distortion, the Sun glare, over- and under-identification. To do so, as seen in Section 3. of the paper, the machine learning based models must utilize the appropriate features and parameterizations to outperform the threshold-based methods and provide accurate results.

Acknowledgments

This work is part of the SFERA III project (`https://sfera3.sollab.eu`), which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 823802.

References

[1] Nou J., Chauvin R., Eynard J., Thil S., Grieu S. *Towards the intrahour forecasting of direct normal irradiance using sky-imaging data*. Heliyon 2018;4(4):e00598.

[2] Jang HS., Bae KY., Park H., Sung D. *Solar Power Prediction Based on Satellite Images and Support Vector Machine*. IEEE Transactions on Sustainable Energy 2016;7:1255-263.

[3] Peng Z., Yu D., Huang D., Heiser J. *A hybrid approach to estimate the complex motions of clouds in sky images*. Solar Energy 2016;138:10-25.

[4] Long CN., Ackerman TP., Gaustad KL., Cole JN. *Estimation of Fractional Sky Cover from Broadband Shortwave Radiometer Measurements*. Journal of Geophysical Research 2006;111(D11):D11204.

[5] Li Q., Lu W., Yang J. *A Hybrid Thresholding Algorithm for Cloud Detection on Ground-Based Color Images*. Journal of Atmospheric and Oceanic Technology 2011;28(10):1286-296.

[6] Marquez R., Coimbra CFM. *Intra-hour DNI forecasting based on cloud tracking image analysis*. Solar Energy 2013;91:327-36.

[7] Liu S., Zhang L., Zhang Z., Wang C., Xiao B. *Automatic Cloud Detection for All-Sky Images Using Superpixel Segmentation*. IEEE Geoscience and Remote Sensing Letters 2015;12(2):354-58.

[8] Dev S., Savoy FM., Lee YH., Winkler S. *High-dynamic-range imaging for cloud segmentation*. Atmospheric Measurement Techniques 2018;11(4):2041-049.

[9] Boykov Y., Veksler O., Zabih R. *Fast approximate energy minimization via graph cuts*. IEEE Transactions on Pattern Analysis and Machine Intelligence 2001;23(11):1222-239.

[10] Kolmogorov V., Zabin R. *What energy functions can be minimized via graph cuts?*. IEEE Transactions on Pattern Analysis and Machine Intelligence 2004;26(2):147-59.

[11] Boykov Y., Kolmogorov V. *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*. IEEE Transactions on Pattern Analysis and Machine Intelligence 2004;26(9):1124-137.

[12] Otsu N. *A threshold selection method from gray-level histograms*. IEEE Transactions on Systems, Man, and Cybernetics 1979;9(1):62-66.

[13] Huo J., Lu D. *Cloud determination of all-sky images under low-visibility Conditions*. Journal of Atmospheric and Oceanic Technology 2009;26(10):2172-181.

[14] Tuominen P., Tuononen M. *Cloud detection and movement estimation based on sky camera images using neural networks and the Lucas-Kanade method*. AIP Conference Proceedings 2017;1850:140020.

[15] West SR., Rowe D., Sayeef S., Berry A. *Short-term irradiance forecasting using skycams: Motivation and development*. Solar Energy 2014;110:188-207.

[16] Cazorla A., Olmo F., Arboledas L. *Development of a sky imager for cloud cover assessment*. Journal of the Optical Society of America. A - Optics, Image Science, and Vision 2008;25:29-39.

[17] Linfoot A., Alliss RJ. *A cloud detection algorithm applied to a whole sky imager instrument using neural networks*. In: Proceedings of the 19th Conference on Probability and Statistics and Sixth Conference on Artificial Intelligence Applications to Environmental Science; January 20-24, 2008; New Orleans, USA. Available at: `https://ams.confex.com/ams/88Annual/techprogram/paper_129536.htm`.

[18] Rumelhart DE., Hinton GE., Williams RJ. *Learning representations by back-propagating er-*

rors. Nature 1986;323(6088):533-36.

[19] Prasath VBS., Alfeilat HAA., Hassanat ABA., Lasassmeh O., Tarawneh AS., Alhasanat MB., Salman HSE. *Distance and similarity measures effect on the performance of k-nearest neighbor classifier - A review*. Computing Research Repository 2017. Available at: `https://arxiv.org/abs/1708.04321`.

[20] Bentley JL. *Multidimensional binary search trees used for associative searching*. Commun. ACM 1975;18:509-17.

[21] Ronneberger O., Fischer P., Brox T. *U-Net: Convolutional networks for biomedical image segmentation*. In: Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI); October 5-9, 2015; Munich, Germany. Available at: `https://arxiv.org/abs/1505.04597`.

[22] Chollet F. *Keras: The Python deep learning API*. Available at: `https://keras.io/`.

[23] Bradski G. *The OpenCV Library*. Dr. Dobb's Journal of Software Tools; November 01, 2000.

[24] Caldas M., Alonso-Suárez R. *Very short-term solar irradiance forecast using all-sky imaging and real-time irradiance measurements*. Renewable Energy 2019;143:1643-658.