# Using sky-classification to improve the short-term prediction of irradiance with sky images and convolutional neural networks

Martinez Lopez, Victor Arturo; van Urk, Gijs; Doodkorte, Pim J.F.; Zeman, Miro; Isabella, Olindo; Ziar, Hesan

**DOI**
10.1016/j.solener.2024.112320

**Publication date**
2024

**Document Version**
Final published version

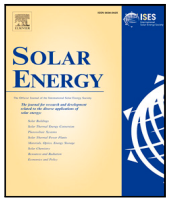
**Published in**
Solar Energy

Solar Energy 269 (2024) 112320

Contents lists available at ScienceDirect

Solar Energy

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



# Using sky-classification to improve the short-term prediction of irradiance with sky images and convolutional neural networks

Victor Arturo Martinez Lopez *, Gijs van Urk, Pim J.F. Doodkorte, Miro Zeman, Olindo Isabella, Hesan Ziar

*Photovoltaic Materials and Devices, Mekelweg 4, Delft, 2628CD, The Netherlands*

ARTICLE INFO

*Keywords:*
Sky-image processing
Irradiance nowcasting
Deep learning
All-sky images

ABSTRACT

Clouds moving in front or away from the sun are the leading cause of irradiance variability. These variations have a repercussion on the electricity production of photovoltaic systems. Predicting such changes is essential for proper control of these systems and for maintaining grid stability. Images from the sky have proven to help with short-term solar irradiance forecasting, especially when combined with artificial intelligence. Nevertheless, these models tend to smooth the irradiance fluctuations. We propose a forecasting model to predict the clear-sky index in a forecast horizon of 20 min with a 1-minute resolution. Our model, based on a classifier to determine the sky conditions and, on an optical flow, applies an artificial intelligence model explicitly trained on each class of sky conditions. This strategy has an equivalent performance to an unclassified model and a forecast skill between 5 and 20% with respect to the smart persistence model for most classes of sky conditions while requiring considerably less training data. Although our model reduces the overall predicting error, it still has difficulties predicting irradiance changes and mainly overcast days. Our classifying strategy can be applied to other models targeting different objectives to predict sudden changes in either irradiance or power related to photovoltaic systems.

## 1. Introduction

Clouds influence the irradiance from the Sun as it travels through the atmosphere with their shape, size, and constitution [1]. One of the most obvious effects is that they block the sunlight, reducing the available Sun power reaching the ground. They can also cause the opposite: increasing (or *enhancing*) the irradiance above the theoretical irradiance at the top of the atmosphere [1,2], caused by scattering inside the cloud [3] which leads to an increase in the diffuse irradiance [2]. Cloud enhancement, for example, can cause blowing of fuses (leading to production loss), overloading of the inverters (leading to efficiency loss) [4] or underestimation of maximum Photovoltaic (PV) production [5]. Therefore, the movement of clouds in front of the Sun produces fluctuations in the solar irradiance available for PV production. These fluctuations occur even in the sub-minute range, negatively affecting the PV systems in the form of significant PV power ramps which impose a challenge to grid operators [6], and the variability of solar irradiance affect the maximum power point trackers' operation resulting in efficiency loss [7].

Irradiance forecasting is a method to alleviate some of the challenges of variations caused by moving clouds. It is applied in the operation of PV systems with benefits such as increasing battery lifetimes of PV systems that incorporate them [8], and it has proven to be a crucial requirement of grid operations with high renewable energy penetration [9].

In the very short term, defined by a time horizon from seconds to maximum 30 min [10], images from the sky provide valuable information such as cloud identification (e.g., [11]), cloud classification (e.g., [12]), and cloud tracking (e.g., [13]). Such sky images are obtained with an All-Sky Imager (ASI) (a camera equipped with a fish-eye lens facing the sky) which takes a picture of the sky in a particular location. This has the advantage of retrieving higher resolution information on the site than satellite images (which cover a broader area), ceilometers, and radars (which have lower local resolution) [14].

The core idea behind predicting irradiance using sky images is to (1) identify clouds, (2) determine how they are moving, and (3) assess whether they will obstruct the Sun [13,15–19]. Additional information (or features) can be given to the model to improve the prediction. For example, in [16], the authors included not only the cloud pixels and their movement but also identified the cloud edges and corners related to the cloud type and the pixel intensity of the clear and cloudy pixels.

* Corresponding author.
*E-mail address:* V.A.MartinezLopez@tudelft.nl (V.A. Martinez Lopez).

https://doi.org/10.1016/j.solener.2024.112320
Received 17 May 2023; Received in revised form 14 December 2023; Accepted 5 January 2024
Available online 19 January 2024

**Lists of terms**

| | |
|---|---|
| AI | Artificial Intelligence |
| ASI | All-Sky Imager |
| ANN | Artificial Neural Network |
| CC | Cloud Cover |
| CNN | Convolutional Neural Network |
| GHI | Global Horizontal Irradiance |
| HYTA | Hybrid Thresholding Algorithm |
| LSTM | Long Short-Term Memory |
| nBRR | Normalized Blue-Red Ratio |
| RNN | Recurrent Neural Network |
| RGB | Red-Green-Blue |
| RMSE | Root Mean Squared Error |
| PV | Photovoltaic |

Convolutional Neural Networks (CNNs) can automatically detect these features [20] and can be used to perform irradiance forecasting, such as SUNSET [21,22], SolarNet [23] and ECLIPSE [24]. SUNSET is a CNN that receives a sequence of sky images and historical PV power production to forecast a 15-min horizon [21,22]. In highly variable moments, the model predicts the average output PV power in the prediction horizon [22]. SolarNet follows a different approach, as it was designed to predict over a wide range of time horizons using 6 CNNs trained for different forecasting horizons and a single sky image as input. It can predict up to 60 min with a 10-min resolution [23].

The prediction of irradiance ramps can be done by predicting the future position of the cloud pixels and determining whether or not these pixels will block the Sun [24,25]. ECLIPSE uses this approach to predict up to 10 min with a resolution of 2 min, and it tackles specifically the prediction of important irradiance ramps. It achieves this by combining temporal information (using a sequence of images) along with spacial information (location of the clouds and the Sun) [24].

CNNs are not the only architecture for implementing sky image forecasting. Especially when including sequences of images, other Artificial Intelligence (AI) architectures such as Long Short-Term Memory (LSTM) or Recurrent Neural Network (RNN) can also compete and even outperform CNNs [26–30]. Nevertheless, a CNN has a relatively simpler architecture than LSTM or RNN [31] and the improvement in the error reduction that the former offer is marginal (around 2%) on forecasting horizons greater than 10 min [27,30]. In some cases, the CNN can even perform better than the networks incorporating temporal information [30].

In general, sky-image-based forecasting methods outperform persistence models in many sky conditions, although their performance for ramp detection is still one of the main challenges and an ongoing research topic [28]. This holds even for models based on artificial intelligence, as they tend to output a smooth prediction of irradiance that effectively misses steep ramps [26,27,29] or they predict an average of the output [22,27]. Additionally, the output from AI models tends to lag from the actual measurements [27]. Even more, deep learning models are sensitive to weather conditions [29,31].

To tackle these problems, hybrid models have been proposed. Some combine image processing techniques to extract the evaluating features and then feed them to an LSTM which performs the forecast. This model can solve the problem with time alignment and detect some ramps [19].

AI-based hybrid models can also be created by training different models to cover various scenarios. One of these is based on implementing a classifier for four cloud types. Then four different models are trained using Support Vector Regression and Multiple Linear Regressor. The final prediction is made by determining the prevailing cloud type on a sky image and then applying the corresponding model to that cloud type [18]. The authors claim an improvement in the error not only with respect to the persistence model but also to models based on Artificial Neural Network (ANN)'s [18].

The classification can also be done in terms of time. This is achieved by training the same model aiming at different forecasting horizons and then combining their output using a linear combination [32]. This strategy is also useful to deal with models which have many inputs, such as images and irradiance measurements. In this case, the time resolution of the data can have different sampling times. The use of different models and combining their output (ensemble learning) can help to deal with these situations [33] or facilitate training with fewer data by using transfer learning [34].

The purpose of this work is therefore to develop a mixed model based on sky images (Section 2.1), a sky classifier (Section 2.3), and a CNN (Section 2.4).

As opposed to ensemble learning techniques, which are also based on classification, our model does not combine the output of different models but uses the output of a specialized model on a particular sky class which leads to a reduction in the prediction error (Sections 3 and 4).

## 2. Model structure

Our model consists of three main modules: image processing, sky classifier, and artificial intelligence. Fig. 1(a) illustrate the structure of the model. The model receives a sky image and a clear-sky index (defined in Section 2.3) and the coordinates of the Sun in the image. The image is manipulated to obtain the cloud cover (Fig. 1(b)). Using this and the clear-sky index it is classified in one of five categories. The image is then fed, along with the clear-sky index to one of the five CNNs trained for the particular class the image belongs to. The output is a minute resolution of the Global Horizontal Irradiance (GHI) over the next 20 min.

### 2.1. Image processing module

The image processing module processes the images that will serve as input to the classifier. After the processing, the Cloud Cover, the clear-sky index and the position of the Sun within the image are obtained.

#### 2.1.1. Cloud identification

A digital color image (such as that captured by an ASI) is a discrete representation of the intensity of the light impinging upon the sensor of the camera [35]. The *color* of the digital image is obtained by superimposing a checkered pattern where three filters (red, green and blue) are embedded (Bayer pattern) [36] to the sensor or having three separate red, green and blue sensors [35,36]. The reason for these three colors is that the model used to represent each pixel in the image defines them as primary colors and is called the Red-Green-Blue (RGB) model [35]. Using the RGB model results in a 3D representation of an image where each matrix dimension represents one primary color [35,36].

The color channels of a sky image can be used to identify clouds (cloud segmentation). On a clear day when looking at the sky, it appears blue. If, at this instant, a sky image is taken, the value of the blue pixels will be higher than the Red pixels, assuming that our image is mapped in the RGB color space. On the contrary, a cloud-covered sky image will have a greater count of red pixels, making the clouds appear white or gray [12]. Using this fact, the clouds can be detected by using a ratio (e.g., [37]) or a difference (e.g., [12]) between the image's red and blue channels and then establishing a threshold above which the pixels will be classified as clouds while the rest as sky. A combination between the difference and ratio of the red and blue channels can also be used to establish such threshold. This is the case of the Normalized Blue-Red Ratio (nBRR) which is defined in Eq. (1).

$$\mathrm{nBRR} = \frac{\mathrm{Blue} - \mathrm{Red}}{\mathrm{Blue} + \mathrm{Red}} \tag{1}$$

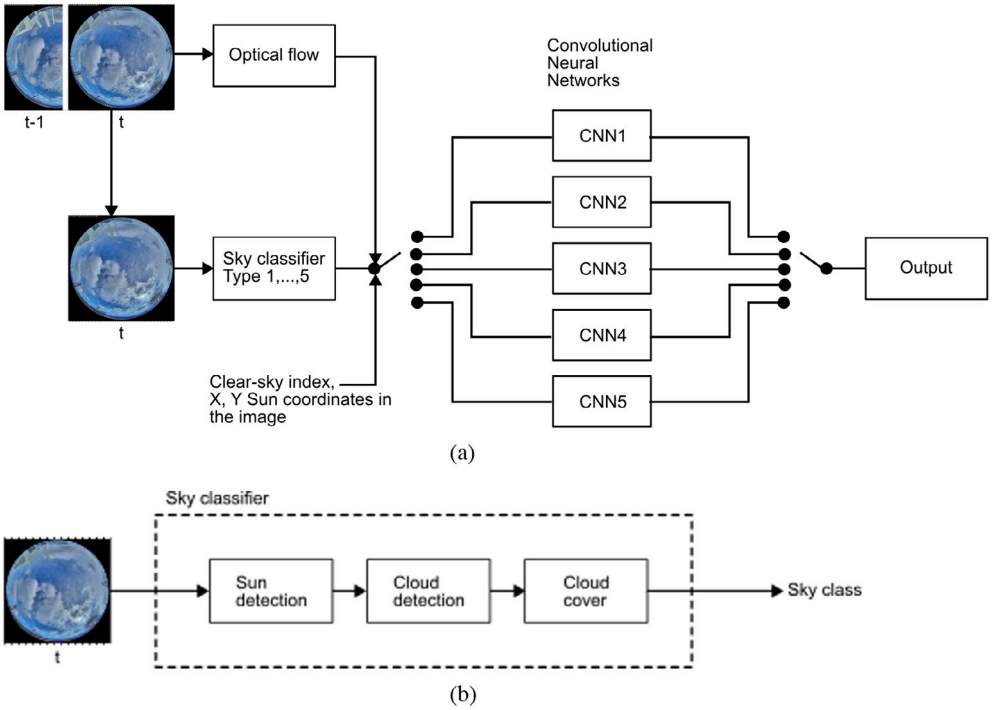



**Fig. 1.** (a) Overview of the prediction model. (b) Detail of the sky classifier module.

The problem with the thresholding method is that it fails to detect very thin clouds, has a large misclassification rate due to the overlap of the pixel distribution between clear and cloudy conditions, and performs poorly on pixels located around the sun [11,38].

The fact that partially cloudy, overcast, and clear days exhibit different nBRR pixel distributions can improve classification accuracy.

The nBRR pixels of partly cloudy images exhibit a bimodal distribution, while this feature disappears on overcast and clear-sky images. The bimodality can be identified from the standard deviation of the histograms [38]. Depending on the sky condition, the thresholding technique can be adapted for partially cloudy (bimodal distribution), partially or clear-sky images (unimodal distribution) [38]. This is the core idea behind the Hybrid Thresholding Algorithm (HYTA), which achieves an overall accuracy of 88.53%. However, HYTA excludes sun pixels, because very few sun pixels, can shift the whole histogram due to their large intensity. For this reason, the improved algorithm version, HYTA+ [11], was developed. This method adds another classification step by first determining whether a cloud covers the sun. Following the logic of the HYTA, the algorithm of [39] is based on sun coverage identification and sky condition classification. We build upon this model because of its simplicity and accuracy. Our implementation of the algorithm [39] is explained below.

1. Sun identification: Determines whether the sun is blocked or not.

    1.1. Identification of the sun. Eq. (2) returns the coordinates (in pixels) of the sun in the image based on the position of the sun in the sky described by its azimuth ($A$) and altitude ($a$) (in degrees). It is based on the projection of the sky dome on a plane [40]. Once the sun is identified, the image is cropped, considering a radius of 200 pixels around the sun's coordinates

$$\begin{aligned} X_{\text{sun}} &= X_{\text{center}} + R_{\text{image}} \times \sin(A)\cos(a) \\ Y_{\text{sun}} &= Y_{\text{center}} + R_{\text{image}} \times \cos(A)\cos(a) \end{aligned} \tag{2}$$

    1.2. Obtaining the sun coverage ratio. The cropped image is converted to grayscale. The sun is detected by identifying very bright and white pixels with a value exceeding 210. The value of 210 is the minimum average pixel intensity of the region around the sun of 150 clear-sky images [39] (Note: the maximum pixel value of a grayscale image is 255). Then, the ratio of white pixels in the whole image is calculated. The sun is clear if at least 95% of the pixels of the cropped image are white [39].

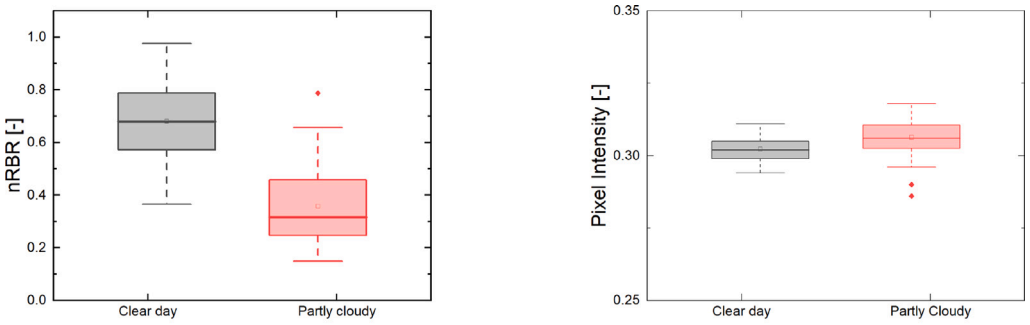
2. If the sun is blocked:

    2.1. Apply a fixed threshold of 0.6 on the Red/Blue ratio to identify the clouds.

3. If the sun is *not* blocked:

    3.1. Determine if there are cloudy pixels in the image. The original work [39] set a fixed threshold of 0.6 on the pixel intensity of the image, excluding 200 pixels around the Sun. We decided to apply the average of the normalized Blue-Red Ratio (Eq. (1)). Our threshold value was set at 0.45, which corresponds to the 75th percentile of the pixel values of 100 partially cloudy days. This means that values above 0.45 will be classified as clear. Fig. 2 shows the statistics of both figures. Using the average of the nBRR (with the sun blocked as in [39]) results in lower overlapping than using the original pixel intensity of the Blue-Red ratio. Our chosen value ensures that most of the clear-sky images are correctly classified, albeit sacrificing accuracy for the partly cloudy days which statistically is 25% of the evaluated images. Despite this considerable large value, a visual inspection of the images suggest that the misclassified images show small clouds, not covering the sun, and a deep blue sky, which is acceptable for the clear-sky classification. Lowering the threshold to the minimum of the observed clear-sky images, results in worst classification as completely overcast where the sun is visible (foggy conditions, for example), can be misclassified as clear-skies.

    3.2. For images classified as partly cloudy, apply superpixel segmentation method following the approach of [41]. The




(a) (b)

**Fig. 2.** Statistics of 100 clear-sky images and 100 partly cloudy images. (a) Mean pixel value using the nBRR and (b) Pixel intensity using the Blue-Red ratio as in [39]. Note that with the pixel intensity, there is a greater overlap between clear day and partially cloudy images.

superpixel segmentation method is, as the name implies, a segmentation strategy that will aggregate pixels with similar characteristics of brightness, texture, and contour smoothness into blocks called *superpixels*[42]. The superpixels offer a segmentation that is closer to how a human would segment the same image manually [42]. The superpixel segmentation method proposed in [41] essentially obtains a global threshold to separate clouds from sky for the whole image using the Otsu algorithm, which is an automated method for finding the optimal threshold. It is based on the fact that the pixels of high-contrast grayscale images have a bimodal distribution. Thanks to this fact, two classes can be created based on the humps of the bimodal distribution. The Otsu thresholding iterates over the image histogram and in each pass separates these two classes by evaluating the probability of each class and its variance, which are dependent on the threshold. The optimal threshold will maximize the variance between both classes [43].

After the global threshold is obtained, the cloud detection algorithm applies superpixel segmentation to the image. For each superpixel, a local threshold is obtained and then compared against the global threshold. This allows establishing a decision on the threshold that should be applied to each superpixel. By doing this, the identification of clouds is performed *per superpixel* instead of globally. Hence a more accurate identification is achieved [41].

The cloud cover is the count of cloudy pixels to the total image pixels. It is important to mention that we masked the pixels of the image corresponding to the horizon to avoid counting these objects as clouds. We also considered only images with a sun elevation higher than 10° over the horizon. Some examples of cloud identification can be found in Appendix A.

### 2.2. Optical flow

The cloud motion is incorporated by calculating a dense optical flow. This tool allows the tracking of pixels between two consecutive images by calculating the displacement of pixels (or group of pixels) between two consecutive images [44]. In particular, we applied the Farneback method [45]. This method approximates the pixel intensity of a greyscale image as a quadratic polynomial. Then, it assumes that the consecutive image is described by the same polynomial function but displaced by a *displacement vector field*. The Farneback algorithm aims to retrieve this displacement vector for every pixel [45].

This algorithm is implemented in the Python package OpenCV as the function `calcOpticalFlowFarneback` and returns the displacement vector for every pixel in terms of magnitude and angle [46].

The optical flow has been applied to irradiance forecasting by first determining the full displacement vector field and then finding an equivalent vector which describes the motion of the main cloud block near the Sun. This can be later related with the direct normal irradiance to predict occlusions of the Sun [47].

### 2.3. Sky classifier

The sky-classifier is built upon the cloud cover retrieved from an instantaneous sky image and the clear sky index obtained simultaneously. To recall, the clear-sky index is an adimensional number relating the measured GHI ($GHI_m$) against the GHI obtained from a clear-sky model $GHI_0$ at any instant. Mathematically, it is expressed as in Eq. (3). The clear-sky index removes the seasonal variation of irradiance, allowing for an analysis for the changes in irradiance caused by clouds [48].

$$k_c = \frac{GHI_m}{GHI_0} \tag{3}$$

For this reason, we hypothesized that there should be an almost linear relation between the cloud cover and the clear-sky index (i.e. the larger the cloud cover, the lower the clear-sky index). Fig. 3(a) shows the scatter plot of the clear-sky index and the cloud cover for our dataset. From this image the hypothesized relation cannot be verified.

Pfister, et al. [1] described the relation between the cloud cover, the irradiance, and the clear-sky index. In their results, two bands, or clusters can be observed: The first occurs at very high clear-sky indices spreading over a wide range of cloud cover, and the second one at very low clear-sky indices scattered from cloud cover values between 40 and 100% [1]. These bands are not identifiable in our dataset (Fig. 3(a)). The potential reasons for these differences are how they processed the irradiance data. In both works, 1-min data was used; however, in the work of Pfister, et al. [1], the average of 1-s measurements over the minute is taken, while in our work, we took a single irradiance measurement that coincides with the instant that the sky-image is taken. Our approach results in measurements with higher variability (hence, the scattering observed in Fig. 3(a)). Nevertheless, with this strategy, we do not include events that are not captured by the camera at that instant. For example, at a particular moment, a sky image might show a clear sun. However, in the course of the minute between images, the sun might have been obscured. By averaging the 60 s of the minute, this information is incorrectly assigned to the unobscured image.

Since we do not observe clusters of data, we applied K-means (50 clusters) and then manually grouped these clusters together to obtain a final classification consisting of five classes. The logic for the cluster strategy is as follows:

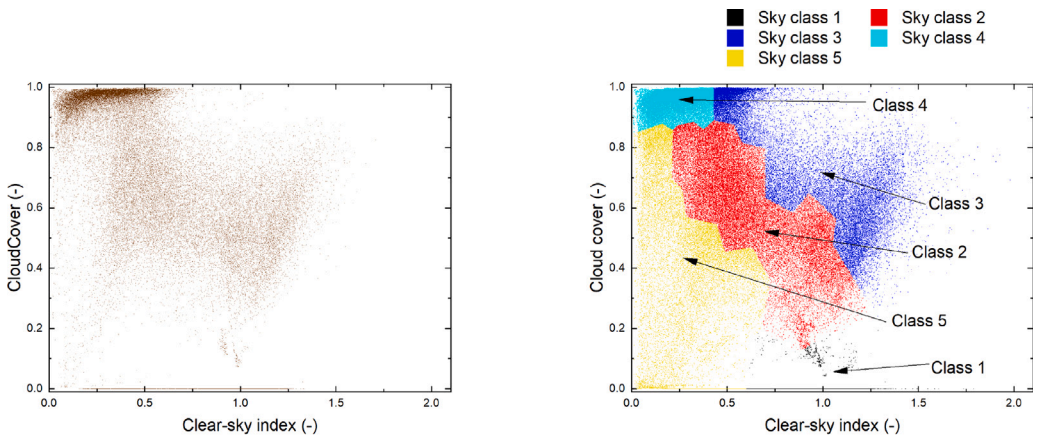



**Fig. 3.** (a) Scatter data from instantaneous clear-sky index and cloud cover. (b) Classified dataset. Observe that there are some data points with very high clear-sky index values. Although it is not unexpected to have values of the clear-sky index exceeding unity (due to cloud enhancement), extreme values (close to two) correspond to a mismatch between the clear-sky index model and the measurements. This occurs mainly in the early morning or late evening.

- Class 1: *clear-sky* images (high clear-sky index, very low cloud cover)
- Class 2: *partly cloudy* images following an almost linear relation of the clear-sky index and cloud cover.
- Class 3: *overcast days with high irradiance* (high cloud cover, high clear-sky index). This group includes the overirradiance phenomenon.
- Class 4: *overcast* images (Very low clear sky-index, very high cloud cover)
- Class 5: *Partly clear skies with sun obscured* images (low cloud cover, low clear-sky index)

Fig. 3(b) shows the resulting clusters after the classification logic.

### 2.4. Artificial intelligence module

The last module of the model is the one that makes the prediction. It implements a CNN, which processes a low-resolution, color sky image merged with the optical flow information at every pixel, resulting in a 64 × 64 pixels image with five channels. The first three channels of the image correspond to the red, green, and blue channels, respectively. The last two channels map the magnitude of the pixels' velocity (channel 4) and their direction (channel 5) obtained from the optical flow. The architecture of this network is a convolutional neural network with only one convolutional step, which processes the image. Then, the output is flattened to a one-dimensional vector and concatenated with the auxiliary input data: the X and Y coordinates of the Sun's position on the image and the clear-sky index. The auxiliary data combined with the flattened image is fed to a three-layer neural network. The output is a vector containing the 20-min ahead clear-sky predictions.

This architecture is trained 6 times: one considering the whole dataset (reference) and one time per sky class. In short, the full pipeline of the model is

1. Get sky-image
2. Using the image processing module, obtain: cloud cover, clear-sky index and position of the Sun in the image.
3. Obtain the class of the image.
4. Use the CNN model trained on the sky-image class to get the final prediction

## 3. Results and discussion

The data set used in this study contains sky images taken using a CMS Shreder ASI-16/50 All Sky Imager located at the monitoring station of the Photovoltaic Materials and Devices. The irradiance was measured with a silicon-based pyranometer provided by the manufacturer of the sky camera. The full dataset contains data from 29-03-2021 until 24-02-2022. For training purposes, we used a reduced sample of this dataset containing 80 000 randomly sampled sky images with their associated irradiance values and with a sun elevation higher than 10° above the horizon. The distribution of the dataset can be found in Appendix B, Fig. B.1. The clear-sky index was calculated using the python pvlib function `clearsky` with the *solis* model [49]. A reference model, hereafter called "full model", was trained to study the potential improvement of the classifier. This model has the same architecture of each CNNs used in the classifier but is trained with a random sample of 64 000 data points containing data from all classes. The rest of the data points are used for testing the model's performance hyperparameter tuning was performed for any of the models, as the study's main goal is to obtain a comparison of the advantages of the classifier. Each model was trained for 20 epochs using the mean squared error as the loss function. It is worth mentioning that the architectures of all models remain the same.

The models for each class were trained using a sample that contains the same amount of data points regardless of the class (9536 datapoints). Similarly, as the reference model, each model was trained for 20 epochs with no hyperparameter tuning.

Fig. 4 shows the forecast skill defined in Eq. (4) [27], where $\mathrm{RMSE}_{\mathrm{eval}}$ is the RMSE of the model under evaluation and $\mathrm{RMSE}_{\mathrm{baseline}}$ is the RMSE of the baseline model (either the model trained with the whole dataset (Full model) or the persistence model). Fig. 4(a) compares both, the full model (model trained with the whole dataset) and the classified models against the smart persistence model, which assumes that the clear-sky index will remain constant throughout the forecast horizon. Note that in Class 4 (overcast) the persistence model performs better than the full and classified models, and in Class 5 (mainly overcast with patches of blue), the smart persistence model performs better than the classified model. The classified model is the best option for predicting Class 1 (clear skies), while the full model is the best option for predicting classes 2, 3, and 5 as seen in Fig. 4(b).

$$FS = \frac{\mathrm{RMSE}_{\mathrm{baseline}} - \mathrm{RMSE}_{\mathrm{eval}}}{\mathrm{RMSE}_{\mathrm{baseline}}} \times 100 \tag{4}$$

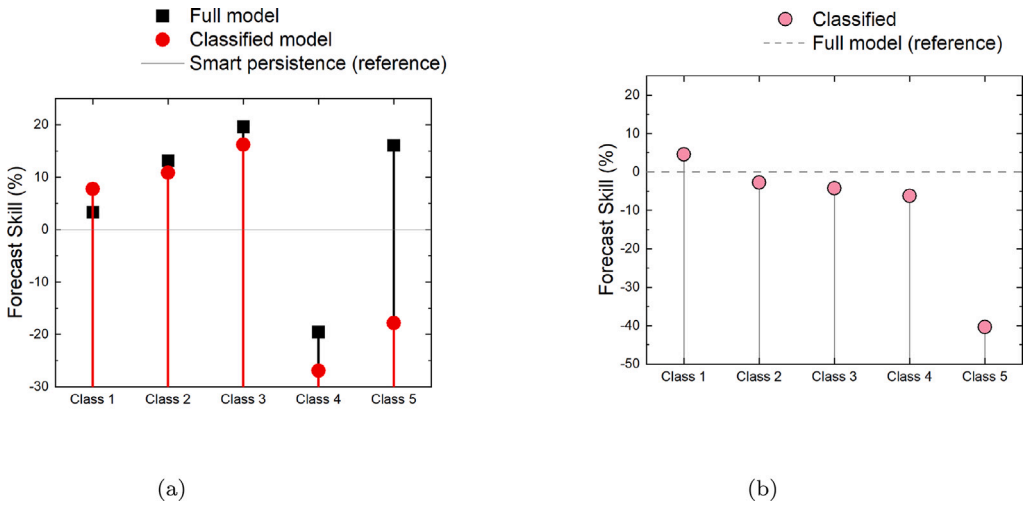



**Fig. 4.** Forecast skill per class using the RMSE. (a) Performance per class of the classified and the full models (model trained with the full dataset) against the smart persistence model. The forecast skill of the smart persistence is zero, and is marked with a solid line. (b) Performance of the classifier against the full model (model trained with the full dataset). The forecast skill of the full model is zero because it is the reference in this particular case. It is marked with a dashed line.

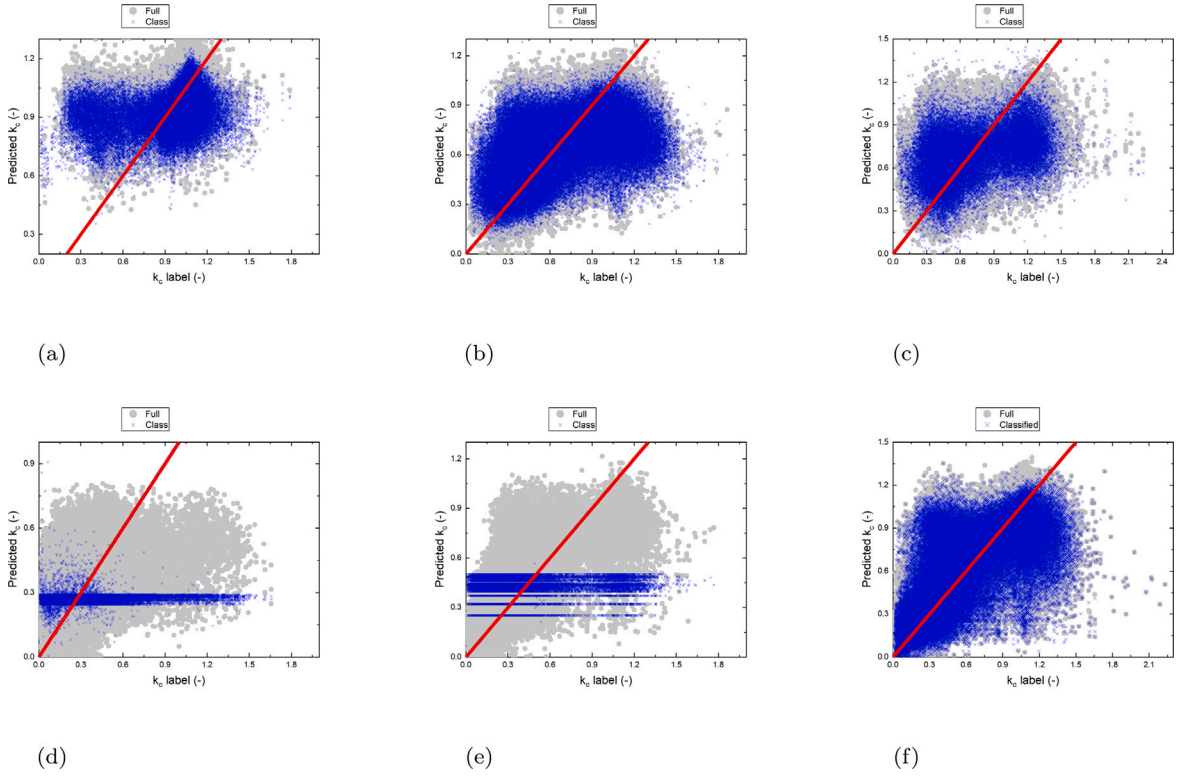



**Fig. 5.** Comparison of the predicted and expected value between each class's full and classified models. (a) Class 1, (b) Class 2, (c) Class3, (d) Class 4, (e) Class 5, and (f) an application of the classifier on the whole dataset, where Class 1 was predicted with a class 1 CNN, classes 2,3 and 5 with the full model and Class 4 with the smart persistence model. The prediction on the same dataset with the full model is also shown. The red line on all graphs indicates a perfect prediction. Points above the red line indicate that the model overestimates the labeled value and points below the red label indicate that the model underestimates the labeled value.

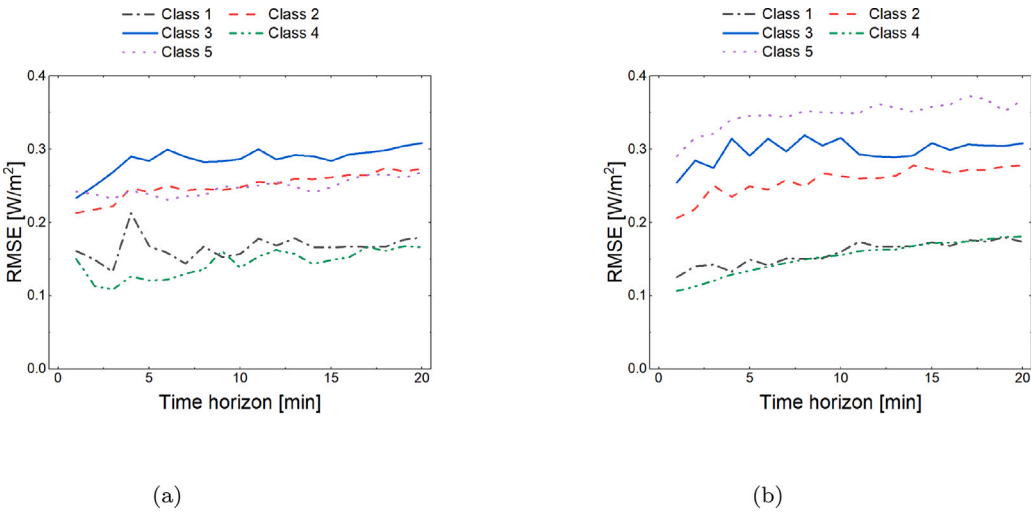



**Fig. 6.** RMSE as a function of the prediction horizon for (a) the reference model performing in each class, and (b) the classified model.

With the results of Fig. 4, it is evident that the full model outperforms the classified model in almost every class. The performance of the classified and the Full model are very similar, with a slight improvement in classes 2 and 3. The full model exhibits a larger scattering between the expected and the predicted values (Fig. 5), while in the classified model, the points are more clustered. Because of this, the full model has more points close to the perfect prediction line resulting in lower error, especially in large clear-sky indices. We can conclude that the full model is more *accurate*, but the classified model is more *precise* in classes 1, 2, and 3 (Figs. 5(a), 5(b), and 5(c)).

The classified model fails on classes 4 and 5 (Figs. 4(a), 5(d), and 5(e)), and the smart persistence model is, by far, the best option for predicting overcast days (Class 4) (Fig. 4(a)).

Fig. 6 shows the evolution of the RMSE as the forecast horizon increases. The error increases as the forecast horizon increases, as expected. For the full model (Fig. 6(a)), the increase of the error is minimal, while for the classified model, it is more evident (Fig. 6(b)). The models tend to ignore the changes in irradiance, and minimize the error along the whole prediction horizon. For this same reason, the classified model fails in classes 4 and 5, as the model predicts always the same value (Figs. 5(d) and 5(e)).

Fig. 7 shows the application of the model to predict the Global Horizontal Irradiance (GHI). Multiplying the model's output with the modeled clear-sky gives the value of the GHI. During the clear part of the day, the classified model outputs a smoother prediction closer to the measured GHI. In the variable part of the day, both models predict a trend rather than the actual value of irradiance, and neither of them captures the peaks and valleys of the measured irradiance. The Figure also shows the moments where the classified model failed (Class 5, in both cases).

The use of the optical flow to incorporate temporal information of the cloud movement did not show any improvement. There was no difference in the training error of the reference model with and without the optical flow. This suggests that the optical flow does not provide useful information to the model and can be ruled out as a predictor.

One potential explanation for this, is that the optical flow used in this work (the Farneback method [45]) tracks the movement of *pixels*. But this does not necessarily implies that it tracks the *clouds*. The Farneback method may provide too much information that the models interpret as noise. Another potential reason for the poor performance of the optical flow is that clouds do not move as a block, but they

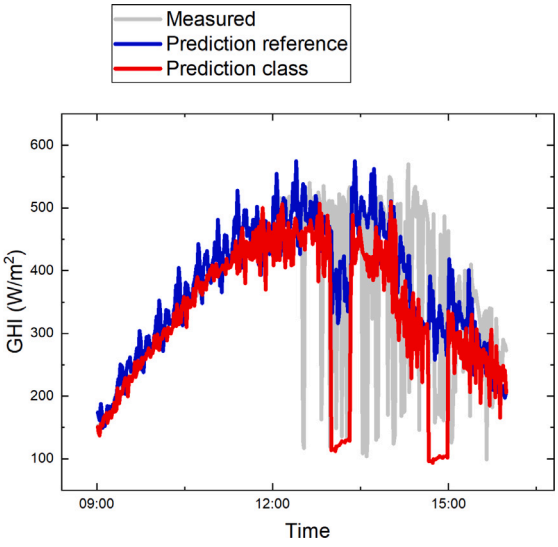



**Fig. 7.** Example of 1 day with prediction using the classification model and the reference model. The measurements for such a day are also included as a guide. Note that both models have difficulties during the variable part of the day.

also change, which is not captured by the optical flow method. This makes the correlation between past images and future measurements a difficult task [25]. Nevertheless, although including the temporal information of clouds in forecasting models leads to improved prediction accuracy as proved by [27,28], this improvement is marginal and might not justify the extra effort needed for such an improvement.

Despite an overall higher performance of the full model, our approach has the advantage of improving the prediction by combining different models. The results presented here assumed that the prediction module is the same for the five classes, but the classifier approach can also be combined with different models. For example, use the classified model for Class 1, the reference model for Classes 2, 3, and 5, and the smart persistence for Class 4.

Fig. 5(f) shows an example of such an ensemble. When the predicted instant corresponds to Class 1, the predictor is the specialized CNN of Class 1. When it corresponds to classes 2, 3, and 5 the predictor is the

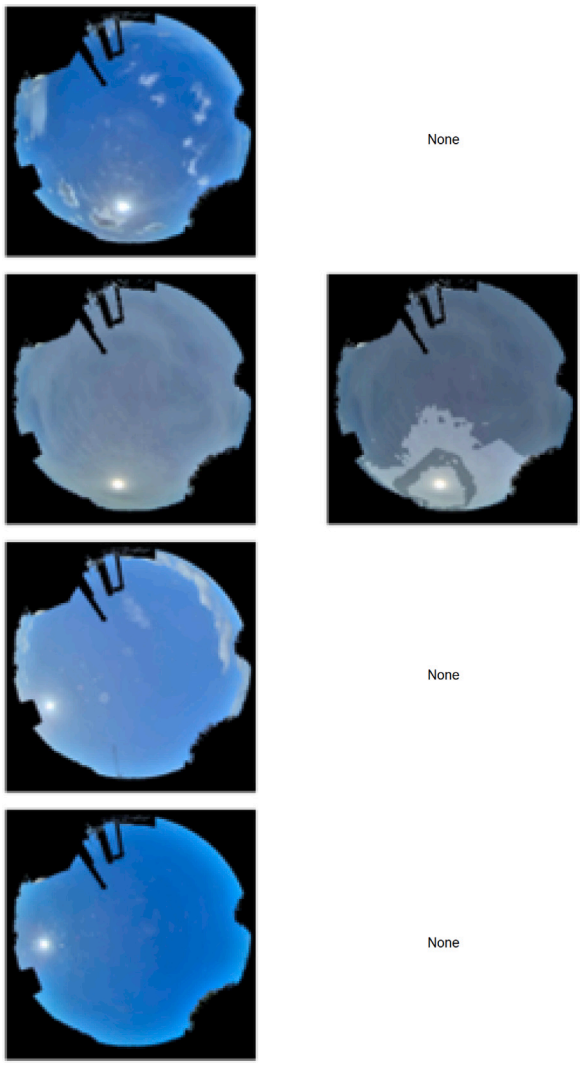



**Fig. A.1.** Random examples of Class 1 cloud identification. The images on the left column are the original images. On the right, the clearer areas identify "Cloud" pixels while the darker are "Not cloud" pixels. The label "None" indicates that there are no clouds identified in that image. Note that the horizon has a mask of black pixels.

full model. The predictor is the smart persistence model for instants that correspond to Class 4. This strategy reduces the error by 2.05% with respect to the full model. Although the improvement is not large, it proves that using specialized systems for different sky classes reduces the prediction error. This opens the possibility of implementing other prediction algorithms even better than our full model for a particular class, while using the classifier concept.

Note that, except for classes 4 and 5, the performance of the full and classified models is considerably close. Even if the full model outperforms the classified model, this could be acceptable, considering that the amount of data to train the classified networks is only a fraction of the data needed for the full model (around 14% per class, 42% for classes 1,2, and 3, 70% for the five classes). Naturally, increasing the number of training examples also helps the prediction accuracy of the classified model.

Our model (both the reference and classifier) still suffers from limitations regarding the prediction of ramps. This phenomenon is not particular to our model, but is present in many prediction models using sky images [22,27–29].

Improving the accuracy of predictions directly impacts applying these methods to control PV systems. Grid operators impose restrictions on the ramp rate (i.e., they impose a limit on the rate of change in the plant's output power) [50–53] that are typically handled with energy storage systems [51,53]. Incorporating prediction can help in reducing the size of electric storage [52], increasing its lifetime with proper control [53] or even eliminating [50–52] it, which has a direct effect on the cost of the system [53]. However, these benefits are only achieved if the prediction accuracy is high. Otherwise, it might be detrimental as the curtailed power is higher because of an incorrect prediction or a larger energy storage system is needed to compensate for the inaccuracy.

## 4. Conclusion

We presented a model for minutely predicting the clear-sky index in a 20-min horizon. Our model is based on sky images, a sky classifier, and a convolutional neural network. The reference model has a superior performance than the classified model. The classified model outperforms the reference model in Class 1 with a forecast skill of 5%, while the difference between the classified and reference is small for most of the classes. The classified model fails in overcast and mostly overcast days and partially clouded moments still represent a challenge

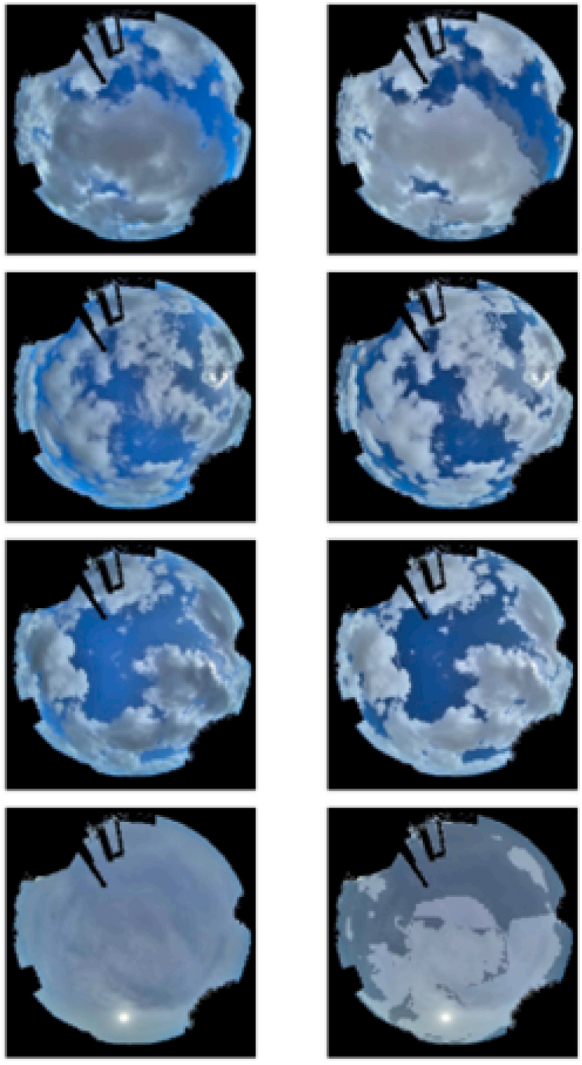


**Fig. A.2.** Random examples of Class 2 cloud identification. The images on the left column are the original images. On the right, the clearer areas identify "Cloud" pixels while the darker are "Not cloud" pixels. Note that the horizon has a mask of black pixels.

for both models when the smart persistence model outperforms the most complex artificial intelligence models. The extra effort needed to classify the sky conditions and train more models is justified because using specialized models per sky class can yield better accuracy than a universal one for all classes.

## CRediT authorship contribution statement

**Victor Arturo Martinez Lopez:** Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Gijs van Urk:** Conceptualization, Methodology, Software, Investigation. **Pim J.F. Doodkorte:** Conceptualization, Methodology, Software, Investigation. **Miro Zeman:** Funding acquisition, Resources. **Olindo Isabella:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Resources. **Hesan Ziar:** Conceptualization, Writing – review & editing, Supervision.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This activity is co-financed by Shell and a PPP-allowance from Top Consortia for Knowledge and Innovation (TKI's) of the Dutch Ministry of Economic Affairs and Climate in the context of the TU Delft e-Refinery program.

## Appendix A. Examples of cloud detection

This section shows examples of the cloud identification of each class. The examples were taken randomly from the training dataset. In all cases, the images on the left correspond to the image under evaluation. On the right is the cloud identification output. The cloud pixels are highlighted with clear areas. Figs. A.1, A.2, A.3, A.4, and A.5 show the examples per class.

## Appendix B. Dataset distribution

Fig. B.1 shows the distribution of the dataset in terms of the sun's elevation. As a remainder, the minimum sun elevation that we considered is 10°.

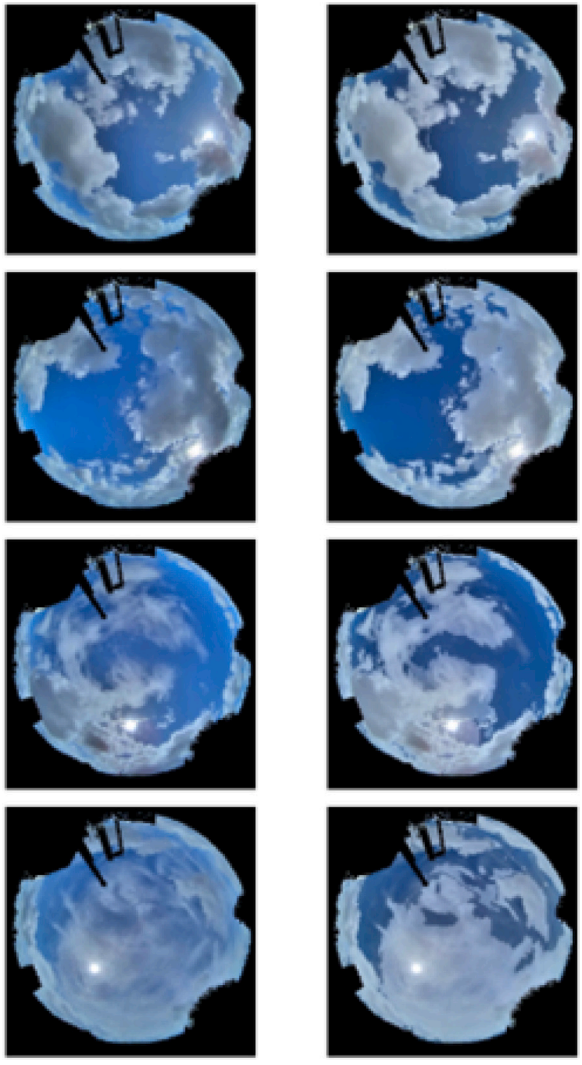


**Fig. A.3.** Random examples of Class 3 cloud identification. The images on the left column are the original images. On the right, the clearer areas identify "Cloud" pixels while the darker are "Not cloud" pixels. Note that the horizon has a mask of black pixels.

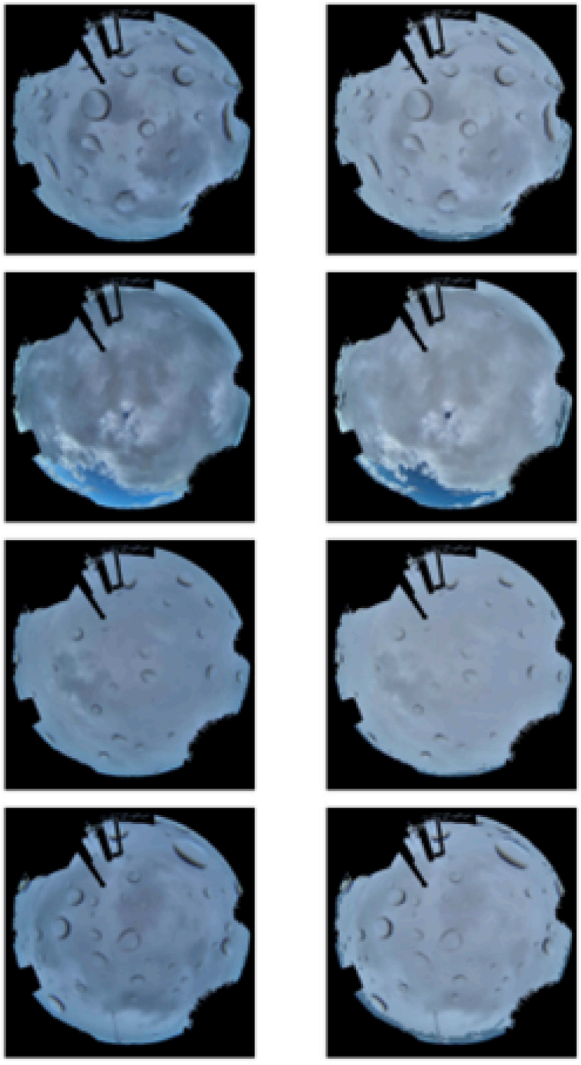


**Fig. A.4.** Random examples of Class 4 cloud identification. The images on the left column are the original images. On the right, the clearer areas identify "Cloud" pixels while the darker are "Not cloud" pixels. Note that the horizon has a mask of black pixels.

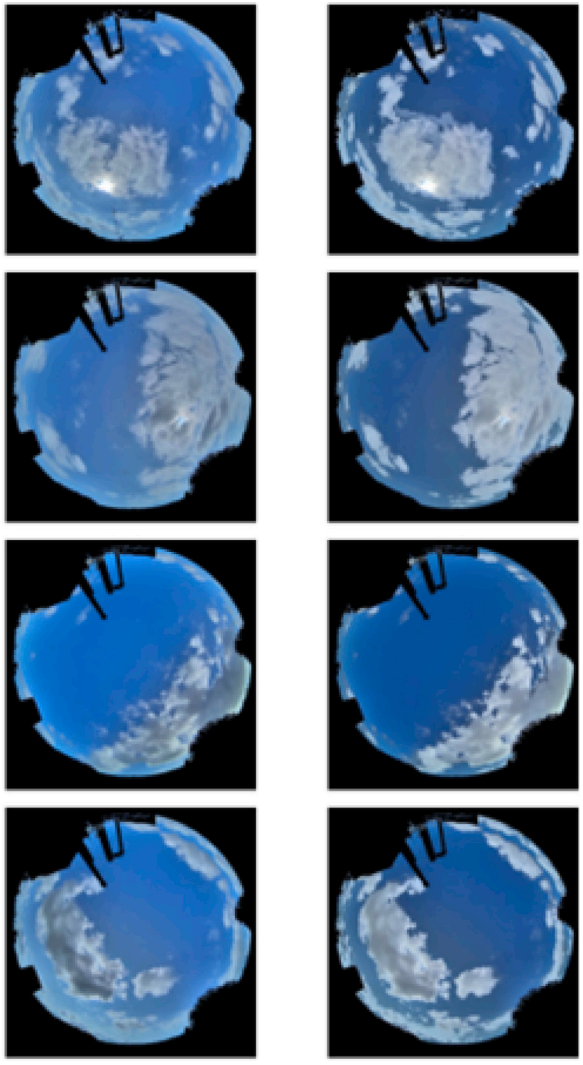


**Fig. A.5.** Random examples of Class 5 cloud identification. The images on the left column are the original images. On the right, the clearer areas identify "Cloud" pixels while the darker are "Not cloud" pixels. Note that the horizon has a mask of black pixels.

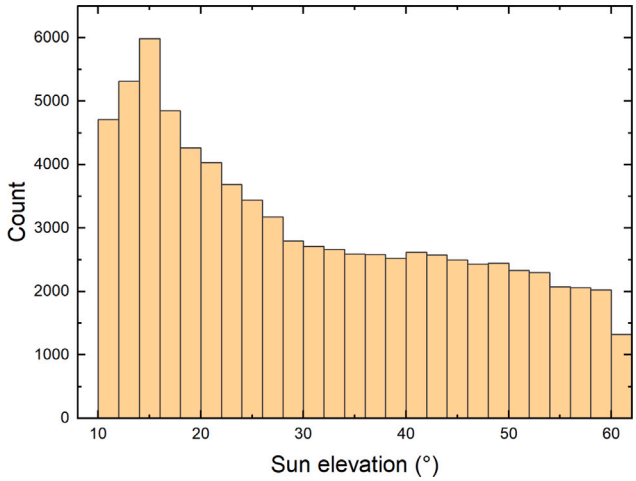



**Fig. B.1.** Distribution of the dataset to the sun's elevation above the horizon.

## References

[1] G. Pfister, R.L. McKenzie, J. Liley, A. Thomas, B.W. Forgan, C.N. Long, Cloud coverage based on all-sky imaging and its impact on surface solar irradiance, J. Appl. Meteorol. Climatol. 42 (2003) 1421–1434.

[2] C.A. Gueymard, Cloud and albedo enhancement impacts on solar irradiance using high-frequency measurements from thermopile and photodiode radiometers. Part 1: impacts on global horizontal irradiance, Sol. Energy 153 (2017) 755–765.

[3] Z.K. Pecenak, F.A. Mejia, B. Kurtz, A. Evan, J. Keissl, Simulating irradiance enhancement dependence on cloud optical depth and solar zenith angle, Sol. Energy 136 (2016) 675–681.

[4] L.R. do Nascimento, T. de Souza Viana, R.A. Campos, R. Rüther, Extreme solar overirradiance events: occurrence and impacts on utility-scale photovoltaic power plants in Brazil, Sol. Energy 186 (2019) 370–381.

[5] F.P. Kreuwel, W.H. Knap, L.R. Visser, W.G. van Sark, J. Vilà-Guerau de Arellano, C.C. van Heerwaarden, Analysis of high frequency photovoltaic solar energy fluctuations, Sol. Energy 206 (2020) 381–389.

[6] A. Mills, M. Ahlstrom, M. Brower, A. Ellis, R. George, T. Hoff, B. Kroposki, C. Lenox, N. Miller, M. Milligan, J. Stein, Y.-h. Wan, Dark shadows, IEEE Power Energy Mag. 9 (3) (2011) 33–41, http://dx.doi.org/10.1109/MPE.2011.940575.

[7] V.A. Martinez Lopez, U. Žindžiūtė, H. Ziar, M. Zeman, O. Isabella, Study on the effect of irradiance variability on the efficiency of the perturb-and-observe maximum power point tracking algorithm, Energies 15 (20) (2022) 7562, http://dx.doi.org/10.3390/en15207562.

[8] A. Gonzalez-Moreno, J. Marcos, I. de la Parra, L. Marroyo, A PV ramp-rate control strategy to extend battery lifespan using forecasting, Appl. Energy 323 (2022) 119546, http://dx.doi.org/10.1016/j.apenergy.2022.119546.

[9] G. Notton, C. Voyant, Chapter 3 - forecasting of intermittent solar energy resource, in: I. Yahyaoui (Ed.), Advances in Renewable Energies and Power

Technologies, Elsevier, 2018, pp. 77–114, http://dx.doi.org/10.1016/B978-0-12-812959-3.00003-4.
[10] N. Krishnan, K.R. Kumar, C.S. Inda, How solar radiation forecasting impacts the utilization of solar energy: A critical review, J. Clean. Prod. 388 (2023) 135860, http://dx.doi.org/10.1016/j.jclepro.2023.135860.
[11] M. Hasenbalg, P. Kuhn, S. Wilbert, B. Nouri, A. Kazantzidis, Benchmarking of six cloud segmentation algorithms for ground-based all-sky imagers, Sol. Energy 201 (2020) 596–614.
[12] A. Heinle, A. Macke, A. Srivastav, Automatic cloud classification of whole sky images, Atmos. Meas. Tech. 3 (3) (2010) 557–567, http://dx.doi.org/10.5194/amt-3-557-2010.
[13] S. Quesada-Ruiz, Y. Chu, J. Tovar-Pescador, H. Pedro, C. Coimbra, Cloud-tracking methodology for intra-hour DNI forecasting, Sol. Energy 102 (2014) 267–275.
[14] J.E. Shields, M.E. Karr, R.W. Johnson, A.R. Burden, Day/night whole sky imagers for 24-h cloud and sky assessment: history and overview, Appl. Opt. 52 (2013) 1605–1616.
[15] M. Caldas, R. Alonso-Suárez, Very short-term solar irradiance forecast using all-sky imaging and real-time irradiance measurements, Renew. Energy 143 (2019) 1643–1658, http://dx.doi.org/10.1016/j.renene.2019.05.069.
[16] C.-L. Fu, H.-Y. Cheng, Predicting solar irradiance with all-sky image features via regression, Sol. Energy 97 (2013) 537–550, http://dx.doi.org/10.1016/j.solener.2013.09.016.
[17] H.-Y. Cheng, C.-C. Yu, Multi-model solar irradiance prediciton based on automatic cloud classification, Energy 91 (2015) 579–587, http://dx.doi.org/10.1016/j.energy.2015.08.075.
[18] Y. Chu, H.T. Pedro, C.F. Coimbra, Hybrid intra-hour DNI forecasts with sky image processing enhanced by stochastic learning, Sol. Energy 98 (2013) 592–603, http://dx.doi.org/10.1016/j.solener.2013.10.020.
[19] A.H. Eşlik, E. Akarslan, F.O. Hocaoğlu, Short-term solar radiation forecasting with a novel image processing-based deep learning approach, Renew. Energy 200 (2022) 1490–1505, http://dx.doi.org/10.1016/j.renene.2022.10.063.
[20] I. Vasilev, Python Deep Learning: Exploring Deep Learning Techniques and Neural Network Architectures with PyTorch, Keras and TensorFlow, Packt Publishing, Birmingham, UK, 2019.
[21] Y. Sun, G. Szűcs, A.R. Brandt, Solar PV output prediction from video streams using convolutional neural networks, Energy Environ. Sci. 11 (7) (2018) 1811–1818, http://dx.doi.org/10.1039/C7EE03420B.
[22] Y. Sun, V. Venugopal, A.R. Brandt, Short-term solar power forecast with deep learning: Exploring optimal input and output configuration, Sol. Energy 188 (2019) 730–741, http://dx.doi.org/10.1016/j.solener.2019.06.041.
[23] C. Feng, J. Zhang, SolarNet: A sky image-based deep convolutional neural network for intra-hour solar forecasting, Sol. Energy 204 (2020) 71–78, http://dx.doi.org/10.1016/j.solener.2020.03.083.
[24] Q. Paletta, A. Hu, G. Arbod, J. Lasenby, ECLIPSE: Envisioning cloud induced perturbations in solar energy, Appl. Energy 326 (2022) 119924, http://dx.doi.org/10.1016/j.apenergy.2022.119924.
[25] Z. Zhen, J. Liu, Z. Zhang, F. Wang, H. Chai, Y. Yu, X. Lu, T. Wang, Y. Lin, Deep learning based surface irradiance mapping model for solar PV power forecasting using sky image, IEEE Trans. Ind. Appl. 56 (4) (2020) 3385–3396, http://dx.doi.org/10.1109/TIA.2020.2984617.
[26] J. Zhang, R. Verschae, S. Nobuhara, J.F. Lalonde, Deep photovoltaic nowcasting, Sol. Energy 176 (2018) 267–276, http://dx.doi.org/10.1016/j.solener.2018.10.024.
[27] Q. Paletta, G. Arbod, J. Lasenby, Benchmarking of deep learning irradiance forecasting models from sky images -An in -depth analysis, Sol. Energy 224 (2021) 855–867, http://dx.doi.org/10.1016/j.solener.2021.05.056.
[28] S.-A. Logothetis, V. Salamalikis, S. Wilbert, J. Remund, L.F. Zarzalejo, Y. Xie, B. Nouri, E. Ntavelis, J. Nou, N. Hendrikx, L. Visser, M. Sengupta, M. Pó, R. Chauvin, S. Grieu, W. van Sark, A. Kazantzidis, Benchmarking of solar irradiance nowcast performance derived from all-sky imagers, Renew. Energy 199 (2022) 246–261.
[29] S.-A. Logothetis, V. Salamalikis, B. Nouri, J. Remund, L.F. Zarzalejo, Y. Xie, S. Wilbert, E. Ntavelis, J. Nou, N. Hendrikx, L. Visser, M. Sengupta, M. Pó, R. Chauvin, S. Grieu, N. Blum, W.v. Sark, A. Kazantzidis, Solar irradiance ramp forecasting based on all-sky imagers, Energies 15 (17) (2022) 6191, http://dx.doi.org/10.3390/en15176191.
[30] W. Kong, Y. Jia, Z.Y. Dong, K. Meng, S. Chai, Hybrid approaches based on deep whole-sky-image learning to photovoltaic generation forecasting, Appl. Energy 280 (2020) 115875, http://dx.doi.org/10.1016/j.apenergy.2020.115875.
[31] C. Feng, J. Zhang, W. Zhang, B.-M. Hodge, Convolutional neural networks for intra-hour solar forecasting based on sky image sequences, Appl. Energy 310 (2022) 118438, http://dx.doi.org/10.1016/j.apenergy.2021.118438.
[32] H. Wen, Y. Du, X. Chen, E. Lim, H. Wen, L. Jiang, W. Xiang, Deep learning based multistep solar forecasting for PV ramp-rate control using sky images, IEEE Trans. Ind. Inform. 17 (2) (2021) 1397–1406, http://dx.doi.org/10.1109/TII.2020.2987916.
[33] S. Shan, C. Li, Z. Ding, Y. Wang, K. Zhang, H. Wei, Ensemble learning based multi-modal intra-hour irradiance forecasting, Energy Convers. Manage. 270 (2022) 116206, http://dx.doi.org/10.1016/j.enconman.2022.116206.
[34] P. Manandhar, M. Temimi, Z. Aung, Short-term solar radiation forecast using total sky imager via transfer learning, Energy Rep. 9 (2023) 819–828, http://dx.doi.org/10.1016/j.egyr.2022.11.087.
[35] R. Gonzalez, R. Woods, Digital Image Processing, Global Edition, Pearson Education, Limited, 2017.
[36] C.-W. Kok, W.-S. Tam, Digital Image Interpolation in Matlab, Wiley-IEEE Press, 2019.
[37] D. Slater, C. Long, T. Tooman, Total Sky Imager/Whole Sky Imager cloud fraction comparison, in: Eleventh ARM Science Team Meeting Proceedings, Atlanta, USA, 2001.
[38] Q. Li, W. Lu, J. Yang, A hybrid thresholding algorithm for cloud detection on ground-based color images, J. Atmos. Ocean. Technol. 2 (2011) 1286–1296.
[39] H.-M. Zuo, J. Qiu, Y.-H. Jia, Q. Wang, F.-F. Li, Ten-minute prediction of solar irradiance based on cloud detection and a long short-term memory (LSTM) model, Energy Rep. 8 (2022) 5146–5157.
[40] A. Niccolai, A. Nespoli, Sun position identification in sky images for nowcasting application, Forecasting 2 (4) (2020) 488–504, http://dx.doi.org/10.3390/forecast2040026.
[41] S. Liu, L. Zhang, Z. Zhang, C. Wang, B. Xiao, Automatic cloud detection for All-sky images using superpixel segmentation, IEEE Geosci. Remote Sens. Lett. 12 (2) (2015) 354–358.
[42] Ren, Malik, Learning a classification model for segmentation, in: Proceedings Ninth IEEE International Conference on Computer Vision, Vol. 1, 2003, pp. 10–17, http://dx.doi.org/10.1109/ICCV.2003.1238308.
[43] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66, http://dx.doi.org/10.1109/TSMC.1979.4310076.
[44] J. Gibson, O. Marques, Optical flow fundamentals, in: Optical Flow and Trajectory Estimation Methods, Springer International Publishing, Cham, 2016, pp. 1–7, http://dx.doi.org/10.1007/978-3-319-44941-8_1.
[45] G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in: J. Bigun, T. Gustavsson (Eds.), Image Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 363–370.
[46] OpenCV, Optical flow [online].
[47] Y. Karout, S. Thil, J. Eynard, E. Guillot, S. Grieu, Hybrid intrahour DNI forecast model based on DNI measurements and sky-imaging data, Sol. Energy 249 (2023) 541–558, http://dx.doi.org/10.1016/j.solener.2022.11.032.
[48] A. Woyte, R. Belmans, J. Nijs, Analysing short-time irradiance fluctuations by their characteristic time scales, in: Proceedings of the 3rd World Conference on Photovoltaic Energy Conversion, 2003, Vol. 3, Osaka, Japan, 2003, pp. 2290–2293.
[49] W.F. Holmgren, C.W. Hansen, M.A. Mikofski, Pvlib python: a python package for modeling solar energy systems, J. Open Source Softw. 3 (2018) 884, http://dx.doi.org/10.21105/joss.00884.
[50] H. Wen, Y. Du, X. Chen, E. Lim, H. Wen, L. Jiang, W. Xiang, Deep learning based multistep solar forecasting for PV ramp-rate control using sky images, IEEE Trans. Ind. Inform. 17 (2) (2021) 1397–1406, http://dx.doi.org/10.1109/TII.2020.2987916.
[51] X. Chen, Y. Du, H. Wen, Forecasting based power ramp-rate control for PV systems without energy storage, in: 2017 IEEE 3rd International Future Energy Electronics Conference and ECCE Asia (IFEEC 2017 - ECCE Asia), 2017, pp. 733–738, http://dx.doi.org/10.1109/IFEEC.2017.7992130.
[52] X. Chen, Y. Du, H. Wen, L. Jiang, W. Xiao, Forecasting-based power ramp-rate control strategies for utility-scale PV systems, IEEE Trans. Ind. Electron. 66 (3) (2019) 1862–1871, http://dx.doi.org/10.1109/TIE.2018.2840490.
[53] A. Gonzalez-Moreno, J. Marcos, I. de la Parra, L. Marroyo, A PV ramp-rate control strategy to extend battery lifespan using forecasting, Appl. Energy 323 (2022) 119546, http://dx.doi.org/10.1016/j.apenergy.2022.119546.