SAND2021-7209C

Solar Transposition Modeling via Deep Neural Networks with Sky Images

Benjamin G. Pierce*, Jennifer L. Braid*, Joshua S. Stein*, Jim Augustyn†, and Daniel Riley*
* Sandia National Laboratories, Albuquerque, New Mexico, 87123, USA
†Augustyn & Company, Berkeley, California, 94707, USA

***Abstract*—This paper presents a notable advance toward development of a new method of increasing single-axis tracking PV system power output by improving determination and near-term prediction of optimum module tilt angle. The tilt angle of the plane receiving the greatest total irradiance changes with sun position and atmospheric conditions including cloud formation and movement, aerosols and particulate loading, as well as varying albedo within a module's field of view. In this work, we present a multi-input convolutional neural network that can create a profile of plane-of-array irradiance versus surface tilt angle over a full 180 degree arc from horizon to horizon. As input, the neural network uses calculated solar position and clear-sky irradiance values, along with sky images. The target irradiance values are provided by the Multi-Planar Irradiance Sensor (MPIS). In order to account for varying irradiance conditions, the MPIS signal is normalized by the theoretical clear-sky global horizontal irradiance (GHI). Using this information, the neural network outputs a N-dimensional vector, where N is the number of points to approximate the MPIS curve via Fourier resampling. The output vector of the model is smoothed with a Gaussian kernel to account for error in the downsamping and subsequent upsampling steps, as well as to smooth the unconstrained output of the model. These profiles may be used to perform near-term prediction of angular irradiance, which can then inform the movement of a PV tracker.**

***Index Terms*—irradiance, deep learning, neural network, photovoltaics**

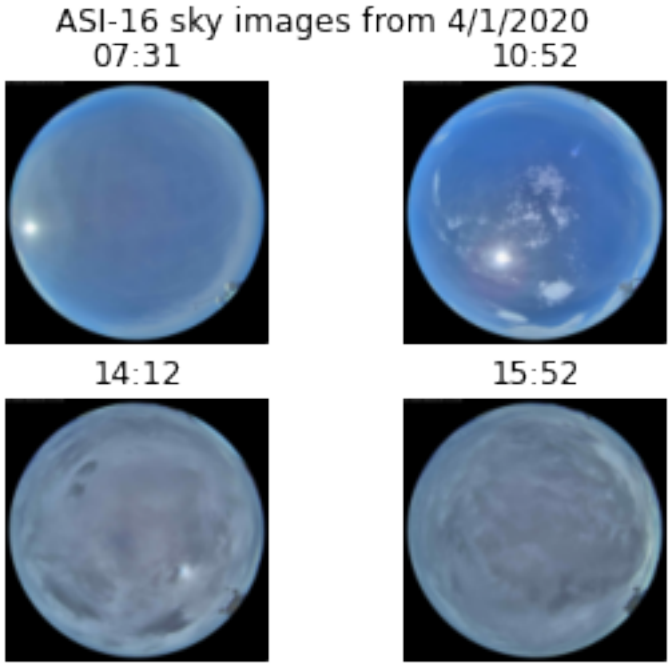

Fig. 1. ASI-16 sky images taken over a single day, 4/1/2020.

I. Introduction

For any photovoltaic (PV) system, it is necessary to profile the site in order to have a working knowledge of the solar insolation available for various system configurations. It is known that solar insolation is dependent on the angular position of the plane of measurement [1]. The solar resource profile is important for both fixed-tilt and tracking photovoltaic systems, as it informs the optimal position of the modules; the more irradiance the module receives, the greater the power it generates. On clear-sky days, the irradiance versus tilt angle profile is easy to predict from the sun's position. However, on partially or fully overcast days, the irradiance profile can be severely distorted due to the obstruction and scattering of light by clouds. The increase in diffuse light and reduction in direct irradiance leads to a relative reduction in power conversion efficiency for PV systems that continue to track the sun's position throughout these conditions. Our calculations indicate that optimal tracking of the angle of maximum irradiance (as opposed to solar position) can increase the power production of single-axis tracking PV systems by up to 2% per year, without modification of the tracking mechanism. Here, we present prediction of angular irradiance profiles from sky images. The value of this approach is that the angular irradiance can be informed in real time by local, observable weather conditions. Therefore the use of sky images provides distinct advantages over multi-angle irradiance sensors: 1) the sky camera involves no moving parts and a single sensor to collect data, and 2) sky images enable forecasting of future weather conditions, which is not possible with real-time irradiance sensors.

A. Dataset

The dataset for this project consists of output from various measurement devices, located in Albuquerque, New Mexico. The first device is the ASI-16 sky camera. This camera produces fisheye-style images of the sky, and is mounted at a fixed point facing upwards. A collection of example sky

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

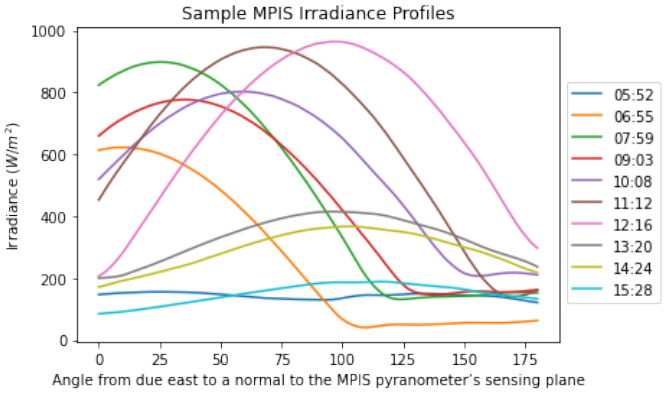

Fig. 2. Sample MPIS profiles for different times on the same day. This day is clear in the morning and progressively becomes more cloudy, which can be seen in a decrease in peak insolation and the maximum of the curve shifting towards the normal (180°).

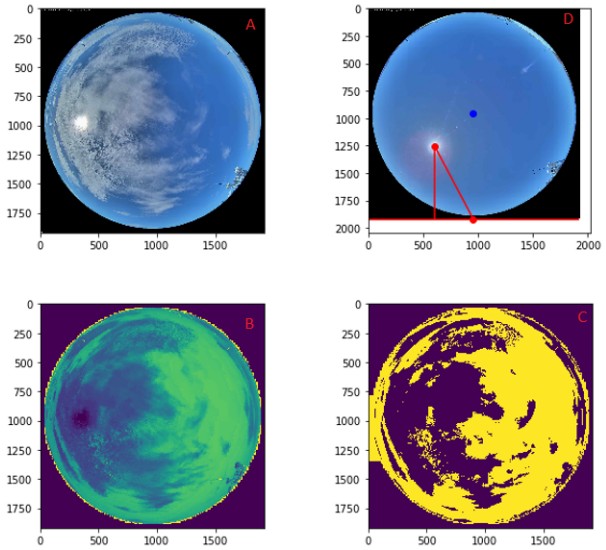

Fig. 3. Our classical image processing pipeline. A) the original image, B), an image transformed to hue-saturation-value color space, C), an adaptive threshold to extract clouds and D) location of the the sun via maximum pixel intensity. Image D shows the sun location for a different original image than the others because this method is prone to error in cloudy conditions, especially when there is greater cloud cover

images from the ASI-16 sky camera, taken 4/1/20, is shown in Fig. 1. Additionally, there are numerous pyranometers and other weather sensors at the site, these can be used to verify the normalization procedure.

The sensor that enables this study is the Multi-Planar Irradiance Sensor (MPIS) developed by Augustyn & Company, Berkeley, California [2]. This sensor takes single-axis sweeps of the sky from the eastern horizon to the western horizon, and records the irradiance profiles. This profiling provides direct measurement of the angle of maximum Plane-of-Array (POA) irradiance, and allows comparison of solar insolation for different tracker positions for the same time. Fig. 2 shows multiple MPIS curves on different times during the same day as the sample of sky images shown in Fig. 1.

However, installing and operating MPIS units at individual solar plants will add cost and complexity to plant operation. While it is expected that ongoing improvement of the MPIS design will result in cost reductions, the sky camera model developed here also offers the potential of near term forcasting of irradiance vs. tilt profiles as tracker control system feedback, which the MPIS alone cannot do. In this work, we use MPIS data as ground truth to build solar transposition models from sky images at the same location. It is hoped that co-location of MPIS and sky camera units at a limited number of sites may be enough to train the model for more universal application. Therefore we use this data as ground-truth to build solar transposition models from sky images at the same location, to simplify the data collection process for determining the angle of maximum POA irradiance. Although conventional sky imagers like the ASI-16 are quite costly as well, we find that a less expensive option would be adequate, as experimental study has shown that image resolutions as low as 128x128 pixels are viable inputs to the model. In fact, downscaling the image to a lower resolution both decreases model training time and improves convergence; this effect is common to convolutional neural networks applied to very high resolution data. It is hypothesized that downscaling decreases overfitting as the linear interpolation smoothes out camera noise. As we are only concerned with the larger features (such as cloud cover) and not detail, there is no significant advantage to excessively high-resolution sky images.

As seen in the sky images in Fig. 1, the day shown begins with some scattered clouds and progresses into heavy cloud coverage. The corresponding MPIS curves in Fig. 2, show the effects of this cloud coverage, such as reduced irradiance and a trend towards horizontal in the afternoon. During clear-sky conditions, MPIS curves in the afternoon achieve maximum irradiance facing West (above 90°).

B. Exploratory Data Analysis

The concept of extracting weather data from sky images has been explored previously. Most approaches, such as Alonso *et al.* [3] and Long *et al.* [4] use classical image processing approaches to extract pertinent information, such as cloud coverage and movements. These approaches focus on applying multiple image filters and other transformations to individual images, as opposed to a statistical learning technique like deep neural networks. Inspired by these works, we conducted exploratory data analysis on our images.

As shown in Figure 3, quite a bit of information can be extracted from these images. The first area of focus is identifying clouds in the images. This is a concern because the presence of clouds in the sky will result in a different irradiance profile from a clear sky. In the literature, cloud area is generally extracted via a simple threshold; however, we found this approach to be error-prone in RGB color-space. By switching to hue-saturation-value (HSV) color space, we were

able to get a clear delineation between cloud and sky using an adaptive intensity threshold operation for segmentation. Although this method is simple, it can be prone to error due to sub-optimal parameter choices. Additionally, since the intensity of the images may change due to automatic adjustments in the camera, it is difficult to choose a global threshold for the entire dataset.

Another metric of importance is the location of the Sun. As in [5], we locate the Sun in an image by thresholding the red channel of the image; in a similar manner, we locate the center of the sky by thresholding the blue channel. As noted by Savoy *et al.*, if cloud cover prevents visibility of the Sun, this method becomes very error prone. However, image detection of the sun's position is unnecessary due to its ease of calculation from the date/time and location of the camera.

As the central motivation of this work is to observe the effects of cloud cover on irradiance profiles, classical image processing methods alone are inadequate due to the limitations listed above. Therefore we have developed a neural network-based method for predicting angular profiles of POA irradiance from sky images.

II. Methodology

In this work, we present a multi-input convolutional neural network, shown in Figure 4, to predict irradiance profiles. Multi-input neural networks have begun to gain traction in other fields, such as identifying flowers from multiple different angles [6]. This is a viable approach for sky images because they contain a large amount of qualitative information about the current status of the sky. Additionally, the multi-input structure allows us to utilize additional information that is relevant to the angular irradiance profile, such as calculated solar position and clear-sky irradiance.

Convolutional neural networks are a method where a small filter (also called a kernel) slides (convolves) over an image; the output of this operation is a matrix where each entry is the dot product of the filter and each set of points from the image. This operation is expressed mathematically in Eq. 1.

$$(I * K)_{i,j} = \sum_{m=1}^{k} \sum_{n=1}^{k} K_{m,n} I_{i+m-1,j+n-1} + b \qquad (1)$$

In this equation, I is the image (a matrix of shape (i, j)), b is a bias term, and K is the kernel. In our neural network, the kernel is usually of size (3,3) or (5,5). The weights of the kernel (the numbers inside the matrix) are learned via back-propagation like the weights and biases of a fully connected layer.

The second input to the neural network is the position of the sun relative to the sky camera, given in a spherical coordinate system. The position of the sun in the sky can be calculated for a given time of day by a variety of methods. In this work, we use the solar position method presented by Reda and Andreas of NREL [7], implemented in PVlib-python [8]. This calculation provides the location of the sun in terms of zenith and azimuth, which is fed into the neural network in a multilayer perceptron-style sub-network. This sub-network can be considered a parallel branch of the whole that is responsible for its own specialized computation. We hypothesize that the neural network uses this information to map the spherical coordinates onto the image, thereby locating the sun irrespective of cloud cover. In order to do so, the two sub-networks (the CNN and the multilayer perceptron) are subsequently joined. This concept was also developed independently by Paletta *et al.* [9] in 2020, who used a similar data augmentation process for irradiance prediction from sky images, but opted for additional ResNet-like residual connections we found to be unnecessary. Additionally, Paletta *et al.* directly predict irradiance, which is not useful for our intended application, and is easily measured via conventional instruments that do not provide information on angular irradiance.

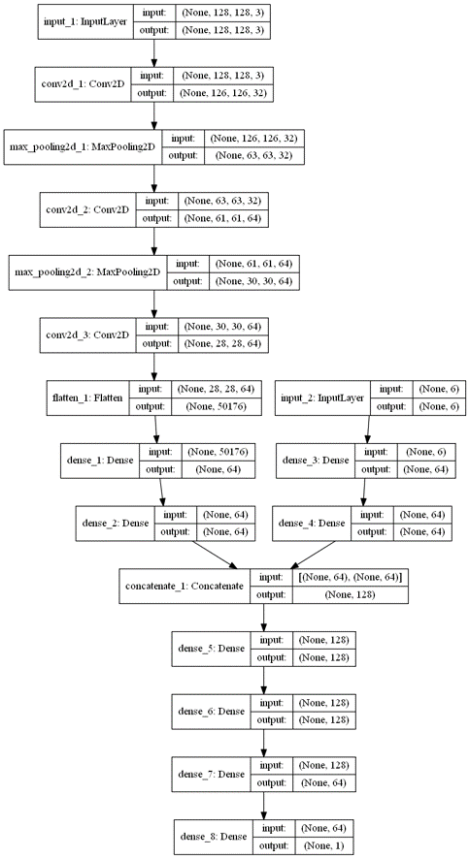

Fig. 4. The structure of our multi-input CNN. The number of neurons in the output layer are a function of the resampling procedure; for a signal resampled to $N = 50$ points, the final layer will have 50 neurons. Dropout and batch normalization layers, omitted from this diagram, are included as well: batch normalization following convolution layers and dropout after every other dense layer.

In the final combination step, dense (fully-connected) outputs of the two sub-networks are concatenated and fed into a final sequence of dense layers, the output of which is a regression vector of shape N, where N is the number of points to approximate angles of irradiance. The MPIS sensor samples 360 points over the 180 degree range from horizon

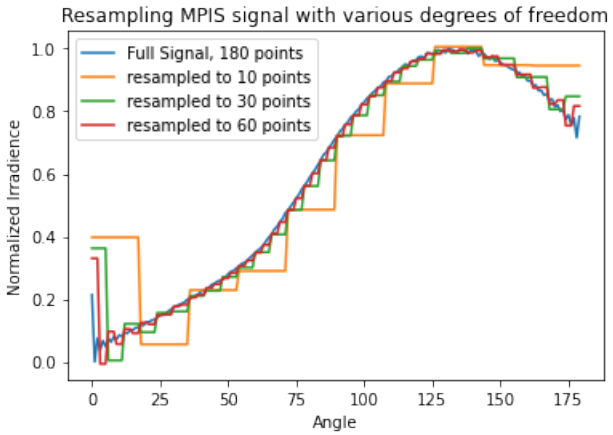

Fig. 5. Resampling MPIS signal via Fourier method

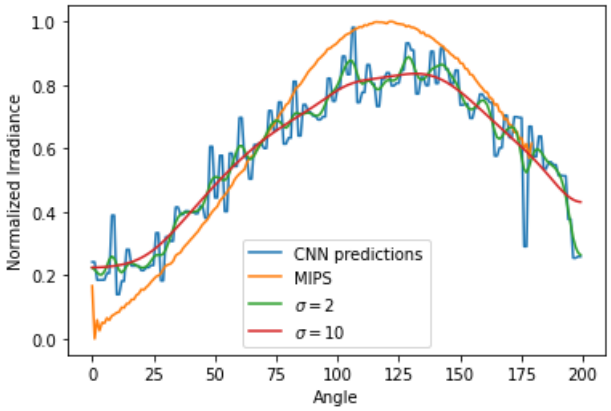

Fig. 6. Smoothing the model output using a Gaussian kernel

to horizon; that is, it samples at half-angles. However, that level of specificity is difficult to predict because this is a high-dimensional regression problem over the real numbers, which is a continuous search space. The problem is then much more difficult numerically than a classification problem as there are not a discrete number of possible states. It is therefore useful to approximate the curve with fewer points. Experimentally, we have found that $N = 90$ is optimal; a finer regression results in a model that is more difficult to train and prone to overfitting. Thus, the target of the model equals the number of output neurons equals N, as the MPIS curves are pre-processed prior to passing them to the model.

The points in the training data are resampled using the Fourier method, implemented in the scipy package, and the model is then trained on the downsampled data, resulting in an output of the same dimension. The Fourier downsampling method simply transforms the data into the frequency domain and removes the second and third groups of elements, which are the half with the highest frequency components As the data are real numbers, the FFT results in mirrored complex conjugates in the lower half of the transformed space. Thus, the middle 50% is removed to preserve this symmetry while downsampling. This then removes the samples with the highest frequencies. The results of this process are shown in Figure 5. In order to compare the downsampled output and the true value, the model output is padded via equal repetition of elements, or Fourier upsampling. However, this series of transforms and the additional error in the predicted curves can sometimes cause a jagged model output, whereas real anglular irradiance profiles are smooth. Gaussian smoothing, a common technique in image and signal processing, can be applied to smooth the curve, assuming Gaussian error. This method is calibrated by the σ parameter of the underlying distribution, which specifies a higher or lower degree of smoothness. This process can be seen in Figure 6.

Training this model requires a slight modification of gradient descent. This is because the model is split into two sub-networks; typically, the model is sequential and the gradient flows directly backwards, but here the model forks into two branches. In this case, there are N outputs and 2 inputs. Thus, the rule for gradient flow can be described using the Chain Rule in Eq. 2

$$\frac{\partial \mathcal{L}}{\partial w_i} = \sum_{n=1}^{N} \frac{\partial \mathcal{L}}{\partial y_n} \frac{\partial y_n}{\partial w_i} \tag{2}$$

where $\mathcal{L}$ is the loss function, w_i is a weight and y_n is an output. The loss function is mean squared error of the samples Y_i:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{3}$$

In this manner, contribution to the overall loss (Equation 3) can be calculated for each weight. As the loss propagates through the network, both sub-networks are updated according to the overall loss. Since the model outputs a vector, the total loss is just the sum of the loss of the individual elements.

The normalization process constrains the value of each element to be between zero and some constant ϵ, where ϵ depends on the specific normalization procedure used. To reflect this constraint, a custom activation function for the output is used. Usually, no activation is provided for a regression output. However, since the range of values is known *a priori*, we can give the network a hint by applying the activation function in Eq. 4

$$\epsilon\text{-}ReLU(x) = min(ReLU(x), \epsilon) \tag{4}$$

where ϵ is set to the maximal acceptable value and $ReLU(x)$ is the standard Rectified Linear Unit activation function. For example, if the data is normalized between 0 and 1, setting $\epsilon = 1$ will prevent the model from overshooting without affecting the true values. Thus, the activation function is linear over the range $[0, \epsilon]$, unlike the nonlinear sigmoid, which may cause outputs to trend towards the extremes. Experimentally, the addition of this activation function allowed the model to converge faster than without. Additionally, if the model encounters a severe outlier or other anomaly when predicting a new sample, then the amount of possible error can be limited.

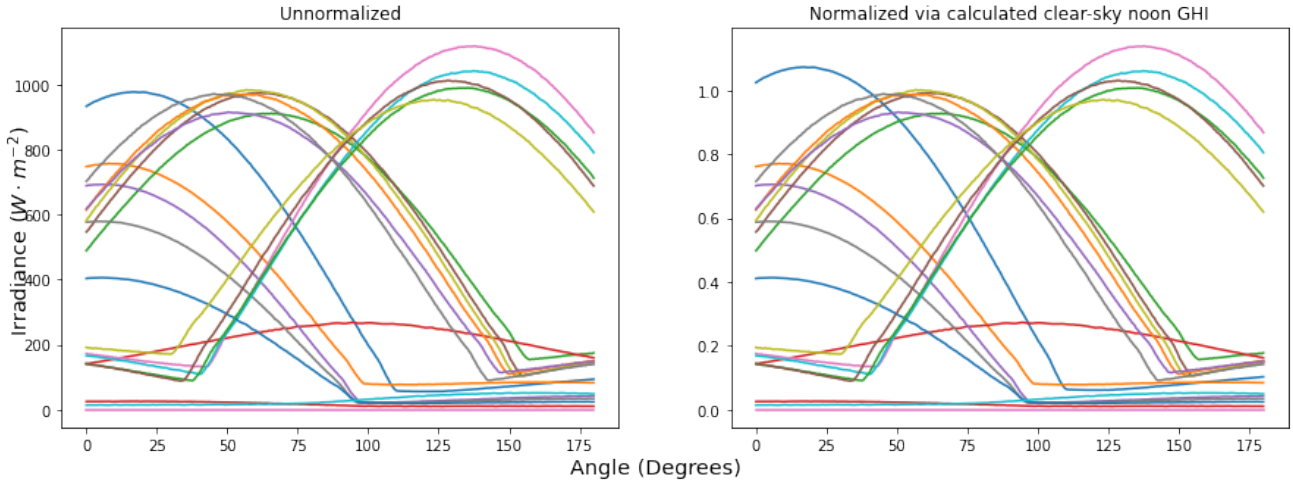

Fig. 7. A) Raw MPIS curves, and B) normalized MPIS curves that retain relative curve scaling

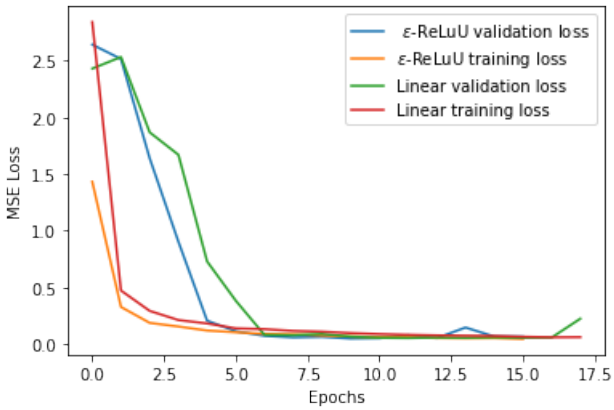

Fig. 8. Loss curves for two separate activation functions. The limited ϵ-ReLU appears to add some stability to the training process.

Training curves for both the ϵ-ReLU and linear activation functions can be seen in Figure 8.

Experimentally, it was found the ϵ-ReLU activation produced smoother output curves; although the exact cause is uncertain, the current hypothesis is that it prevents overfitting by working in conjunction with early stopping.

A particular issue of this model is that of normalization. The MPIS signal is a function of raw irradiance values, which are very difficult to predict from only a sky image as there is no direct correlation between the image and the raw irradiance values. Thus, the model would likely attempt to memorize an irradiance/date/time mapping, which is difficult to learn and very site-specific. Therefore, the MPIS signal should be normalized or transformed to mitigate these problems. One simple metric is to squeeze all values between zero and one via min-max scaling. However, this is a lossy metric; for example, solar noon and diffuse conditions can have MPIS curves that are fairly similar in shape, but have drastically different magnitudes. Notably, this issue does not impact the correct angle of maximal irradiance; thus, predicting the argmax of the MPIS signal is a significantly easier problem.

A different metric that preserves this magnitude difference is the L2 norm, defined as $||x||_2 \equiv \sqrt{x \cdot x}$. The L2 norm is a common baseline for normalization of an arbitrary vector. Another choice may also be the infinity norm, which is defined as $||x||_\infty = \max_i(x_i)$. This would essentially result in normalizing by GHI on a clear day, giving the normalization process physical meaning.

In fact, the ideal, clear-sky GHI is easily calculable for a given time, latitude, and longitude; this work uses the Ineichen model [10], implemented in PVlib-python. Normalizing by this quantity results in a much smoother curve that preserves relative magnitudes. Precisely, we normalize by the clear sky GHI at solar noon; this results in all curves, regardless of seasonality, to be within the same range of values. The values range in magnitude from zero to about four, as seen in Figure 7. This is because maximum value of the MPIS signal (maximum POA irradiance) on a clear day will always be greater than or equal to the projected clear-sky GHI.

Experimentally, each type of normalization has different drawbacks. The min-max scaling method is simple, and does preserve the correct maximum, but the resulting curve is often much less smooth and lacks the ability to distinguish between curves of the same shape and different magnitude. Clear-sky GHI normalization was found to be the most optimal in this work, but does involve some additional computation and is dependent on correct calculation of clear-sky GHI. Regardless of normalization method, the model accurately produces the argmax of the MPIS signal, which is arguably the most important point, as it represents the optimal real-time angle for a single-axis tracker.

This work uses the Python programming language, and the TensorFlow library with the Keras front-end. The model was implemented using the Keras functional API for automatic differentiation. The Adam optimizer was used for training with mean squared error as the loss function. Gradient clipping

and weight decay were also used to aid the training process and prevent exploding/vanishing gradients. An early stopping callback was used to terminate the training process when the validation loss does not improve for a set duration, as well as restore the best set of weights on the validation set. Batch normalization and dropout layers were used in addition to convolution, pooling, and dense layers. The model was trained with a custom generator function to operate in batch mode. A Tesla V100 GPU with 32GB of VRAM was used to accelerate training on Sandia's deep learning platform, which is also equipped with dual 24 core Intel Xenon Platinum CPUs with 1.5 TB of RAM.

III. Results

The model was trained for 80 epochs, with 25 steps per epoch and 8 samples per step, which is a total 16,000 samples. The hyperparameter N, which controls the number of points to estimate, was set to 90. As N increases, so does the time to model convergence. Experimentally, a higher N results in lower training loss, but higher validation loss, implying that the optimal value lies somewhere in the middle of the range [0,180].

An series of example results can be shown in Figure 9. All samples shown are drawn from a period of about a month; the model does capture seasonality effects, but it is easier to interpret the model without seasonality. The curves very clearly show that the model captures the general shape of the angular irradiance profile, given the sky image and calculated solar position. However, there does remain some error, particularly in magnitude, due to the high dimensionality required as well as the complex up and down scaling required for model convergence. Notably, without both the normalization and downscaling steps (that is, asking the model to predict the MPIS curves directly), the model fails to converge, even if it is supplied the measured GHI, DNI, and DHI. Although the problem is complex, there are actually a small number of possible curves shapes in general. Thus, the model has three "jobs": learn possible shapes, match to current conditions, and scale accordingly; however, as neural networks are "black box" models, it is impossible to confirm if these are truly the concepts learned.

As mentioned above, the model learns shape quite well; as a result, it can identify the angle of maximal irradiance with high accuracy, as shown in Figure 10. A distribution of the absolute error in the irradiance itself at the maximum point is shown in Figure 11. As shown, the model performs exceedingly well, as the vast majority of errors are less then 5 degrees off the true value. This error can be considered basically negligible in when considering tracker movement and precision.

Example error in irradiance for each predicted curve shown in Fig. 9 can be seen in Figure 12.

The error appears to be most significant in the early mornings and late evening. This is likely due to error in upscaling to the true magnitude or an albedo effect. Additionally, Fourier upsampling does cause some of the oscillation at the endpoints as well. However, the model appears to capture the relevant information for a near-term forecasting tracking algorithm quite well.

The caveat to this model is that it is specific to the site at which it was trained; conditions at a different geographic location may or may not have similar angular irradiance profiles and will not have similar magnitudes. The normalization process should remove error due to differing magnitudes, but there may be additional seasonal or geographic effects that the model has not yet seen. In the future, we plan to deploy more MPIS sensors and sky cameras to verify these and other assumptions about the model.

IV. Conclusion

In this work we have demonstrated a convolutional neural network transposition model that can replicate the MPIS instrument signal (angular solar irradiance profiles) with high accuracy using only sky images and calculable quantities, such as the solar position and clear-sky GHI. We also present several modifications of the traditional sequential convolutional neural network pipeline, including multiple input branches, resampling and smoothing techniques, custom activation functions, and other adaptions to this particular problem. Without these modifications, training the network proved to be impossible.

Overall, the model allows easy deployment and profiling of angular solar irradiance in real time. In particular, the model proved to be very accurate at predicting the real-time angle of maximal plane-of-array irradiance, which is the theoretically optimal angle of a single-axis tracker. In the future, this information will be integrated into a single-axis tracking algorithm in order to optimize tracker movements for overcast sky conditions.

References

[1] P. Ineichen, A. Zelenka, O. Guisan, and A. Razafindraibe, "Solar radiation transposition models applied to a plane tracking the sun," *Solar Energy*, vol. 41, no. 4, pp. 371–377, Jan. 1988. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0038092X88900333

[2] J. R. Augustyn, "Method, system, and apparatus for rapidly measuring incident solar irradiance on multiple planes of differing angular orientations," US Patent US10 281 552B2, May, 2019. [Online]. Available: https://patents.google.com/patent/US10281552B2/en?oq=U.S.+Patent+10281552+

[3] J. Alonso and F. J. Batlles, "Short and medium-term cloudiness forecasting using remote sensing techniques and sky camera imagery," vol. 73, pp. 890–897. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544214008007

[4] C. N. Long, J. M. Sabburg, J. Calbó, and D. Pagès, "Retrieving cloud characteristics from ground-based daytime color all-sky images," vol. 23, no. 5, pp. 633–652, publisher: American Meteorological Society. [Online]. Available: https://journals.ametsoc.org/jtech/article/23/5/633/2718/Retrieving-Cloud-Characteristics-from-Ground-Based

[5] F. M. Savoy, S. Dev, Y. H. Lee, and S. Winkler, "Geo-referencing and stereo calibration of ground-based whole sky imagers using the sun trajectory," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 7473–7476, ISSN: 2153-7003.

[6] Y. Sun, L. Zhu, G. Wang, and F. Zhao, "Multi-input convolutional neural network for flower grading," vol. 2017, pp. 1–8.

[7] I. Reda and A. Andreas, "Solar position algorithm for solar radiation applications," vol. 76, no. 5, pp. 577–589. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X0300450X

[8] W. F. Holmgren, C. W. Hansen, and M. A. Mikofski, "pvlib python: a python package for modeling solar energy systems," vol. 3, no. 29, p. 884. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.00884

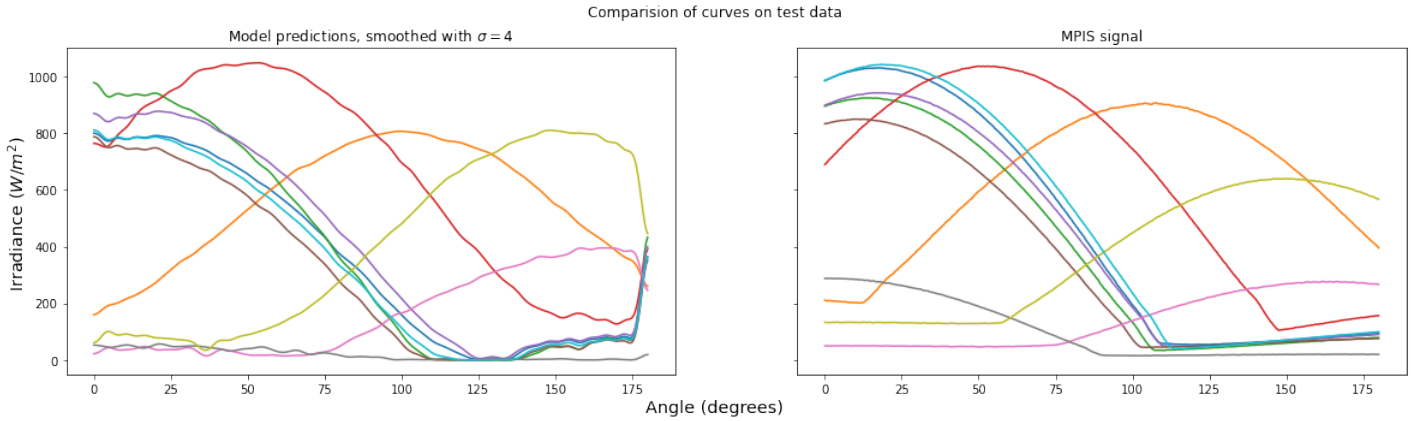

Fig. 9. A) 10 randomly selected angular irradiance profile predictions for a variety of sky conditions. In the orange is a typical near solar noon with some light clouds. The red is a cloudy morning, whereas the light blue is a bright, clear morning. B) The corresponding MPIS curves for the selected profile predictions.

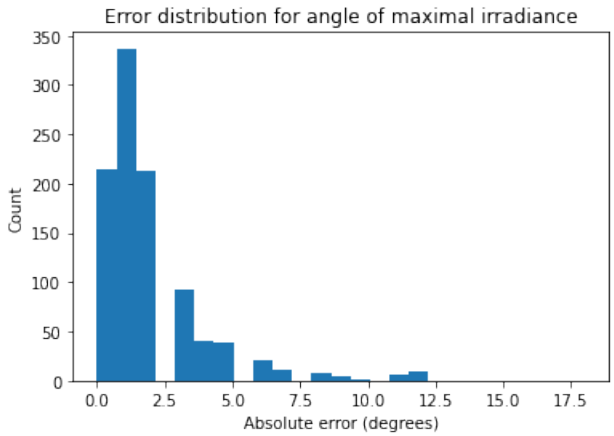

Fig. 10. Histogram of absolute error in predicting the angle of maximum irradiance for 1000 test images.

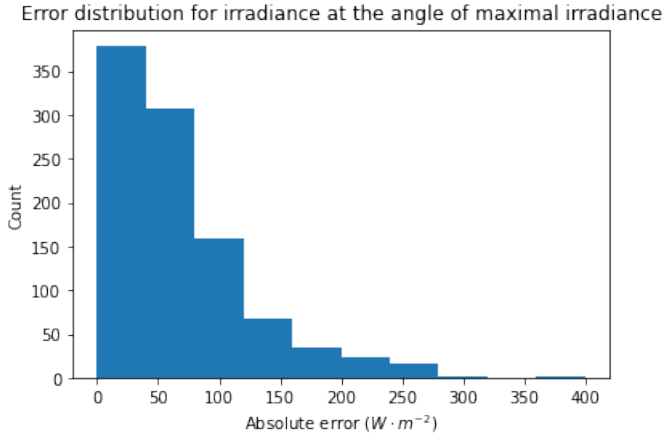

Fig. 11. Histogram of absolute error in predicting the irradiance ath the angle of maximum irradiance.

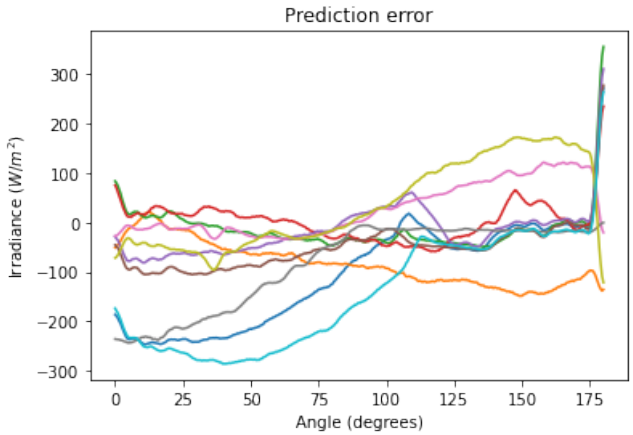

Fig. 12. Error for the same randomly selected test points

[9] Q. Paletta and J. Lasenby, "Convolutional Neural Networks applied to sky images for short-term solar irradiance forecasting," *arXiv:2005.11246 [cs, eess]*, May 2020, arXiv: 2005.11246. [Online]. Available: http://arxiv.org/abs/2005.11246

[10] P. Ineichen and R. Perez, "A new airmass independent formulation for the Linke turbidity coefficient," *Solar Energy*, vol. 73, no. 3, pp. 151–157, Sep. 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X02000452