

## Article

# Deep Learning-Based Image Regression for Short-Term Solar Irradiance Forecasting on the Edge

Elissaios Alexios Papatheofanous <sup>1</sup> , Vasileios Kalekis <sup>1,2</sup>, Georgios Venitourakis <sup>1,2</sup>, Filippos Tziolos <sup>1</sup> and Dionysios Reisis <sup>1,2,\*</sup>

<sup>1</sup> Electronics Lab, Department of Physics, National and Kapodistrian University of Athens, 15784 Athens, Greece

<sup>2</sup> Institute of Accelerating Systems and Applications, 15784 Athens, Greece

\* Correspondence: dreisis@phys.uoa.gr

**Abstract:** Photovoltaic (PV) power production is characterized by high variability due to short-term meteorological effects such as cloud movements. These effects have a significant impact on the incident solar irradiance in PV parks. In order to control PV park performance, researchers have focused on Computer Vision and Deep Learning approaches to perform short-term irradiance forecasting using sky images. Motivated by the task of improving PV park control, the current work introduces the Image Regression Module, which produces irradiance values from sky images using image processing methods and Convolutional Neural Networks (CNNs). With the objective of enhancing the performance of CNN models on the task of irradiance estimation and forecasting, we propose an image processing method based on sun localization. Our findings show that the proposed method can consistently improve the accuracy of irradiance values produced by all the CNN models of our study, reducing the Root Mean Square Error by up to 10.44 W/m<sup>2</sup> for the MobileNetV2 model. These findings indicate that future applications which utilize CNNs for irradiance forecasting should identify the position of the sun in the image in order to produce more accurate irradiance values. Moreover, the integration of the proposed models on an edge-oriented Field-Programmable Gate Array (FPGA) towards a smart PV park for the real-time control of PV production emphasizes their advantages.

**Keywords:** computer vision; deep learning; convolutional neural networks; edge computing; irradiance forecasting; photovoltaic



**Citation:** Papatheofanous, E.A.; Kalekis, V.; Venitourakis, G.; Tziolos, F.; Reisis, D. Deep Learning-Based Image Regression for Short-Term Solar Irradiance Forecasting on the Edge. *Electronics* **2022**, *11*, 3794. <https://doi.org/10.3390/electronics11223794>

Academic Editor: Gwanggil Jeon

Received: 20 October 2022

Accepted: 15 November 2022

Published: 18 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The ongoing transition from traditional coal and fossil fuels to Renewable Energy Sources (RES) has led to solar photovoltaic (PV) parks having an increased share in the energy production mix of many countries. In order to serve the needs of this energy transition, PV production plants have to be integrated in the utility electrical grids, or even in mini-grid and off-grid systems such as in autonomous islands. Moreover, the development of technologies in the fields of Artificial Intelligence (AI), Computer Vision (CV), and edge computing, combined with the Smart Grid (SG) concept [1], promotes on-site data intelligence in PV parks. In this direction, AI-enabled smart PV parks can contribute towards adapting PV power production to the dynamic requirements of the grid [2].

Currently, attempts to efficiently and reliably integrate PV production into the energy mix come up against the challenge of controlling its high intermittency and variability [3]. A source of this variability is short-term meteorological effects, especially dynamic changes in cloud coverage over PV facilities. Cloud features, such as their thickness, distribution in the sky, and position with regard to the sun, significantly affect the incoming solar irradiance, which is the most important factor in PV power generation. The ability to locally forecast

short-term irradiance for PV facilities plays a key role in controlling the intermittence of PV-generated electricity [4]. However, forecasting methods that are based solely on historical irradiance data cannot reliably model the short-term effects of cloud flows. On the other hand, image analysis techniques and CV-based methods can provide information about the state of the clouds in the sky with a high spatial resolution when the images are extracted from sky imagers (SIs) located close to the PV panels [5]. Furthermore, continuous advances in the field of Machine Learning (ML), and particularly Deep Learning (DL) and Convolutional Neural Networks (CNNs), have led modern CV tasks and applications to employ these DL and CNN techniques. The combination of CNN-based sky image processing with irradiance data with high temporal frequency (at least 1 min) constitutes a very promising direction towards short-term (up to 15 min) forecasting, sometimes called nowcasting, of irradiance [6]. The current availability of high quality and publicly accessible image datasets [7–9] is in favor of the image-based ML approaches for irradiance forecasting as well. Finally, DL-based image processing is following the trend of moving from cloud servers towards edge devices in order to enable smart applications and the Internet of Things (IoT) [10]. Edge devices can have resource and energy constraints, which has led to the development of lightweight CNN models such as MobileNetV2 [11] and SqueezeNet [12].

Aiming at an improved performance in short-term irradiance forecasting using sky images, the current article introduces an image processing block, the Image Regression Module (IRM). The IRM is a CNN-based block that processes a sky image and produces a single value, out of a continuous range, that represents the estimated irradiance for the particular sky image. The IRM is modular and can serve the task of irradiance forecasting by processing an externally predicted future sky image or perform as standalone by being trained on sky images with backwards shifted irradiance values. The development and evaluation of CNN models and image processing methods for the IRM is based on the Folsom, CA dataset [13], an extensive and publicly available dataset. Moreover, this work aims to support the concept of the edge-enabled smart PV park [14] that meets the real-time requirements of the PV power production control. For this purpose, we implement and accelerate the CNNs of the IRM on an edge device, the Xilinx Zynq UltraScale+ MPSoC Field-Programmable Gate Array (FPGA). The current work makes the following contributions:

1. To improve the performance of image regression CNN models, an image processing method based on sun localization is proposed which improves the accuracy of the irradiance values that the models produce, by up to 13.75% for the MobileNetV2 model.
2. To showcase the applicability of the proposed method for many CNN models and for both irradiance estimation and forecasting, a study on four popular CNN models is conducted, where the method improves the results in all cases.
3. To demonstrate the concept of a smart PV park with edge computing capabilities, we deploy the image regression CNN models on an edge FPGA using the Xilinx Vitis AI framework, achieving real-time processing rates.

The rest of this paper is organized as follows. First, Section 2 reports related results in the literature. Then, Section 3 presents a formulation of the image regression for the irradiance forecasting problem and an analysis of the dataset that we used. Section 4 introduces the proposed image processing method based on sun localization. Section 5 describes the development flow for porting the CNN models to the edge FPGA. Section 6 presents evaluation results for both the performance of the CNN models as well as FPGA implementation results. Finally, Section 7 concludes the article.

## 2. Related Work

Image processing techniques towards irradiance forecasting can be applied on satellite imagery [15–17]. However, images obtained from fisheye lens (180° field of view) SIs local to the PV parks can provide increased spatial and temporal resolution. This fact can

favor the precise irradiance forecasting for specific areas, such as PV parks, and for very short-term forecast horizons of up to 15 min.

Several works in the literature propose image processing techniques to extract information regarding cloud coverage from sky images for irradiance forecasting [18–21]. The authors of [18] used ground-based sky images statistical features such as the Red Blue Ratio (RBR) and Red Blue Difference (RBD) for cloud cover calculation. The authors performed cloud cover forecasting using machine learning approaches such as Support Vector Regression (SVR) and Artificial Neural Network (ANNs). In [19], the authors introduced the Hybrid Thresholding Algorithm (HYTA) for cloud detection, which utilizes Normalized Red Blue Ratio (NRBR) image metrics and adaptive thresholding methods. The authors in [20] proposed a method for forecasting irradiance using the cloud cover index. They used the RBR to calculate the cloud cover index, a Long Short-Term Memory (LSTM) model for forecasting the future cloud cover index and a numerical solar radiation model for calculating future irradiance. The work of [21] used image processing methods to identify the cloud coverage and combined it with the clear sky index and LSTM models to perform irradiance forecasting. The current work is different from the above results, as it proposes an image processing method that precedes the corresponding CNN processing and provides information to CNNs regarding the position of the sun in the image instead of the cloud coverage features.

Regarding sun localization in sky images, different image processing approaches have been reported in the literature [22–24]. In [22], the authors introduced a sun localization algorithm based on pixel values and a masking algorithm for background removal based on edge detection. Numerical NRBR values were generated from the processed sky images and these values were fed to a Multilayer Perceptron (MLP) network for irradiance forecasting. The work of [23] implemented and compared three different approaches for the identification of the sun position in sky images targeting nowcasting applications. The ML-based approach was shown to overcome the ones that are based on solar coordinates calculation and traditional image processing. The sun tracking algorithm introduced in [24] used image pixel values to identify the position of the sun when it is visible and an interpolation method when it is not. While the aforementioned works performed sun localization based on pixel values, the proposed method utilizes the solar angles and the mapping function of the fisheye lens to accurately calculate the position of the sun in the image regardless of any sky image effects such as clouds, high-intensity glare, and background objects.

The reported results that are the most closely related to the current work are the ones which perform sky image regression using Deep Learning to produce irradiance values [25–29]. The work in [25] provided an in-depth comparison of deep learning model types for short-term irradiance forecasting from sky images. The four model types, namely, CNN, CNN plus LSTM, 3D-CNN, and Convolutional LSTM (ConvLSTM), were all shown to achieve around 20% forecast skill on the 10 min ahead prediction. The authors in [26] used the ResNet CNN model for performing irradiance regression and forecasting tasks for forecasting. By stacking different RGB channels of past images on the input of the model they could achieve up to 18% forecast skill for a 10 min horizon. In [27], the authors proposed replacing costly irradiance measurement instruments, e.g., pyranometers, with a CNN-based image regression model to produce irradiance values. They particularly focused on cloudy days that have the most sudden irradiance deviations, showing the feasibility of their approach. The work in [28] investigated the performance of CNN and LSTM models on image regression for irradiance mapping. Apart from the deterministic methods, it applied probabilistic ones to statistically evaluate the performance of the models. The authors of [29] performed irradiance nowcasting from sky images by enhancing a traditional CNN with attention modules to improve its performance. It is worth noting that all of the above works utilized datasets with a relatively limited number of samples of up to a few tenths of thousands. In contrast, we utilize the entire Folsom, CA dataset, which contains three consecutive years of data with more than 700,000 image samples. This allows us to extract the evaluation results of the CNN models from an extensive test

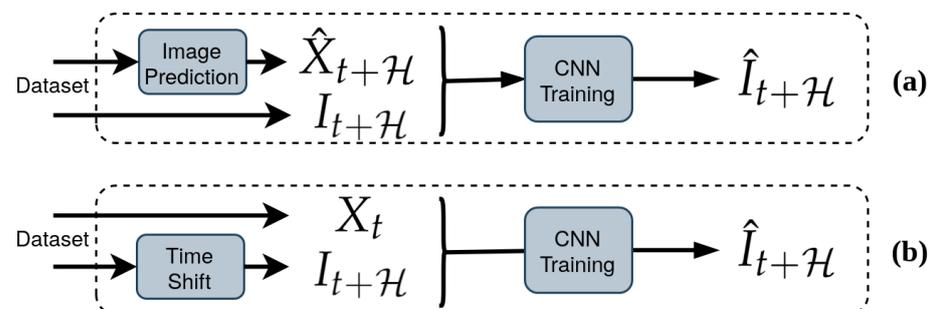
dataset of one whole year, which includes indicative samples of all sky image effects. By examining the performance on individual months, we quantify how the distributions of sky image phenomena affect the irradiance forecasting problem. Finally, we showcase the task of image regression for irradiance estimation performed on an edge FPGA. The CNN models of the study all achieve real-time processing rates on the FPGA towards the real-time control of PV power production in a smart PV park.

### 3. Problem Formulation & Image Dataset Analysis

In the current section, we first present the task of the image regression for irradiance estimation and forecasting and describe its main objectives. Then, we explain the dataset pre-processing steps and perform an analysis of the Folsom, CA dataset utilized in this work. Through this analysis, we highlight the sky image features and their distributions, which play a key role in the performance of the CNN models that perform image regression for irradiance estimation and forecasting.

#### 3.1. Image Regression for Irradiance Forecasting Problem Formulation

The problem of image regression for irradiance estimation can be formulated as follows. Considering a dataset that consists of a paired sky image  $X_t$  and corresponding irradiance measurement  $I_t$  at time  $t$ , an image regression CNN model can be trained to produce a single value that represents an estimation of the irradiance  $\hat{I}_t$  when provided with the sky image  $X_t$ . With the above formulation, the image regression for irradiance estimation can be expanded to image regression for irradiance forecasting in two ways. First, assuming an external module that can predict the future sky image  $\hat{X}_{t+\mathcal{H}}$  after time  $\mathcal{H}$  in a sequence [30–32], the image regression CNN model can then produce the corresponding estimation of the irradiance  $\hat{I}_{t+\mathcal{H}}$  from this image. Alternatively, an image regression CNN model can be trained on sequences of images where the corresponding irradiance values have been shifted backwards in time by  $\mathcal{H}$ . In this way, when the model is provided with a sky image  $X_t$  at time  $t$  it produces the irradiance forecast  $\hat{I}_{t+\mathcal{H}}$ . The above image regression for the irradiance forecasting problem formulation is illustrated in Figure 1.



**Figure 1.** Irradiance forecasting using an image regression CNN with either (a) an external module for future sky image prediction or (b) irradiance values shifted backwards in time by  $\mathcal{H}$ .

The objective of the tasks described above is to minimize the error between the actual  $I_t$  values and the  $\hat{I}_t$  values produced by a CNN model after processing an image. In the evaluation of such models, it is common to use the error metrics of mean absolute error (MAE) and root mean square error (RMSE). In order to compare the performance of a model on different subsets that can have largely different distributions of irradiance values (e.g., a cloudy winter month vs. a clear sky summer one), we emphasized the normalized RMSE (nRMSE), which we calculate using the mean irradiance value,  $\bar{I}$  of the set as

$$nRMSE = \frac{RMSE}{\bar{I}}. \quad (1)$$

Regarding forecasting, the Persistence Model (PM) serves as a reference forecasting model against which to benchmark other CNN models. It assumes that the predicted future irradiance value  $\hat{I}_{t+\mathcal{H}}$  remains unchanged from the current irradiance value  $I_t$  into the forecast horizon  $\mathcal{H}$ , and can be expressed as

$$\hat{I}_{t+\mathcal{H}} = I_t. \quad (2)$$

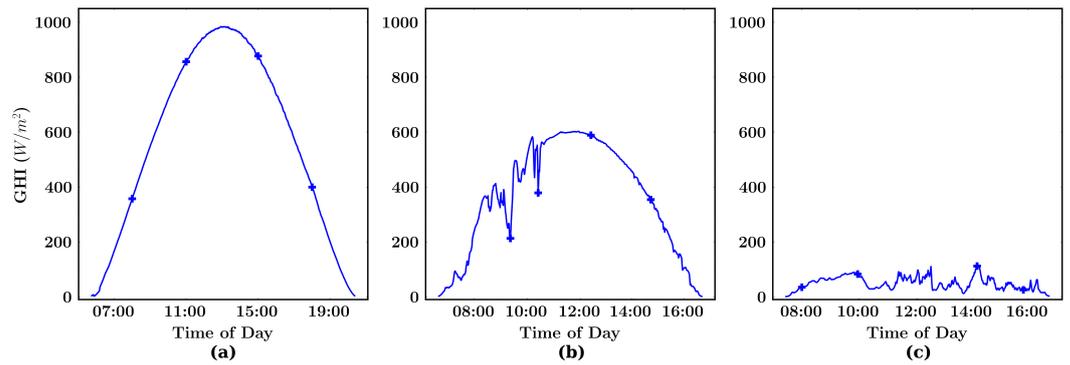
In order to measure the improvement of any model against the persistence model, the Forecast Skill (FS) metric is introduced. It is common for solar irradiance forecasting applications to use the root mean square error of the persistence model, denoted as  $RMSE_P$ , and of the benchmarked model,  $RMSE_M$ , for the FS calculation, as follows:

$$FS = 1 - \frac{RMSE_M}{RMSE_P}. \quad (3)$$

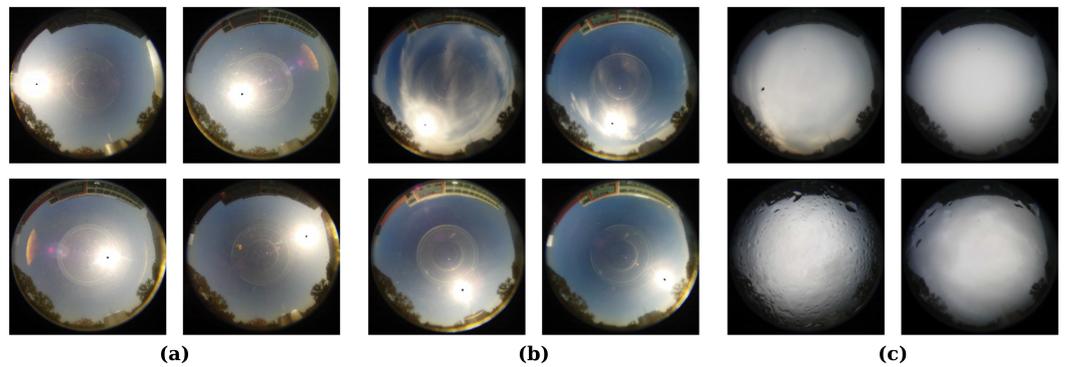
### 3.2. Folsom, CA Dataset Analysis

For the development and evaluation of image processing methods and CNN models for our IRM block, we utilize the Folsom, CA dataset introduced in the work of [13] and publicly available in [8]. The dataset contains high-resolution  $1536 \times 1536$  sky images of three consecutive years, 2014, 2015, and 2016 taken at a site in Folsom, CA, USA ( $38.642^\circ$ ,  $-121.148^\circ$ ). The images come with corresponding Global Horizontal, Direct Normal, and Diffuse Horizontal Irradiance (GHI, DNI, DHI) measurements, and have a high temporal resolution of 1 min, making the dataset suitable for very short-term irradiance forecasting. The GHI describes the total irradiance from the sun on a horizontal surface, including both the DNI and DHI terms; thus, we focus on GHI for irradiance forecasting. In the remaining sections of this paper, the term irradiance refers to the GHI. The first step towards adjusting the dataset for our application is to perform data cleaning. In particular, several images are missing corresponding irradiance measurements, and vice versa. For this purpose, the first step taken was the rounding of the image timestamps to the closest integer minute value. Then, we matched the images to the irradiance timestamps and formulated pairs that include both an image and a valid irradiance measurement. During the latter step, we removed all the samples corresponding to the very low solar elevation values during the night. At this point, we performed dataset analysis.

In order to distinguish the image features that affect the irradiance measurements, we first performed a daily analysis. A single day demonstrates the lowest degree of periodicity in sky image events, that is, the daily change of the sun's elevation angle. To perform the analysis, we transformed the original dataset timestamps from the Universal Time Coordinated (UTC) +0 timezone to the Los Angeles timezone (UTC-7, UTC-8) corresponding to the location of the sky imager in order to correctly identify the day and night periods. Figure 2 presents daily GHI plots for three indicative days, while Figure 3 shows the sky images which correspond to the markers of each daily GHI plot. From the first day, which includes only clear sky images, it can be observed that the position of the sun in the image, specifically, its distance from the image center, is the feature that is directly correlated with the GHI measurement. The second day, which features both clear sky and complex cloud effects in the images, indicates that there can be fluctuations and sudden drops in the measured irradiance when clouds obstruct the sun. Moreover, due to the larger distance of the sun from the center of the image during noon (lower elevation angle), the peak irradiance measurement is lower when compared to the first day. Finally, in the third day, which is characterized by images with only overcast sky, the measured GHI maintains consistent and very low values.

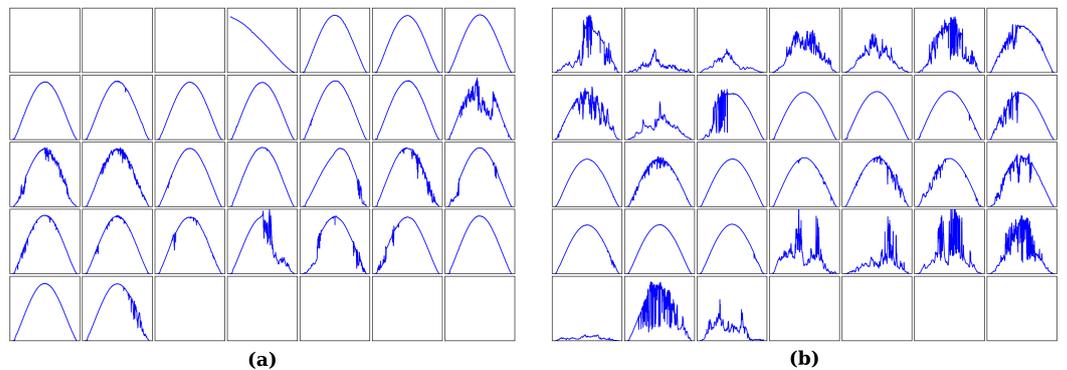


**Figure 2.** Daily GHI plots for the days (a) 7 June 2014, (b) 3 November 2014, and (c) 24 December 2014.



**Figure 3.** Sky images corresponding to the markers on the daily irradiance plot, from left to right and top to bottom, for the days (a) 7 June 2014, (b) 3 November 2014, and (c) 24 December 2014.

Based on the observations of the daily analysis, we expect CNN models that perform image regression to produce a less accurate irradiance value in cases with complex cloud effects and overcast sky, resulting in a larger nRMSE. The reason for this is the lack of information regarding the position of the sun in the image, which is the main image feature that the CNNs can model in order to produce an irradiance value. We note here that clear days consistently occur in summer months, while days with complex cloud effects occur during the rest of the year. We highlight this in Figure 4, where we show all the daily GHI plots for June and March 2014 arranged in a calendar. Thus, we can quantify the effects of cloud phenomena on the performance of CNN models by evaluating them on different months that have different distributions of cloudy vs. clear sky days.

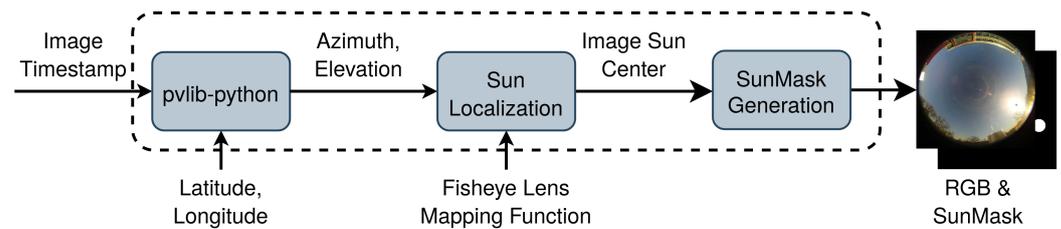


**Figure 4.** Daily GHI plot calendars of two different months: (a) June 2014 and (b) March 2014.

#### 4. SunMask Generation Image Processing Method

Based on the analysis in Section 3, we propose an image processing method to support the CNNs in providing more accurate irradiance estimations from sky images. The method

is based on sun localization in the image. The intuition behind it is that the CNNs can model the position of the sun in the image and the cloud effects in the sun disk area in order to produce an irradiance value. The proposed method consists of locating the sun's center in the image and generating the SunMask, a circular mask around the center of the sun which we append as an additional fourth channel to the original three-channel RGB image. This method provides information to the CNN models about the position of the sun's center in the image, which is particularly useful in images where the sun is covered by clouds. The steps in our image processing method are summarized in Figure 5, and are explained in detail in the following paragraphs.



**Figure 5.** Summary of the steps in the SunMask generation image processing method.

The first step in our method is to calculate the center of the sun disk in the image. Approaches that are based on the RGB values of the image often fail to identify the position of the sun when it is hidden by clouds. Furthermore, high-intensity background objects and glare effects can be erroneously identified as the sun. Due to the above, we opt for an approach based on the solar azimuth and zenith angles,  $\phi_s$  and  $\theta_s$ . The summary of our approach to sun localization is illustrated in Figure 6. Considering the timestamp of an image along with the latitude and longitude coordinates, we calculate  $\phi_s$  and  $\theta_s$  using pvlib-python [33]. Because of the fisheye distortion of the lens of the camera, simply projecting the solar coordinates on a flat surface does not result in accurate identification of the sun's center in the image. To address this, we use the mapping function of the lens to calculate the distance of the sun from the center of the image  $R$  as a function of the focal length of the lens  $f$  and the angle from the optical axis  $\Phi$ . The mapping functions vary based on the type of projection (stereographic, equidistant, etc.) of the lens, and neither this information nor the focal length is available for the camera used to create the Folsom, CA dataset. Through trial and error, we were able to identify that the stereographic projection with a focal length of  $f = 0.48$  provides the most accurate results. Thus, we use the mapping function of the stereographic projection

$$R = 2f \tan \frac{\Phi}{2} \quad (4)$$

where we replace  $\Phi$  with the solar zenith angle  $\theta_s$ . After calculating the distance of the sun from the center of the image, we calculate the Cartesian coordinates  $x_c, y_c$  using the solar azimuth angle  $\phi_s$  with

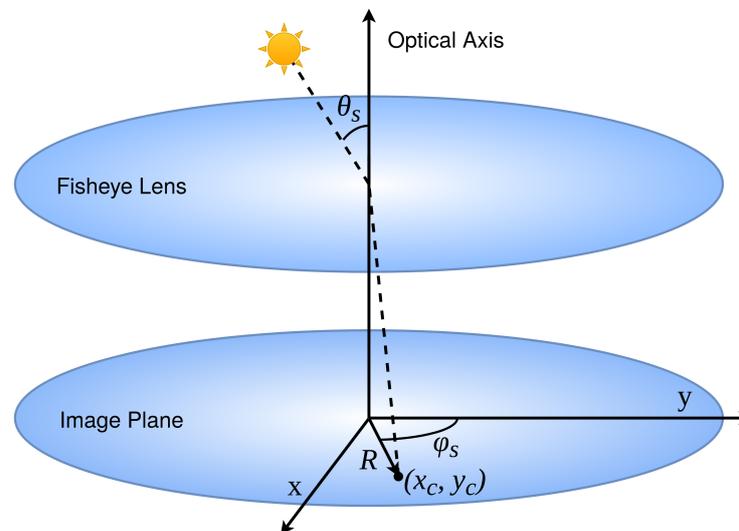
$$x_c = R \sin(\phi_s), \quad (5)$$

$$y_c = R \cos(\phi_s). \quad (6)$$

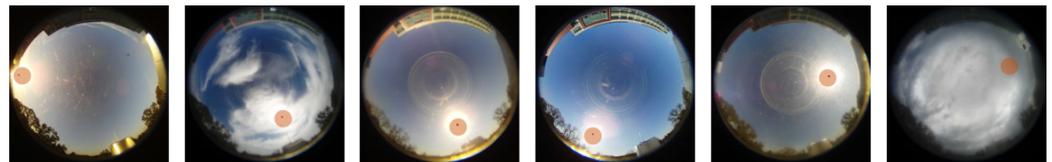
In our case, we adjust  $\phi_s$  with an angular correction of  $165^\circ$  to compensate for the orientation of the camera. Finally, we transform the Cartesian coordinates  $x_c, y_c$  to pixel coordinates  $x_p, y_p$  in the image by multiplying them with the image radius which is  $ImageWidth/2$ .

After calculating the position of the sun's center in the image, we generate the SunMask and append it as a fourth channel to the RGB image. The SunMask consists of a mask around the center of the sun with a pixel value of 255, while the rest of the pixels have a value of 0. In Figure 7, we showcase the results of our image processing method for indicative images, with the SunMask portrayed in orange. It can be observed that the proposed sun localization method accurately identifies the position of the sun in all the

images, regardless of their variations in terms of the solar azimuth angle, zenith angle, and cloud coverage of the sun.



**Figure 6.** Overview of our sun localization approach on a sky image produced by fisheye lens.



**Figure 7.** The SunMask, in orange colour for visualization purposes, for a few selected image samples.

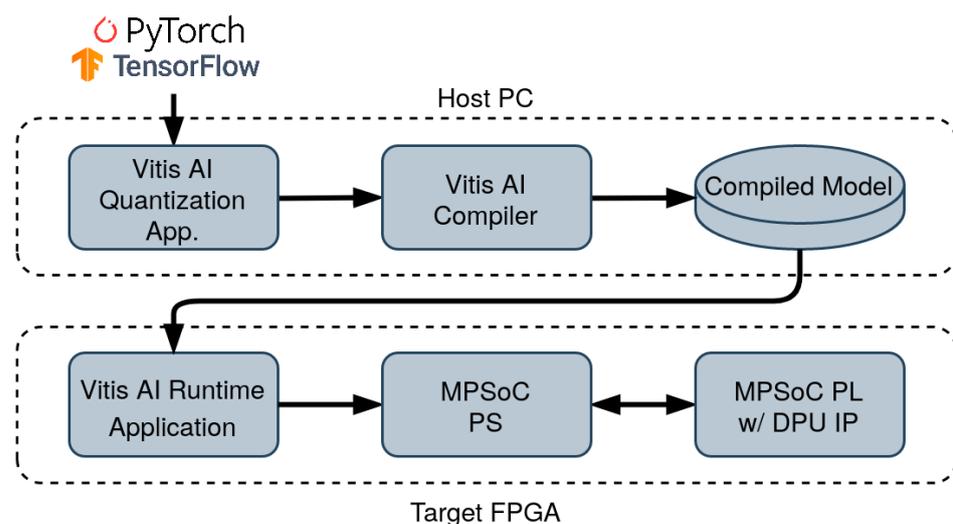
Based on the above, we applied the two steps used in our image processing method to the entire Folsom, CA dataset and provided four-channel images to the Irradiance Regression Module during both training and testing of CNN models. With the IRM configured to perform forecasting in standalone mode by being trained on backwards-shifted GHI data, we perform one additional image processing step. Instead of providing only one four-channel image (RGB & SunMask), denoted as  $X_t$ , we stack two additional four-channel images,  $X_{t-\mathcal{H}}$  and  $X_{t-2\mathcal{H}}$ , which correspond to two steps backwards in the forecast horizon  $\mathcal{H}$ . As a result, the IRM operates on an input of 12 channels in total. For example, if the IRM is trained to perform forecasting 5 min ahead, we provide as input the images  $X_t$ ,  $X_t - 5$  and  $X_t - 10$  and their corresponding SunMasks. In this way, the IRM has additional information regarding the past state of the sky.

## 5. Porting and Acceleration on Edge FPGA

In the current work, in order to support the concept of an edge-enabled smart PV park that can fulfill the control requirements of real-time PV, we implement the CNN models of our Irradiance Regression Module on an FPGA device targeting an edge computing application, the Xilinx Zynq UltraScale+ MPSoC [34]. This System-on-Chip (SoC) family of FPGAs features heterogeneous resources, including both an ARM-based Processing Subsystem (PS) and a configurable fabric known as the Programmable Logic (PL). For porting the models to the FPGA, we utilized the Xilinx Vitis AI framework [35]. This framework provides an end-to-end set of tools that allows the developer to port and execute CNN models from popular frameworks such as PyTorch and Tensorflow to Xilinx FPGAs. The development workflow that we followed for implementing and accelerating our models on the FPGA is summarized in Figure 8, and is described in the following paragraphs.

On the host PC side, the tools included the Vitis AI Quantizer. Using the Quantizer, we first developed a Python application for quantizing our 32-bit floating-point CNN model descriptions to the corresponding 8-bit fixed-point ones, as required by the framework. Vitis AI offers several quantization related solutions, such as Post-Training Quantization (PTQ), which implements the cross layer equalization algorithm [36], Fast Fine-tuning (FF), which implements the AdaQuant algorithm [37], and Quantization-Aware Training (QAT). In this work, we combine the PTQ, FF, and QAT solutions to reduce the effect of quantization to the performance of the studied models as much as possible. After quantization, we utilized the Vitis AI Compiler to produce the graph description and instructions for the compiled model that are executed on the target FPGA during runtime.

On the target FPGA side, the framework provides the Deep-Learning Processor Unit (DPU) IP Core. The DPU is a programmable computation engine which is implemented on the PL resources of the target FPGA. Its architecture and the set of instruction it supports is highly optimized for accelerating a wide range of operations that most of the popular CNNs require. In the current work, we configured the DPU for optimized resource utilization on the FPGA in order to enable the employment of two processing cores of the DPU for increased throughput. The DPU is integrated in the FPGA design, which includes the ARM-based PS. After FPGA programming, we developed the runtime application that is executed on the PS of the FPGA and controls the DPU during runtime. The development of the runtime application was based on the APIs that the Vitis AI Runtime library provides for developers, which are available in both C++ and Python.



**Figure 8.** The Vitis AI development flow that we followed in the current work.

## 6. Model Evaluation and Implementation Results

In the current section, we present the evaluation results of the studied CNN models and our image processing method on the task of image regression for irradiance estimation and forecasting. We study the performance of the VGG11 [38] and ResNet-50 [39] models, for which similar comparative studies have been conducted on various applications [26,40]. We study the performance of two CNN models which target edge devices with limited resources, namely, the MobileNetV2 [11] and SqueezeNet [12] models. With this selection of models, we cover a wide range of model sizes and number of operations, as shown in Table 1. The models and all training and evaluation processes were implemented in Python using PyTorch on a Linux workstation with an Intel(R) Core(TM) i7-9700K CPU @ 3.60 GHz and NVIDIA GeForce RTX 3080 GPU. The edge FPGA deployment and evaluation of the models was performed on a Xilinx ZCU104 FPGA board using the Xilinx Vitis 2021.2, Vivado 2021.2, and Vitis AI 2.0 tools.

**Table 1.** Number of parameters and number of operations for the four models of the study.

Model	# Parameters	# OPs (Mult-Adds)
VGG11	128.77 Mil.	2.57 G
ResNet-50	23.51 Mil.	1.33 G
MobileNetV2	2.23 Mil.	0.10 G
SqueezeNet	0.74 Mil.	0.23 G

### 6.1. Image Regression Models Training and Performance Evaluation

First, we trained the four models of the study to perform irradiance estimation from sky images. The training dataset included the years 2015 and 2016 (522,320 samples), while the test dataset consisted of the entire year 2014 (240,944 samples). During training, we used the Mean Square Error (MSE) loss function, which is suitable for image regression tasks. The hyperparameters were tuned based on the RMSE results of the trained models on the entire test dataset. Tuning was performed for the ResNet-50 model and the hyperparameters were kept the same for all the training procedures in the work in order to keep the experimental environment consistent. All the models are trained for ten epochs, which was identified as sufficient, because all models showcased overfitting after only a few epochs. This can be attributed to the high number of training steps during each epoch due to the large size of the training dataset. Regarding the batch size and the image size, these were limited to 16 and  $128 \times 128$ , respectively, due to GPU memory limitations and the training time required. Regarding the learning rate, it was initialized to  $10^{-3}$  and automatically tuned by a scheduler that reduced the learning rate by a factor of 0.75 if the validation loss plateaued for five epochs.

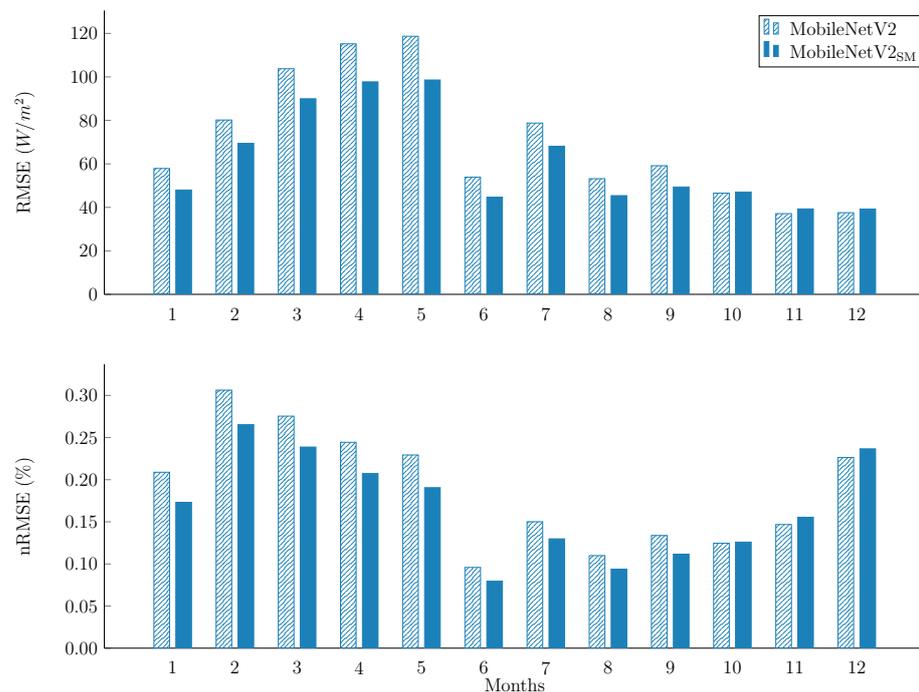
The training dataset was split into the training subset and validation subset for evaluation of the models during training. Instead of a random split, we selected all the samples of one random day of each consecutive month of the training dataset and added them to the validation subset until we reached the desired training–validation split ratio of 80–20%. In this way, the validation subset included indicative samples of the entire dataset in terms of the yearly periodical phenomena in the sky images. Furthermore, we avoided including in the validation subset samples only 1 min apart from almost identical ones in the training subset. We evaluated the performance of the models after every training epoch and selected the saved model with the minimum validation loss to avoid overfitting.

In Table 2, we present the performance evaluation results of the four different models on the original dataset as well as on the dataset with the images enhanced with the SunMask channel. The ResNet-50 model appears to result in the lowest error metrics, with an RMSE of  $64.83 \text{ W/m}^2$ . When introducing our SunMask generation method, it can be observed that it consistently improves the performance of all models. The MobileNetV2 model is favoured the most by our SunMask generation method, with its RMSE decreasing from  $75.95 \text{ W/m}^2$  to  $65.51 \text{ W/m}^2$ , a 13.75% improvement.

**Table 2.** Evaluation and comparison of the four models; SM indicates the models trained and evaluated with the four-channel images that include SunMask and the improved models are highlighted in bold.

Model	RMSE ( $\text{W/m}^2$ )	nRMSE (%)	MAE ( $\text{W/m}^2$ )
VGG11	65.25	15.88	38.31
<b>VGG11<sub>SM</sub></b>	<b>59.03</b>	<b>14.36</b>	<b>32.93</b>
ResNet-50	64.83	15.77	37.23
<b>ResNet-50<sub>SM</sub></b>	<b>60.31</b>	<b>14.67</b>	<b>36.02</b>
MobileNetV2	75.95	18.48	47.74
<b>MobileNetV2<sub>SM</sub></b>	<b>65.51</b>	<b>15.94</b>	<b>39.53</b>
SqueezeNet	70.18	17.08	44.65
<b>SqueezeNet<sub>SM</sub></b>	<b>62.93</b>	<b>15.31</b>	<b>38.56</b>

In order to quantify the effect that the SunMask generation method has on different distributions of sky image phenomena, we evaluated the MobileNetV2 model, which shows the largest improvement with the SunMask, on individual months. The RMSE and nRMSE results are presented in Figure 9. We note here that we focus on the nRMSE metric in order to directly compare the performance of the models across different months. The reason for this is that the distribution of the sky image features and irradiance values is different for each month. Thus, a month with a smaller RMSE can correspond to a larger nRMSE, and vice versa; this is apparent in months such as February, May, and December. The nRMSE plot indicates that, in general, the model tends to perform better in months when there are less complex effects of clouds obstructing the sun in the image, such as June and August. Furthermore, the improvement in the performance thanks to SunMask is larger in months with complex sky image phenomena. In particular, MobileNetV2 shows the largest nRMSE improvement with SunMask in February (4.1%), May (3.9%), March (3.7%), April (3.7%), and January (3.6%). The current work focuses on the dataset generated at the location of Folsom, CA, USA ( $38.642^\circ$ ,  $-121.148^\circ$ ). However, it would be worth exploring the performance of image regression CNNs for irradiance estimation and of the proposed SunMask generation method on other geographic areas. These geographic areas could include regions with significantly different sky image feature distributions, such as higher latitude regions where the sun is close to the horizon. Of course, this would require an extensive, publicly available, and high-quality dataset for the particular area. The availability of such extensive, annotated, and public datasets is currently an open issue in the field of Deep Learning-based irradiance forecasting and Machine Learning in general.



**Figure 9.** RMSE and nRMSE monthly plots for MobileNetV2 with and without the SunMask.

Following the results for irradiance estimation, we selected the ResNet-50 model, which had the best performance, to perform standalone irradiance forecasting. For this purpose, we shifted the irradiance values of the dataset backwards in time by the forecast horizon  $\mathcal{H}$ , as explained in Section 3. We formulated the dataset in this way for three different forecast horizons 5, 10, and 15 min ahead. We trained the ResNet-50 model similarly to before, with the simple RGB input and the 12-channel stacked SunMask images described in Section 4. The results of the ResNet-50 model on image regression for

irradiance forecasting and the persistence model are shown in Table 3. It can be observed that the ResNet-50 model achieves a forecast skill which is incremental with regards to the forecast horizon, having slightly worse forecasting performance than the persistence model for the 5 min forecasting horizon. When ResNet-50 is trained to operate on the proposed stacked SunMask images, the results show that it achieves consistently improved forecast skill for all forecast horizons. Using this method, the ResNet-50 model can surpass the persistence model even in the very short-term forecast horizon of 5 min, adding 8.79% to its forecast skill.

**Table 3.** Standalone irradiance forecasting results for the ResNet-50 model on 5, 10, and 15 min horizons. SSM indicates that the input is three stacked images of four channels each (i.e., RGB and SunMask) and the improved results are highlighted in bold.

Horizon	Model	RMSE (W/m <sup>2</sup> )	nRMSE (%)	FS (%)
5-min	Persistence	72.64	17.32	-
	Resnet-50	73.18	17.44	-0.75
	<b>Resnet-50<sub>SSM</sub></b>	<b>66.79</b>	<b>15.93</b>	<b>8.04</b>
10-min	Persistence	86.77	20.26	-
	Resnet-50	78.06	18.23	10.04
	<b>Resnet-50<sub>SSM</sub></b>	<b>73.06</b>	<b>17.06</b>	<b>15.80</b>
15-min	Persistence	94.52	21.67	-
	Resnet-50	80.87	18.54	14.44
	<b>Resnet-50<sub>SSM</sub></b>	<b>75.78</b>	<b>17.37</b>	<b>19.83</b>

## 6.2. Edge FPGA Porting and Acceleration Results

Following the evaluation of the performance of the CNN models, including those that included our proposed image processing method, in this subsection we present the results of our implementation flow on the edge-oriented Xilinx MPSoC FPGA. The first step towards implementing the CNN models of the IRM on the FPGA is to perform quantization. The Python quantization application that we developed utilizes several different quantization functionalities of the Xilinx Vitis AI 2.0 Quantizer. The results of the quantization process for the four different models of our study are presented in Table 4. First, we performed Post-Training Quantization using a batch of unlabeled images. It can be observed that the PTQ has a very significant effect of 111.23 W/m<sup>2</sup> increased RMSE compared to the performance of the original floating-point VGG11 in Table 2. After performing an additional Fast Fine-tuning step using 1000 unlabeled images, the effect was reduced to 4.26 W/m<sup>2</sup>. For the ResNet-50 model, PTQ results in a slight increase in RMSE of 3.52 W/m<sup>2</sup>, which is reduced to 2.18 W/m<sup>2</sup> with FF. MobileNetV2 suffers a loss in performance which cannot be corrected even after FF, resulting in a loss of 12.47 W/m<sup>2</sup>. Finally, the quantized SqueezeNet model has a large performance degradation from its original floating-point model, with an increase in RMSE of 19.76 W/m<sup>2</sup>. With FF the increase in RMSE becomes 6.84 W/m<sup>2</sup>. The SqueezeNet model architecture allows us to perform an additional Quantization-Aware Training step instead of FF. We trained the SqueezeNet model for an additional epoch using the QAT capabilities of the Vitis AI Quantizer. After the QAT step, the performance of SqueezeNet is restored to a similar level as the original floating-point model, suffering an increase of only 2.09 W/m<sup>2</sup> RMSE.

**Table 4.** Performance metrics for the four quantized models with combinations of PTQ, FF, and QAT.

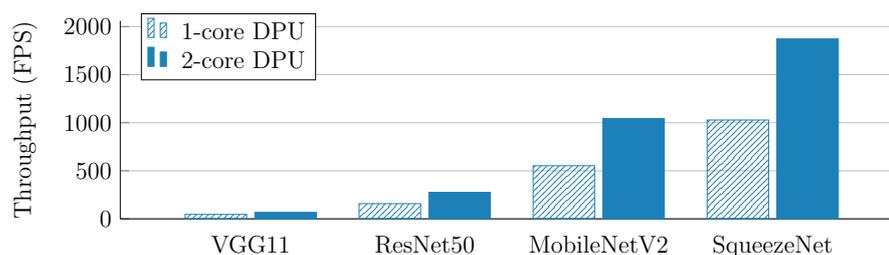
Model	Quantization Method	RMSE (W/m <sup>2</sup> )	nRMSE (%)	MAE (W/m <sup>2</sup> )
VGG11	PTQ	176.48	42.94	38.31
	PTQ & FF	69.51	16.91	41.04
ResNet-50	PTQ	68.37	16.63	40.39
	PTQ & FF	67.01	16.30	39.18
MobileNetV2	PTQ	88.42	21.51	59.91
	PTQ & FF	88.78	21.60	60.67
SqueezeNet	PTQ	89.94	21.89	63.24
	PTQ & FF	77.02	18.74	50.30
	PTQ & QAT	72.27	17.58	45.84

After quantization of the CNN models, we implemented the FPGA processing architecture described in Section 5 on the Xilinx ZCU104 FPGA board using the Xilinx Vitis and Vivado 2021.2 tools. The dual-core DPU IP was operated at 300 MHz; the entire design consumed 15.585 W as measured by the Vivado power analysis tool. In Table 5, we present the resources utilization of the PL of the FPGA. We observe that the dual-core DPU IP consumes a very significant amount of resources, especially regarding the Digital Signal Processing (DSP) slices responsible for performing most computations. It is worth noting that for applications where processing throughput is not critical, the developer can configure the DPU with a single processing core, reducing resource utilization by about half for most resources.

**Table 5.** FPGA resource utilization for the implemented design on the ZCU104 board.

Resource	2-Core DPU IP	1-Core DPU IP
LUTs	108K (47%)	50K (22%)
FFs	204K (44%)	98K (21%)
DSPs	1394 (81%)	690 (40%)
RAMBs	203 (65%)	145 (46%)

In order to showcase the real-time capabilities of the edge FPGA on the image regression task, we benchmarked the four different CNNs on the DPU IP Core. We evaluated their throughput in terms of Frames per Second (FPS) on both cores of the DPU using multithreading; the results are presented in Figure 10. The results show that the VGG11 model with the highest number of parameters and operations has the lowest throughput of 46 FPS on a single core of the DPU. The lightweight MobileNetV2 and SqueezeNet models showcase significantly higher throughput than the VGG11 and ResNet-50 models. In particular, SqueezeNet has the highest throughput of 1028 FPS, even though its original floating-point model requires more operations with a lower number of parameters than MobileNetV2. When utilizing both cores of the DPU, all the models can achieve a little less than  $\times 2$  throughput. For the ResNet-50, MobileNetV2 and SqueezeNet models, the achieved throughput rates can be considered to satisfy real-time requirements, e.g., for a sky imager providing a video at 60 FPS, leaving space for additional algorithms to complete more PV related processes.



**Figure 10.** Throughput results for the four studied models on one and two processing cores of the DPU.

## 7. Conclusions

The current work presents the Image Regression Module, a CNN-based block for performing sky image regression to produce irradiance values. We have highlighted the findings from our analysis of the Folsom, CA image dataset, which includes sky image features that can affect the performance of CNN models. For better performance in this regard, we propose the SunMask generation image processing method to support CNNs in providing more accurate results for irradiance estimation and forecasting tasks. Our study on the VGG11, ResNet-50, MobileNetV2, and SqueezeNet models for the IRM shows that the SunMask generation method can consistently improve the RMSE of the irradiance values produced by all of these models by up to 13.75%. The current study on the performance of the CNN models and our image processing method was limited to only one publicly available dataset which covers a geographic area with a particular distribution of sky image phenomena. As a future step, we aim to extend our study to include more regions, particularly from our region of interest, namely, Greece, where we have already installed a sky imager and a pyranometer in order to create our own dataset. Furthermore, in the current work we utilized only image regression CNNs. Future work might include more advanced DL models such as ConvLSTMs to produce future sky images for irradiance forecasting. Finally, we have outlined the concept of a smart PV park with edge computing capabilities achieved by implementing CNN models, such as in our study of the Xilinx MPSoC FPGA targeting edge applications and using the Xilinx Vitis AI framework for development. The results show that the FPGA can achieve real-time processing rates for the image regression CNN process. In the future, we plan on extending the proposed edge implementation with more processes in order to present a complete and real-time irradiance forecasting application for edge computing applications.

**Author Contributions:** Conceptualization, E.A.P. and D.R.; methodology, E.A.P. and V.K. and G.V. and F.T.; software, E.A.P. and V.K. and G.V. and F.T.; validation, E.A.P. and V.K.; formal analysis, E.A.P. and V.K. and G.V. and F.T.; investigation, E.A.P. and V.K.; resources, D.R.; data curation, V.K.; writing—original draft preparation, E.A.P.; writing—review and editing, D.R.; visualization, V.K.; supervision, D.R.; funding acquisition, D.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH—CREATE—INNOVATE (project code:T2EDK-00864). The research work was supported in part by the Hellenic Foundation for Research and Innovation (HFRI) under the Third Call for HFRI PhD Fellowships (Fellowship Number: 6220).

**Data Availability Statement:** The Folsom, CA sky images dataset analyzed and processed in the current work is introduced in [13] and the repository [8] can be accessed through the following link <https://zenodo.org/record/2826939#.Ywd6edVBw5k> (accessed on 15 October 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CV	Computer Vision
ConvLSTM	Convolutional LSTM
CNN	Convolutional Neural Network
DL	Deep Learning
DPU	Deep Learning Processor Unit
DSP	Digital Signal Processing
DHI	Diffuse Horizontal Irradiance
DNI	Direct Normal Irradiance
FF	Fast Fine-tuning
FPGA	Field-Programmable Gate Array
FFs	Flip-Flops
FPS	Frames per Second
FS	Forecast Skill
GHI	Global Horizontal Irradiance
HLS	High-Level Synthesis
HYTA	Hybrid Thresholding Algorithm
IoT	Internet of Things
LSTM	Long Short-Term Memory
LUT	Lookup Table
ML	Machine Learning
MAE	Mean Absolute Error
MLP	Multilayer Perceptron
MSE	Mean Square Error
NRBR	Normalized Red–Blue Ratio
PM	Persistence Model
PV	Photovoltaic
PTQ	Post-Training Quantization
PS	Processing Subsystem
PL	Programmable Logic
QAT	Quantization-Aware Training
RBD	Red–Blue Difference
RBR	Red–Blue Ratio
RES	Renewable Energy Sources
RMSE	Root Mean Square Error
SG	Smart Grid
SI	Sky Imager
SVR	Support Vector Regression
SoC	System-on-Chip
UTC	Universal Time Coordinated

### References

1. Mehmood, M.Y.; Oad, A.; Abrar, M.; Munir, H.M.; Hasan, S.F.; Muqheet, H.A.u.; Golilarz, N.A. Edge Computing for IoT-Enabled Smart Grid. *Secur. Commun. Netw.* **2021**, *2021*, 5524025. [[CrossRef](#)]
2. Feng, C.; Liu, Y.; Zhang, J. A taxonomical review on recent artificial intelligence applications to PV integration into power grids. *Int. J. Electr. Power Energy Syst.* **2021**, *132*, 107176. [[CrossRef](#)]
3. Zsiborács, H.; Baranyai, N.H.; Vincze, A.; Zentkó, L.; Birkner, Z.; Máté, K.; Pintér, G. Intermittent Renewable Energy Sources: The Role of Energy Storage in the European Power System of 2040. *Electronics* **2019**, *8*, 729. [[CrossRef](#)]

4. Chen, X.; Du, Y.; Lim, E.; Fang, L.; Yan, K. Towards the applicability of solar nowcasting: A practice on predictive PV power ramp-rate control. *Renew. Energy* **2022**, *195*, 147–166. [[CrossRef](#)]
5. Lin, F.; Zhang, Y.; Wang, J. Recent advances in intra-hour solar forecasting: A review of ground-based sky image methods. *Int. J. Forecast.* **2022**. [[CrossRef](#)]
6. Juncklaus Martins, B.; Cerentini, A.; Mantelli, S.L.; Loureiro Chaves, T.Z.; Moreira Branco, N.; von Wangenheim, A.; Rütther, R.; Marian Arrais, J. Systematic review of nowcasting approaches for solar energy production based upon ground-based cloud imaging. *Sol. Energy Adv.* **2022**, *2*, 100019. [[CrossRef](#)]
7. Pickering, T.E. The MMT all-sky camera. In *Ground-Based and Airborne Telescopes*; Stepp, L.M., Ed.; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA, 2006; Volume 6267, pp. 448–454. [[CrossRef](#)]
8. Carreira Pedro, H.; Larson, D.; Coimbra, C. *A Comprehensive Dataset for the Accelerated Development and Benchmarking of Solar Forecasting Methods*; Zenodo: 2019. [[CrossRef](#)]
9. Andreas, A.; Stoffel, T. *REL Solar Radiation Research Laboratory (SRRL): Baseline Measurement System (BMS)*; NREL Report No. DA-5500-56488; NREL: Golden, CO, USA, 2019. [[CrossRef](#)]
10. Véstias, M.P.; Duarte, R.P.; de Sousa, J.T.; Neto, H.C. Moving Deep Learning to the Edge. *Algorithms* **2020**, *13*, 125. [[CrossRef](#)]
11. Sandler, M.; Howard, A.G.; Zhu, M.; Zhmoginov, A.; Chen, L. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. *arXiv* **2018**, arXiv:1801.04381.
12. Iandola, F.N.; Moskewicz, M.W.; Ashraf, K.; Han, S.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *arXiv* **2016**, arXiv:1602.07360.
13. Pedro, H.T.C.; Larson, D.P.; Coimbra, C.F.M. A comprehensive dataset for the accelerated development and benchmarking of solar forecasting methods. *J. Renew. Sustain. Energy* **2019**, *11*, 036102. [[CrossRef](#)]
14. Richardson, W.; Krishnaswami, H.; Vega, R.; Cervantes, M. A Low Cost, Edge Computing, All-Sky Imager for Cloud Tracking and Intra-Hour Irradiance Forecasting. *Sustainability* **2017**, *9*, 482. [[CrossRef](#)]
15. Miller, S.D.; Rogers, M.A.; Haynes, J.M.; Sengupta, M.; Heidinger, A.K. Short-term solar irradiance forecasting via satellite/model coupling. *Sol. Energy* **2018**, *168*, 102–117. [[CrossRef](#)]
16. Ayet, A.; Tandeo, P. Nowcasting solar irradiance using an analog method and geostationary satellite images. *Sol. Energy* **2018**, *164*, 301–315. [[CrossRef](#)]
17. Ordoñez Palacios, L.E.; Bucheli Guerrero, V.; Ordoñez, H. Machine Learning for Solar Resource Assessment Using Satellite Images. *Energies* **2022**, *15*, 3985. [[CrossRef](#)]
18. Kim, B.Y.; Cha, J.W.; Chang, K.H. Twenty-four-hour cloud cover calculation using a ground-based imager with machine learning. *Atmos. Meas. Tech.* **2021**, *14*, 6695–6710. [[CrossRef](#)]
19. Li, Q.; Lu, W.; Yang, J. A Hybrid Thresholding Algorithm for Cloud Detection on Ground-Based Color Images. *J. Atmos. Ocean. Technol.* **2011**, *28*, 1286–1296. [[CrossRef](#)]
20. Rajagukguk, R.A.; Kamil, R.; Lee, H.J. A Deep Learning Model to Forecast Solar Irradiance Using a Sky Camera. *Appl. Sci.* **2021**, *11*, 5049. [[CrossRef](#)]
21. Zuo, H.M.; Qiu, J.; Jia, Y.H.; Wang, Q.; Li, F.F. Ten-minute prediction of solar irradiance based on cloud detection and a long short-term memory (LSTM) model. *Energy Rep.* **2022**, *8*, 5146–5157. [[CrossRef](#)]
22. Chu, Y.; Li, M.; Coimbra, C.F. Sun-tracking imaging system for intra-hour DNI forecasts. *Renew. Energy* **2016**, *96*, 792–799. [[CrossRef](#)]
23. Niccolai, A.; Nespoli, A. Sun Position Identification in Sky Images for Nowcasting Application. *Forecasting* **2020**, *2*, 488–504. [[CrossRef](#)]
24. Paletta, Q.; Lasenby, J. A Temporally Consistent Image-based Sun Tracking Algorithm for Solar Energy Forecasting Applications. In Proceedings of the NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning, Virtual, 11–12 December 2020.
25. Paletta, Q.; Arbod, G.; Lasenby, J. Benchmarking of deep learning irradiance forecasting models from sky images—An in-depth analysis. *Sol. Energy* **2021**, *224*, 855–867. [[CrossRef](#)]
26. Wen, H.; Du, Y.; Chen, X.; Lim, E.; Wen, H.; Jiang, L.; Xiang, W. Deep Learning Based Multistep Solar Forecasting for PV Ramp-Rate Control Using Sky Images. *IEEE Trans. Ind. Inform.* **2021**, *17*, 1397–1406. [[CrossRef](#)]
27. Jiang, H.; Gu, Y.; Xie, Y.; Yang, R.; Zhang, Y. Solar Irradiance Capturing in Cloudy Sky Days—A Convolutional Neural Network Based Image Regression Approach. *IEEE Access* **2020**, *8*, 22235–22248. [[CrossRef](#)]
28. Wang, F.; Zhang, Z.; Chai, H.; Yu, Y.; Lu, X.; Wang, T.; Lin, Y. Deep Learning Based Irradiance Mapping Model for Solar PV Power Forecasting Using Sky Image. In Proceedings of the 2019 IEEE Industry Applications Society Annual Meeting, Baltimore, MD, USA, 29 September–3 October 2019; pp. 1–9. [[CrossRef](#)]
29. Song, S.; Yang, Z.; Goh, H.; Huang, Q.; Li, G. A novel sky image-based solar irradiance nowcasting model with convolutional block attention mechanism. *Energy Rep.* **2022**, *8*, 125–132. [[CrossRef](#)]
30. Paletta, Q.; Hu, A.; Arbod, G.; Lasenby, J. ECLIPSE: Envisioning CLOUD Induced Perturbations in Solar Energy. *Appl. Energy* **2022**, *326*, 119924. [[CrossRef](#)]
31. Tran, Q.K.; Song, S.K. Computer Vision in Precipitation Nowcasting: Applying Image Quality Assessment Metrics for Training Deep Neural Networks. *Atmosphere* **2019**, *10*, 244. [[CrossRef](#)]

32. Le Guen, V.; Thome, N. A Deep Physical Model for Solar Irradiance Forecasting with Fisheye Images. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 2685–2688. [\[CrossRef\]](#)
33. Holmgren, W.F.; Hansen, C.W.; Mikofski, M.A. pvlib python: A python package for modeling solar energy systems. *J. Open Source Softw.* **2018**, *3*, 884. [\[CrossRef\]](#)
34. Xilinx. Zynq UltraScale+ MPSoC. Available online: <https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html> (accessed on 15 October 2022).
35. Xilinx. Please Confirm if This Author Name Is Correct? Vitis AI User Guide UG1414 (v2.5). 2022. Available online: <https://docs.xilinx.com/r/en-US/ug1414-vitis-ai> (accessed on 15 October 2022).
36. Nagel, M.; Baalen, M.V.; Blankevoort, T.; Welling, M. Data-Free Quantization through Weight Equalization and Bias Correction. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1325–1334. [\[CrossRef\]](#)
37. Hubara, I.; Nahshan, Y.; Hanani, Y.; Banner, R.; Soudry, D. Improving Post Training Neural Quantization: Layer-wise Calibration and Integer Programming. *arXiv* **2020**, arXiv:2006.10518. [\[CrossRef\]](#)
38. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556. [\[CrossRef\]](#)
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
40. Naushad, R.; Kaur, T.; Ghaderpour, E. Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study. *Sensors* **2021**, *21*, 8083. [\[CrossRef\]](#) [\[PubMed\]](#)