Senior Electrical Engineering Project 2102499 Year 2024

Database and image preprocessing for solar energy forecasting

Mr. Kanawut Suwandee 6430031021

Mr. Kongpob In-odd 6430015021

Advisor Dr. Suwichaya Suwanwimolkul

Department of Electrical Engineering, Faculty of Engineering
Chulalongkorn University

Academic Year 2024

Advisor signature	Co-advisor signature	Industry representative
______________________________	______________________________	______________________________
(..)	(..)	(..)
Date ______________________	Date ______________________	Date ______________________

ABSTRACT

Solar irradiance is a critical factor in estimating the electric power output of solar cells, which plays a central role in energy management systems. Ensuring system stability requires accurate forecasting of future energy generation. Therefore, solar irradiance forecasting is essential for effective energy planning. This project investigates sky image datasets from Stanford University, the SIRTA research observatory, and Chulalongkorn University. The study involves training and evaluating three deep learning models SUNSET, Unet, and SolarNet for forecasting solar irradiance up to 15 future timestamps (with 1–2 minute resolution), followed by model improvement. The preprocessing pipeline includes fisheye distortion correction and region of interest (RoI) extraction from sky images, followed by cloud mask generation. All three models were trained on the processed SIRTA dataset and additionally trained with a cloud mask channel, specifically for SUNSET. Experimental results demonstrate that in forecasting task the SUNSET model outperforms SolarNet across all three datasets, spanning over one year. Moreover, incorporating the cloud mask channel improves the SUNSET forecast performance by approximately 3% in RMSE from the original, but in terms of MAE, the addition of cloud mask and cloud foreground channels did not lead to a significant improvement. This suggests that the benchmark system needs to be further improved, as the current training and evaluation in this part can handle no more than 6 months of data due to high CPU RAM requirement. In conclusion, deep learning models can effectively forecast solar irradiance using sky imagery and time-series data. Performance can be further enhanced by integrating auxiliary data sources, such as cloud masks, into the model architecture.

Keywords : Solar forecasting, SUNSET model, SolarNet model, Unet model, Cloud masking

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

A microgrid is a small-scale power system that integrates a power generation system, electrical loads, and energy storage. Typically, a microgrid is connected to the main grid. Under normal conditions, a microgrid will manage power generation and consumption appropriately. However, when a microgrid enters islanding mode, a microgrid will rely on its own power generation system. Therefore, solar irradiance forecasting is important for the stability of the system.

Although solar irradiance forecasting requires physical data such as altitude, solar angle, and air mass to develop solar irradiance models [1], today deep learning has been applied to solar irradiance forecasting. The neural networks used in deep learning have the ability to extract relevant features from images such as satellite or sky images [2, 3]. This enables future development based primarily on image data and solar irradiance values, reducing or even eliminating dependence on physical data [4]

Since sky images from different regions have a significant signature due to terrain and atmospheric conditions. As a result, developing dedicated datasets for solar irradiance forecasting in Thailand is essential. In the past, there are two senior projects, [5], that implemented a sky image data storage system, and [6], which applied different deep learning methods in forecasting. However, these previous studies did not attempt to compare other datasets, nor considered image pre-processing techniques such as edge cropping and lens correction, which could improve the prediction accuracy [7]. In addition, they did not extract additional information that could improve the effectiveness of neural networks, such as [8] by using cloud mask, cloud foreground, and cloud coverage. *Table 1.1* shows the difference between these projects and ours.

1.2 Objectives

1. Analyze and examine the characteristics of three different datasets:
   - CUEE [5] from Chulalongkorn University
   - SKIPP'D [9] from Stanford University
   - SIRTA [10] from SIRTA research observatory.
2. To evaluate the performance of deep learning-based solar irradiance forecasting, SUNSET [11], Solarnet [4], and Unet [12], on the three datasets.

Academic projects	Senior Project [5]	Senior Project [6]	Our senior project
Built database	YES	NO	NO
Datasets	CUEE [5] - -	CUEE [6] - -	CUEE [5] SKIPP'D, USA [9] SIRTA, France [10]
Sky image enhancing	- -	- -	RoI extraction Len distortion correction
Sky image extraction	Clear-sky irradiance - -	- - -	Clear-sky irradiance Cloud extraction Cloud marking
Solar irradiance forecasting	SUNSET [11]	SUNSET [11], Unet [12]	SUNSET [11], Solarnet [4], Unet [12]

Table 1.1: Comparison between academic projects

3. Process and extract additional features, *e.g.*, cloud volume, and cloud mask.

4. Gauge the impact of additional features from sky images on the SIRTA dataset and record any challenges occurred during building the benchmarking system.

1.3 Scope of work

1. This study aims to examine the key characteristic of the tree datasets such as CUEE, SKIPP'D, and SIRTA

2. This study aims to develop a short-term solar irradiance forecasting for 15 minutes ahead using neural networks.

1.4 Expected outcomes

1. A survey of the standard sky image databases.

2. A system to train and benchmark the three neural networks on the three datasets.

3. An implementation for image preprocessing: RoI extraction and lens distortion correction.

4. Performance comparison on SUNSET, Solarnet, and Unet on SIRTA datasets with the preprocessed sky images.

5. An implementation of the baseline and *a new method** for extracting cloud volume and cloud masking.

6. Performance of SUNSET on SIRTA datasets with additional features: cloud volume and cloud masking.

$*$ denotes an extra milestone achieved in this project.

1.5 Engineering knowledge

1. Deep learning and neural networks.
2. Image processing.
3. Time-series analysis.
4. Object-oriented programming

Chapter 2

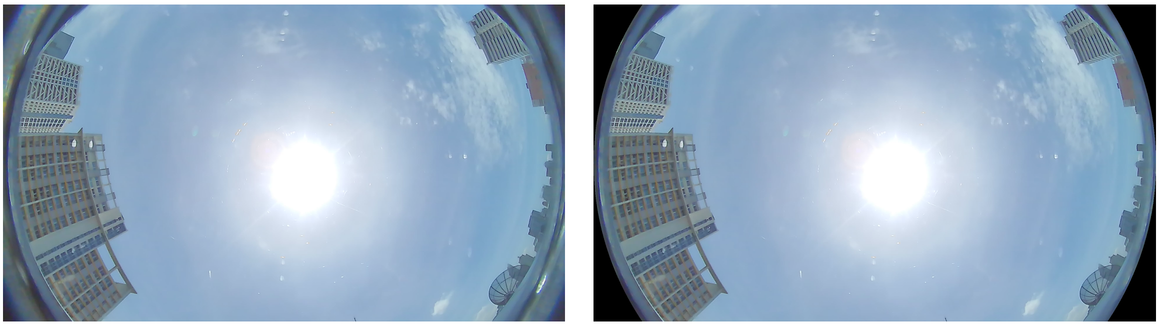
Literature review

This chapter will provide foundations to our proposed method. The topics in this chapter include RoI selection, lens distortion correction, a sky condition classification, cloud cover extraction, and the neural networks considered in this work.

- A fish eye lens is a type of lens that has a very short focal length, so the image from this lens would be circular and can capture a lot of surrounding information compared to other types of lenses, such as wide-angle lenses and standard DSLR lenses. The resulting images often include unwanted objects. Therefore, it is necessary to crop the edges to keep only the region of interest (RoI). Additionally, the objects in the image are distorted due to the short focal length. According to photography principles, these distortions should be corrected. The RoI selection and distortion correction will be discussed in section 2.1 and section 2.2.
- When using a fish-eye lens to capture sky images, cloud density and sun position are key factors in forecasting solar irradiance. Cloud coverage from sky images can be an indicator of sky conditions under different climates, which is related to solar irradiance. This chapter will explore a classification model using solar irradiance in section 2.3 as well as cloud cover extraction in section 2.4.
- Furthermore, the neural network foundation and the benchmarking architectures for solar irradiance forecasting, Solarnet, U-Net, and SUNSET, will be presented in section 2.5 and section 2.7, respectively. Finally, we will explore the potential of using a generative-based neural network such as a Generative Omnimatte to extract the cloud mist and the cloud mask in section 2.8.

2.1 Region of interest

Region of interest (RoI) is a specific part of an image that we want to carve out. Extracting an RoI can be done by defining a boundary around the desired area, such as a circular or rectangular frame, to separate the target region from the rest of the image. The extracted portion can then be further processed according to the method [13]. *Figure 2.1* provides an example of resulting images contain only the area within a pre-defined frame, while the outer regions, such as the edges, are filtered out.



(a) Before ROI extraction

(b) After ROI extraction

Figure 2.1: Example of sky images before and after ROI extraction

2.2 Lens distortion correction

Capturing images with a fisheye lens results in barrel distortion, causing data at the edges of the image to be compressed and appear smaller than their actual size. However, data near the center of the image are enlarged beyond its true proportions. This distortion negatively impacts image processing for accurately estimating cloud coverage. Therefore, a correction method from [7] is applied, which establishes a relationship between the real image and its corresponding representation in polar coordinates. This relationship is given by the following equation.

$$\begin{pmatrix} \rho\cos\theta \\ \rho\sin\theta \\ h \end{pmatrix} = \gamma(s)\left[\frac{2}{R_m}\begin{pmatrix} s\cos\theta \\ s\sin\theta \\ \sqrt{R_m^2 - s^2} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} s\cos\theta \\ s\sin\theta \\ \sqrt{R_m^2 - s^2} \end{pmatrix}\right] \tag{2.1}$$

The left-hand side of the above equation corresponds to the object in real-world (3D coordinates), where ρ is the real-world radial distance and h is the real-world height. Meanwhile, the right-hand side of the above equation corresponds to the object appearing in an image (2D coordinates). s is the apparent radial distance in the image. R is the radius of the fisheye lens. $\gamma(s)$ is the distance between the fisheye lens center and a given point, defined as $\gamma(s) = \|X_c - P\|$, where X_c represents the position of the lens.

In practice, we only adjust the value of $\tilde{\rho}$, which represents the field of view of the image we are interested in. It can be calculated using the following equation.

$$\tilde{\rho} = \frac{\rho}{h} = \frac{2s}{2\sqrt{{R_m}^2 - s^2} - R_m} \tag{2.2}$$

2.3 Sky condition classification with solar irradiance

Research [14] classifies cloud coverage levels into three categories: **Clear** representing a clear sky; **Partly cloudy** represents partial cloud coverage; and **Cloudy** represents significant cloud coverage. This classification is based on the daily average clear sky index, which indicates the impact of actual sky conditions, *e.g.*, clouds, and haze on solar irradiance. The clear sky index can be calculated using the following equation:

$$k = \frac{I}{I_{clr}}, \tag{2.3}$$

where I is measured solar irradiance, in units of Watt$/m^2$, and I_{clr} is clear sky irradiance. Clear sky irradiance is the amount of solar radiation that reaches the Earth's surface under clear, cloudless, and aerosol-free conditions, which are essential for calculating the clear sky index. However, measuring solar irradiance with instruments such as pyranometers may not always reflect true clear sky conditions due to interference from factors such as aerosols, smoke, and atmospheric moisture. Therefore, the work [14] used a calculation model developed by [15]

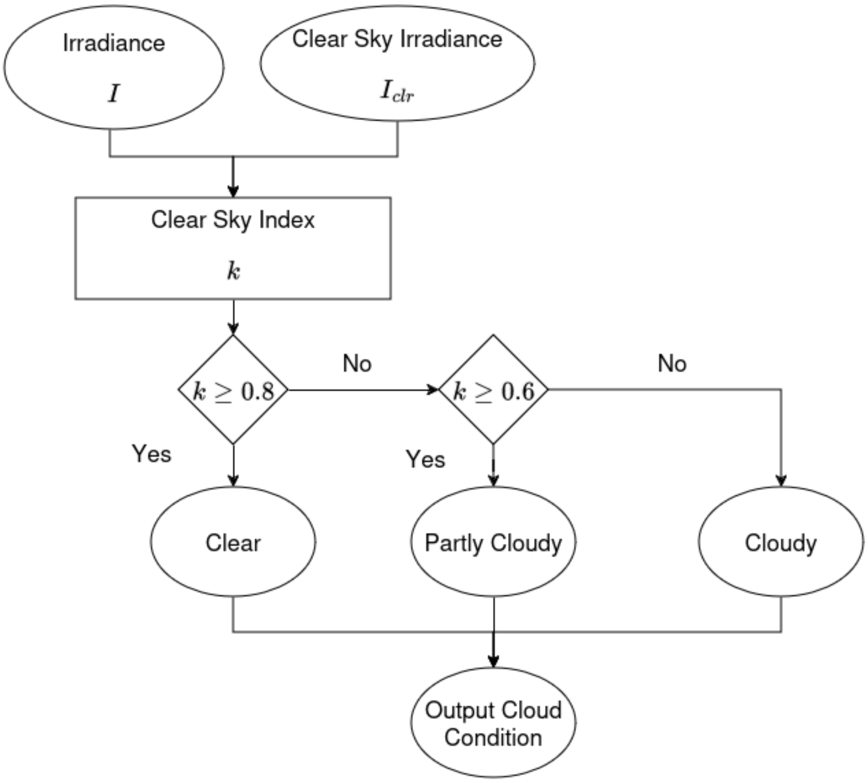


Figure 2.2: Flowchart representing cloud condition classification using clear sky index

2.4 Cloud cover extraction from sky images

The cloud cover extraction proposed in [16] determines cloud cover from sky images. Due to the similarity between the sun and the nearby cloud that can appear as white in RGB images, the method will first track the sun's location and remove the effect of the sun. Then, the normalized red-blue ratio (NRBR) is used to calculate the clarity of the sky. The clear sky library (CSL) is then used as a mechanism to compare the clarity of the sky with the clear-sky days in the database to justify the cloudiness level.

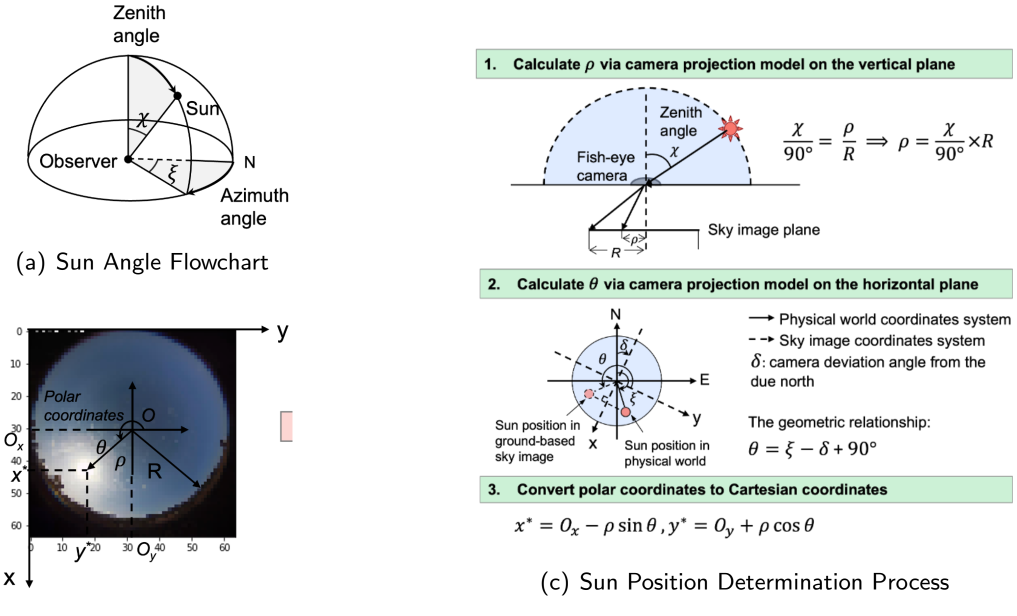
2.4.1 Tracking sun position

Accurate sun positioning enhances cloud extraction from sky images. This is achieved using principles and methods from [16], which involve identifying the sun's position in polar coordinates and then converting it to Cartesian coordinates via linear relationships, as defined by the equations:

$$x^* = O_x - \rho \sin \theta \tag{2.4}$$

$$y^* = O_y + \rho \cos\theta \tag{2.5}$$

where x^* is the position of the sun along the x-axis in the image (b). y^* is the position of the sun along the y-axis in image (b). O is center of the image from the camera. ρ is the distance between the center of the image from the camera and the position of the sun in the image. $\theta = \xi - \delta + 90°$ where ξ, δ are functions derived from Da Rosa's observations [17].


Zenith angle
Sun
χ
Observer
ξ
N
Azimuth angle
Polar coordinates
O
θ
ρ
R
y
x
O_x
x*
y*
O_y
0
10
20
30
40
50
60
1. Calculate ρ via camera projection model on the vertical plane
Zenith angle
χ
Fish-eye camera
Sky image plane
R
ρ
χ/90° = ρ/R ⇒ ρ = χ/90° ×R
2. Calculate θ via camera projection model on the horizontal plane
N
E
δ
θ
ξ
y
x
Sun position in ground-based sky image
Sun position in physical world
Physical world coordinates system
Sky image coordinates system
δ: camera deviation angle from the due north
The geometric relationship:
θ = ξ − δ + 90°
3. Convert polar coordinates to Cartesian coordinates
x* = O_x − ρ sin θ , y* = O_y + ρ cos θ

(a) Sun Angle Flowchart

(b) Sky image coordinates

(c) Sun Position Determination Process

Figure 2.3: Method for sun position calculation (source: [16] image by: Yuhao Nie)

2.4.2 Cloud cover extraction

Since the irradiance values we obtain are directly affected by the amount or density of clouds in the sky, the method [16] utilizes and leverages the concept of NRBR [18] and CSL [16] as follows:

- The normalized red-blue ratio (NRBR) [18] is the ratio of the difference between R and B, which are the pixel values in the red and blue channels, respectively. NRBR can be calculated as follows:

$$\text{NRBR} = \frac{B - R}{B + R} \tag{2.6}$$

  The clear sky appears blue with high NRBR values, whereas the cloud appears white or gray with low NRBR values. This method works well when the sun is completely obscured by clouds. However, the results are inaccurate when the sky is clear, or the sun is partially obscured because it cannot distinguish between the sun and clouds.

- Clear sky library (CSL) [16] compares the NRBR values between the day of interest and a clear sky day. The NRBR values of clear sky days are stored in a database beforehand. CSL detects clouds when the NRBR values of the two images differ by more than a specified threshold. This helps reduce the problem of detecting the sun in images without cloud cover. However, it encounters issues in images where clouds obscure the sun, as it cannot detect clouds that are directly over the sun.

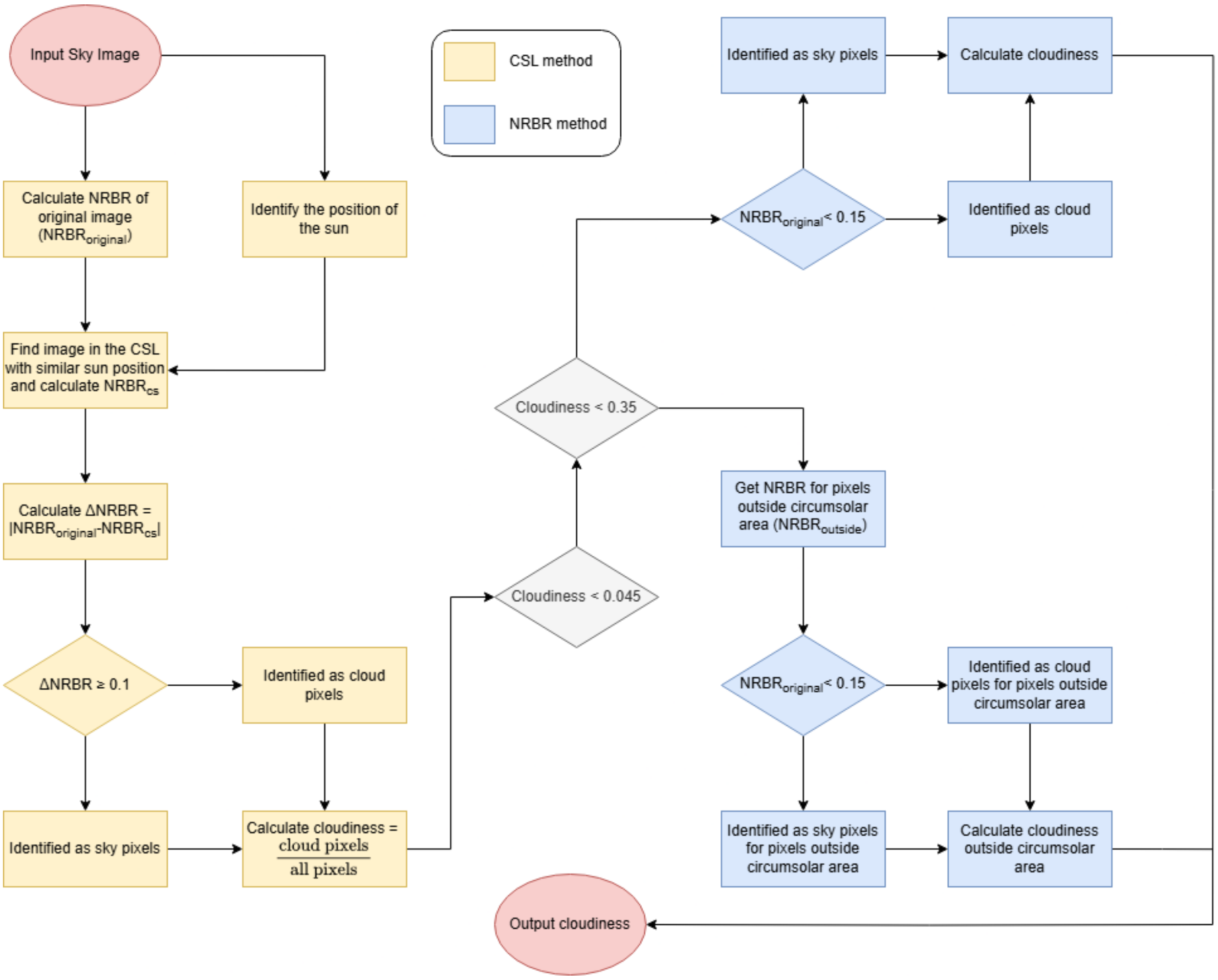

Input Sky Image
CSL method
NRBR method
Identified as sky pixels
Calculate cloudiness
Calculate NRBR of original image (NRBR$_{original}$)
Identify the position of the sun
NRBR$_{original}$ < 0.15
Identified as cloud pixels
Find image in the CSL with similar sun position and calculate NRBR$_{cs}$
Cloudiness < 0.35
Get NRBR for pixels outside circumsolar area (NRBR$_{outside}$)
Calculate ΔNRBR = |NRBR$_{original}$-NRBR$_{cs}$|
Cloudiness < 0.045
ΔNRBR ≥ 0.1
Identified as cloud pixels
NRBR$_{original}$ < 0.15
Identified as cloud pixels for pixels outside circumsolar area
Identified as sky pixels
Calculate cloudiness = $\frac{\text{cloud pixels}}{\text{all pixels}}$
Identified as sky pixels for pixels outside circumsolar area
Calculate cloudiness outside circumsolar area
Output cloudiness

Figure 2.4: Cloud cover extraction method proposed in [16]

2.5 Solar irradiance forecasting with deep learning

In the application of solar irradiance forecasting using deep learning, the neural network is viewed as a function of parameters. This model takes sky images as input and provides solar irradiance values as output, which can be represented by the following equation:

$$y = f(x; \theta) \tag{2.7}$$

where x is sky images inputs; y is solar irradiance input; and $f(\cdot; \theta)$ is a function of parameters that represents the neural network, where θ represents the set of all parameters in that network.

Generally, solar irradiance forecasting using deep learning consists of two components: (i) the learning phase, where a deep neural network is trained for solar irradiance forecasting and (ii) the inference phase, where the trained neural network is deployed for solar irradiance forecasting. The training of the neural network dictates the approach used for solar irradiance forecasting. Meanwhile, the neural network architecture is modified to achieve results consistent with the learning and forecasting objectives.

2.6 Learning for solar irradiance forecasting

The learning for solar irradiance forecasting can be formulated as an optimization problem to find the optimal parameters θ. Let loss$(\cdot,\cdot)$ be the function that measures the loss between the predicted values from $f(\cdot;\theta)$ and the reference solar irradiance (denoted as y). The optimal parameters are found by dividing the data into batches. If each batch contains b data samples (x_i, y_i), the optimization problem for finding the optimal parameters θ can be written as:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{b} \sum_{i=1}^{b} \text{loss}(f(x_i;\theta), y_i) \tag{2.8}$$

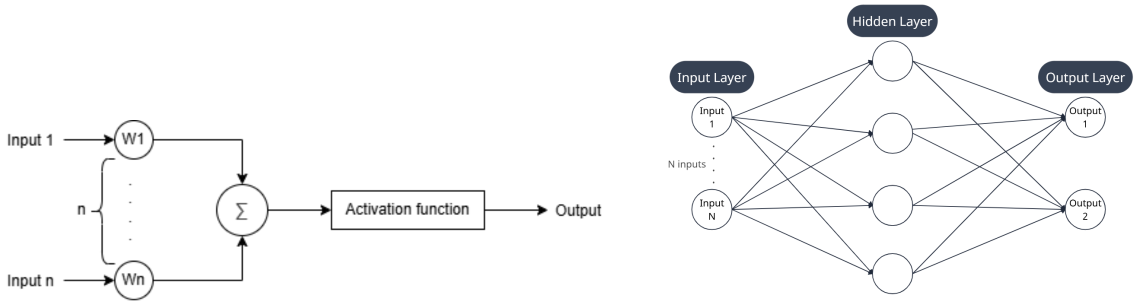
Generally, two forms of loss$(\cdot,\cdot)$ functions are used to measure the loss in irradiance forecasting :

- Absolute error loss (AE), where loss$(\hat{y}, y) = |\hat{y} - y|$
- Square error loss (SE), where loss$(\hat{y}, y) = (\hat{y} - y)^2$

The loss$(\cdot,\cdot)$ function corresponds to the method used to measure solar irradiance forecast performance (as in eq. (4.1) and eq. (4.2) in section 4.3). Furthermore, as seen in eq. (2.8), training a deep neural network for forecasting requires creating a database with (i) reference irradiance data and (ii) sky images for deep network training. However, recent research has been explored using other reference data, such as cloud detection masks [8] and future cloud movement [12], or adding input data such as satellite images [3], to provide the neural network with more comprehensive spatial information.

2.7 Solar irradiance inference

This section will discuss three neural network architectures, namely SUNSET, Unet, and Solarnet, which have been used as benchmarks in recent research articles [2, 8, 19, 12, 20]. These neural networks share some similar components: input layer, hidden layer, and output layer, which is structured according to human information processing and can be adapted and applied to various operational scenarios. Typically, each layer consists of processing units, or neurons, which are interconnected to form the neural network, as shown in *Figure 2.5*.



(a) Neuron

(b) Neural network

Figure 2.5: Neuron and neural network

Generally, each neuron has the following components:

- **Input layer.** The layer receives the input data into the neural network for learning and inference.

- **Weight.** The weight in hidden layers is the primary set of parameters to derive semantic information from input data. This information is passed through weights in each layer of the neural network, forming latent features.
- **Bias.** The bias in hidden layers are parameters that help shift the weighted sum, often included to help the activation function work more easily.
- **Dropout.** The dropout layer determines whether to activate this neuron or not, randomly based on a set probability of activation.
- **Activation function.** The activation function is a non-linear function that maps the latent feature to the output values. It is typically used to constrain the output values to a suitable range (based on the task), such as values between 0 and 1.

2.7.1 SUNSET

SUNSET [11] is a neural network with a convolutional neural network (CNN) architecture specifically designed to use multiple consecutive convolution layers for extracting two-dimensional information, as shown in *Figure 2.6*. SUNSET takes a set of images as sequential input and passes them to pooling layers that are connected consecutively. Subsequently, it is forwarded to fully connected layers to perform the prediction of the irradiance value.

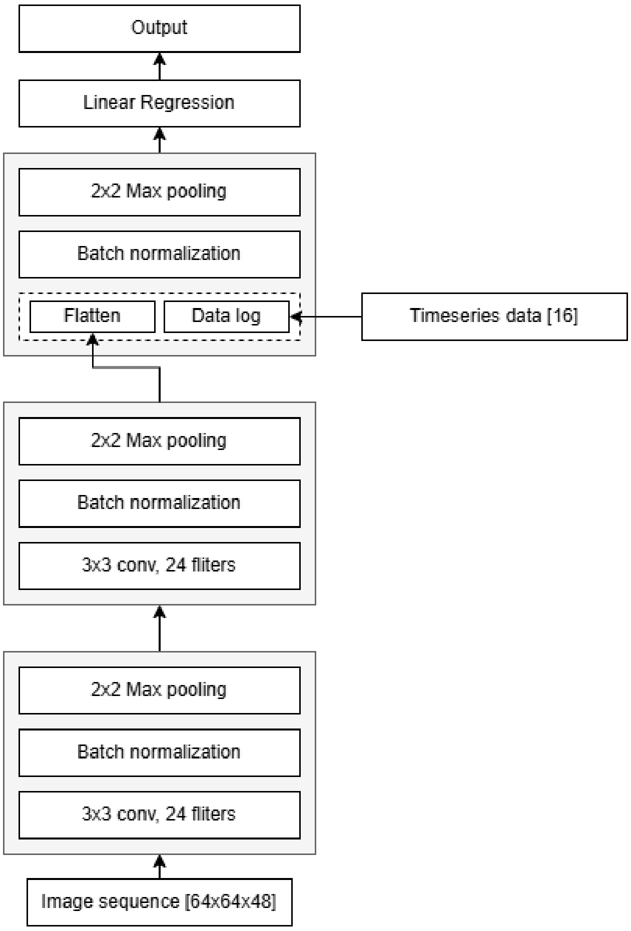


Figure 2.6: SUNSET network architecture (source: [11] image by: Yuchi Sun)

2.7.2 Unet

Unet is a neural network composed of an encoder, bottleneck, and decoder as shown in *Figure 2.7*. An **encoder** is responsible for extracting information from the input image. It uses Convolutional layers and Pooling layers to extract features from the image. This information is then passed on to an encoder with lower resolution, and also to the decoder at the same resolution level through skip connections. **Bottleneck** acts as an intermediary between the encoder and the decoder to transmit information. Internally, it consists of convolutional layers and an activation function. A **decoder** takes the encoder features that have been encoded and transforms them back into an image and into the desired output. At each step, the resolution of the image is increased.

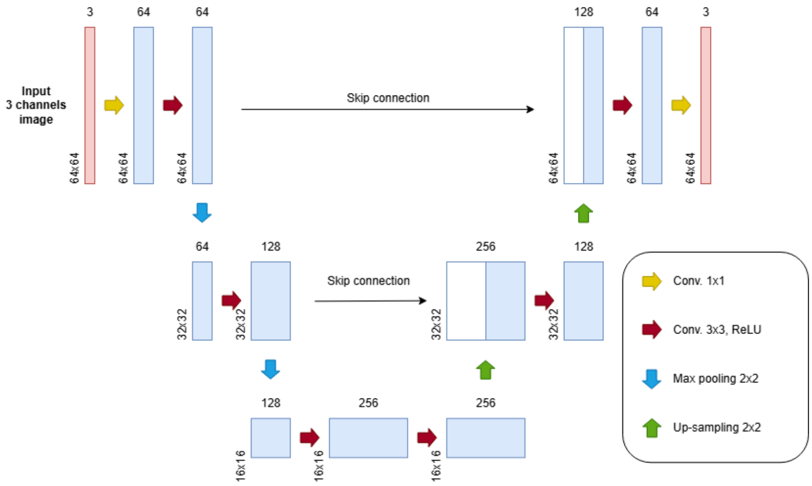


Figure 2.7: Unet architecture [12]

2.7.3 Solarnet

Solarnet is a neural network architecture developed based on VGG16, made up of a total of 20 sequential layers as shown in *Figure 2.8*. These layers include 13 Convolutional layers, 5 Pooling layers, and 2 remaining dense layers that conclude with a Linear activation function. In the original VGG16 architecture, the final three layers are dense layers followed by a ReLU activation function. Also, Solarnet only accepts images as input and does not take solar irradiance values or other feature data as additional input.

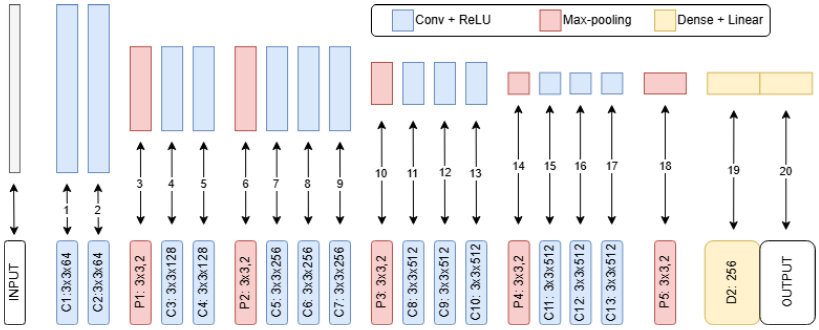


Figure 2.8: VGG16 architecture [4]

2.8 Generative omnimatte

Omnimatte [21] is a task that aims to decompose video input into matte layers that contain individual objects along with their associated effects, such as shadows and reflections. To lay the foundation of an omnimatte task, we include the meaning of the following important variables:

Suppose that an input video consists of n objects that can be decomposed into n matte layers. For $i = 1, ..., n$ ...

- $I_{i,fg}$ is the i^{th} matte layer consisting of each foreground object (and its associated effects);
- α_i is the object mask of the i^{th} matte layer; and
- I_{bg} is the clean plate background;

Omnimatte neural network is Unet-based neural network, that is used to perform the omnimatte task, where, originally, the training of omnimatte requires a static background and depth estimation [21]. However, a recent work [22] proposed a generative omnimatte that circumvents these requirements by using Casper, a text-to-image generation to create static background and objects. This supports the removal of scene effects caused by specific objects, as shown in *Figure 2.9*.

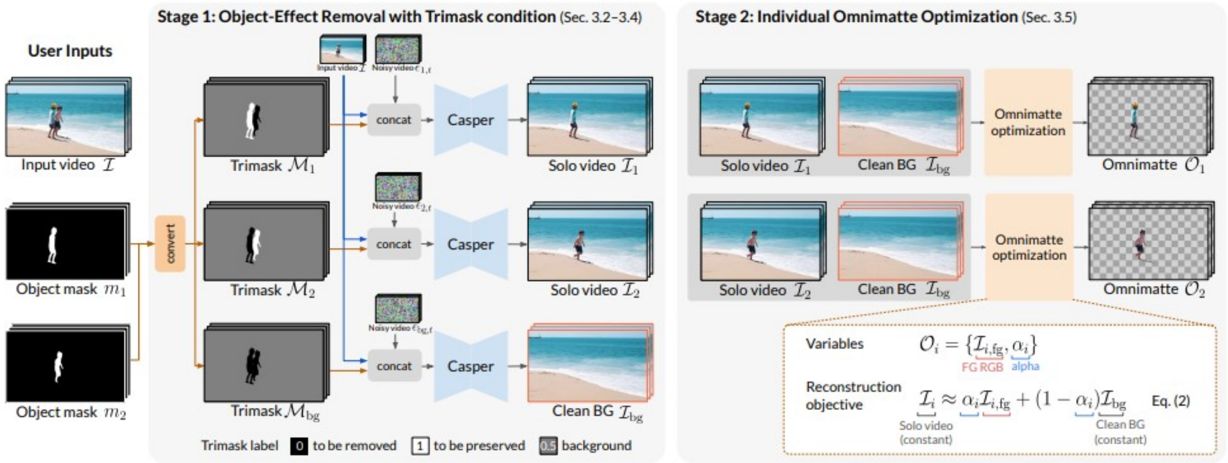

Stage 1: Object-Effect Removal with Trimask condition (Sec. 3.2–3.4)
User Inputs
Input video $\mathcal{I}$
Object mask m_1
Object mask m_2
convert
Input video $\mathcal{I}$
Noisy video $\epsilon_{1,t}$
Noisy video $\epsilon_{2,t}$
Noisy video $\epsilon_{bg,t}$
Trimask $\mathcal{M}_1$
Trimask $\mathcal{M}_2$
Trimask $\mathcal{M}_{bg}$
concat
Casper
Solo video $\mathcal{I}_1$
Solo video $\mathcal{I}_2$
Clean BG $\mathcal{I}_{bg}$
Trimask label
0 to be removed
1 to be preserved
0.5 background
Stage 2: Individual Omnimatte Optimization (Sec. 3.5)
Solo video $\mathcal{I}_1$
Clean BG $\mathcal{I}_{bg}$
Omnimatte optimization
Omnimatte $\mathcal{O}_1$
Solo video $\mathcal{I}_2$
Clean BG $\mathcal{I}_{bg}$
Omnimatte optimization
Omnimatte $\mathcal{O}_2$
Variables
$\mathcal{O}_i = \{\underbrace{\mathcal{I}_{i,fg}}_{\text{FG RGB}}, \underbrace{\alpha_i}_{\text{alpha}}\}$
Reconstruction objective
$\underbrace{\mathcal{I}_i}_{\text{Solo video (constant)}} \approx \underbrace{\alpha_i}\underbrace{\mathcal{I}_{i,fg}} + (1 - \underbrace{\alpha_i})\underbrace{\mathcal{I}_{bg}}_{\text{Clean BG (constant)}}$
Eq. (2)

Figure 2.9: Generative omnimatte framework (source: [22] image by: Lee, Yao-Chih and Lu)

2.9 Learning for omnimatte task

The generative omnimate neural network aims to produce the following output:

$$O_i := \{I_{i,fg}, \alpha_i\}, \tag{2.9}$$

which is a set of the foreground object $I_{i,fg}$ and the object mask α_i. Then, the omnimatte neural network will be trained to learn the following composition of foreground layer, mask layer, and background layer, to resemble the solo video I_i [22]:

$$Comp(I_{bg}, I_{i,fg}, \alpha_i) = \alpha_i I_{i,fg} + (1 - \alpha) I_{bg} \;\approx\; I_i, \tag{2.10}$$

To achieve the learning objective, the learning loss is the combination of the loss functions:

$$L_{recon} + \lambda_{sparsity} L_{sparsity} + \lambda_{mask} L_{mask}. \tag{2.11}$$

Each of these loss terms is defined as follows.

- Reconstruction loss (L_{recon}) is used to enforce the resemblance between the solo video and the composition of the foreground layer, the mask layer and the background layer:

$$L_{recon} = ||I_i - Comp(I_{bg}, I_i, \alpha_i)||_2 \tag{2.12}$$

- Sparsity loss ($L_{sparsity}$), which is used to prevent the object mask from being overly sensitive to the reconstruction error:

$$L_{sparsity} = \beta_1 ||\alpha_i||_1 + \beta_0 \Phi_0(\alpha_i) \tag{2.13}$$

- Mask loss L_{mask}, which is used to guide the object mask α_i to match the ground truth mask m_i:

$$L_{mask} = ||m_i - \alpha_i||_2 \tag{2.14}$$

Chapter 3

Methodology

We proceed with the following steps to achieve all the milestones in this project:

- **Data exploratory** on SUNSET, Solarnet, Unet datasets
- **Image processing techniques**, region of interest (RoI) extraction and lens distortion correction, to improve the quality of sky images
- **Cloud mask and cloud foreground extraction using a baseline method**. The method [16] is used to provide the cloud mask, which is then used to select the cloud foreground.
- **Cloud mask and cloud foreground extraction using a generative omnimatte**. The settings and training of generative omnimatte are modified for cloud mask and cloud foreground extraction.
- **Training with baseline settings**. We train SUNSET, Unet, and Solarnet for solar irradiance forecasting on CUEE, SIRTA, and SKIPP'D datasets.
- **Training with image processing techniques**. We train SUNSET, Unet, and Solarnet using SIRTA data that have been processed with RoI and lens distortion correction.
- **Training with cloud mask and cloud foreground**. Train SUNSET using the SIRTA data with cloud mask and cloud foreground data.

3.1 Data exploratory

This study focuses on three datasets, namely CUEE, SKIPP'D, and SIRTA, as shown in *Table 3.1*

- CUEE contact project owner [5]
- SKIPP'D dowload from `https://purl.stanford.edu/dj417rh1007`
- SIRTA authorized via `https://sirta.ipsl.fr/data-request/`

The explored information on three datasets is provided in *Table 3.1*.

Details	CUEE	SKIPP'D	SIRTA
research center	Chulalongkorn University	Stanford University	SIRTA
Location	Bangkok	Stanford	Palaiseau
Country	Thailand	USA	France
Input data	Sky image	Sky image	Sky image
Reference data	Solar irradiance	Electrical power	Solar irradiance
Data collection period	1/2023 -11/2023	03/2017 - 10/2019	01/2012 - 11/2024
Data study period	1/2023 -11/2023	03/2017 - 10/2019	01/2023 - 12/2023
Number of data	82,327	135,527	102,846
Data splitting	60:20:20	84:9:7	60:20:20
Resolution	1 minute	1 minute	2 minutes
Input data	previous 15 sample	previous 15 sample	previous 15 sample
forecasting data	15 minutes ahead	15 minutes ahead	30 minutes ahead

Table 3.1: Database detail

3.2 Image processing to improve the quality of sky images

Region of interest (RoI) and lens distortion correction, the image processing techniques discussed in section 2.1 and section 2.2, are used to improve image quality, which are the inputs to solar irradiance forecasting.

- An example result from RoI extraction on SIRTA data is shown in *Figure 3.1*.

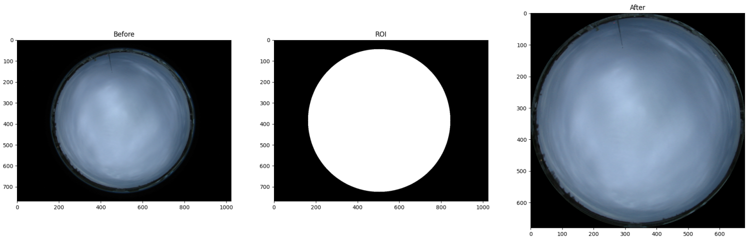


Figure 3.1: Example of RoI image (SIRTA).

- Fisheye lens distortion correction on the SKIPP'D and SIRTA data using different $\widetilde{\rho}$ values. An example result is shown in *Figure 3.2*

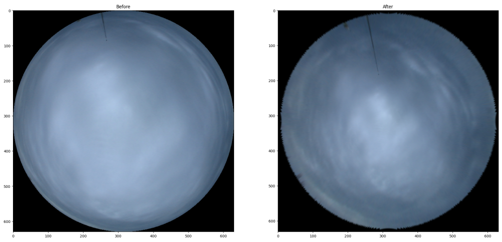


Figure 3.2: Example of distortion corrected image (SIRTA).

The results of solar irradiance forecasting on the three datasets where the sky images that have been corrected by applying RoI and lens distortion correction are provided in section 4.3.2 .

3.3 Cloud mask and foreground extraction

This study performed the cloud mask and cloud foreground extraction on SIRTA sky images to improve the performance of the model. We explored the following approaches:

- Cloud mask extraction from the baseline method [16];
- Cloud foreground extraction by extending [16];
- Cloud extraction using generative omnimatte neural network [22];
- Cloud foreground extraction using generative omnimatte neural network [22].

3.3.1 Cloud mask extraction with the baseline method

Cloud mask extraction [16] is a deterministic method that performs cloud mask extraction via color thresholding NRBR and sky image database comparison CSL. *Figure 3.3* provides an example result from the cloud mask extraction. The parameters used to extract RoI and masking cloud algorithm (discussed in section 2.4.2) on the SIRTA dataset where the image resolution was 64×64 is provided in *Table 3.2*. The minimum NEBR threshold of 0.15 is used for both $\text{NRBR}_{\text{outside}}$ and $\text{NRBR}_{\text{original}}$.

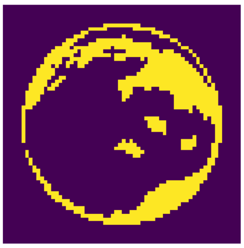


Figure 3.3: Example of cloud mask extracted by [16]

Parameters	Description	SIRTA
Origin X	The center of the image along the x-axis.	31.5
Origin Y	The center of the image along the y-axis.	31.5
Radius	The distance from the center to the edge of the lens within the image.	27

Table 3.2: Cloud masking parameters setting on SIRTA dataset

3.3.2 Cloud foreground extraction by extending [16]

The method [16] is extended to extract the cloud foreground, which could be useful in deriving the solar irradiance forecast. The extension involves multiplying the sky image with the cloud masking channel and then applying it on top of the clear-sky image. The result is shown in *Figure 3.4*.

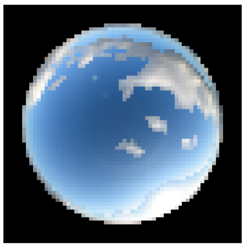


Figure 3.4: A sample of overlaying cloud foreground on a clear-sky image

3.4 Cloud mask and foreground extraction using generative omnimatte

Cloud cover extraction in section 2.4.2 provides a hard decision for cloud mask $\{0, 1\}$. However, naturally clouds have a thickness and can form a mist shape in sky images. In this section, our aim is to extend the binary could mask as a grayscale output $[0, 1]$.

To achieve this soft decision, we explore the generative Omnimatte approach [22] to decompose the sky image into layers of cloud, sky, and sun. The generative omnimatte model along with its associated loss functions is adjusted to extract the cloud from sky images. Specifically, an input sky image is seen as a composition of cloud foreground, soft decision mask, and a clear-sky image. We assign the notation to each component as follows:

- I_{fg} is cloud foreground;
- α is a soft decision mask;
- I_{bg} is a clear-sky image.

3.4.1 Learning phase

To achieve our specific output, we further adjusted certain aspects of their original model architecture and loss function. We adjust the losses from eq. (2.11) including L_{recon}, L_{mask}, $L_{sparsity}$ along with their respective weighting factors λ_{mask}, $\lambda_{sparsity}$.

L_{recon} serves as the main loss function to guide the neural network in learning the cloud coverage. Specifically, L_{recon} is implemented as a mean squared error (MSE) loss calculated between two components: the reconstructed image I_{recon} and the original sky image.

$$I_{recon} \approx I = I_{cloud} + \underbrace{(1 - \alpha) I_{bg}}_{I_{masked,bg}} \tag{3.1}$$

where I_{cloud} provides the new design option when generating the reconstructed image:

- **Comb-off option.** In this option, we follow the traditional foreground image definition, which is the cloud foreground multiplied by the predicted soft-decision mask α_i. :

  $$I_{cloud} = \alpha_i I_{fg} \tag{3.2}$$

  Figure 3.5 provides an example of how I_{recon} can be derived from α_i, I_{fg}, I_{bg} in our settings.

- **Comb-on option.** The cloud foreground I_{fg} is first combined with the clear sky image I_{bg}, and then the result is multiplied by the predicted soft-decision mask α_i:

$$I_{cloud} = \alpha_i(I_{fg} + I_{bg}) \tag{3.3}$$

Our learning still includes L_{mask} and $L_{sparsity}$. These loss functions are necessary in our context in the following aspects:

- L_{mask} is used to supervise the prediction of cloud mask, encouraging it to align closely with the ground-truth mask from cloud detection method [16]. We use same formula eq. (2.14).
- $L_{sparsity}$ is loss that encourages the predicted cloud mask to be sparse. This loss is important because it helps reduce the reliance on the ground-truth binary mask that cannot capture the mist and tends to overly simplify the captured cloud. $L_{sparsity}$ encourages the network outputs to have high values in some regions where clouds are actually present and remain close to zero elsewhere. The formula is the same as eq. (2.13).

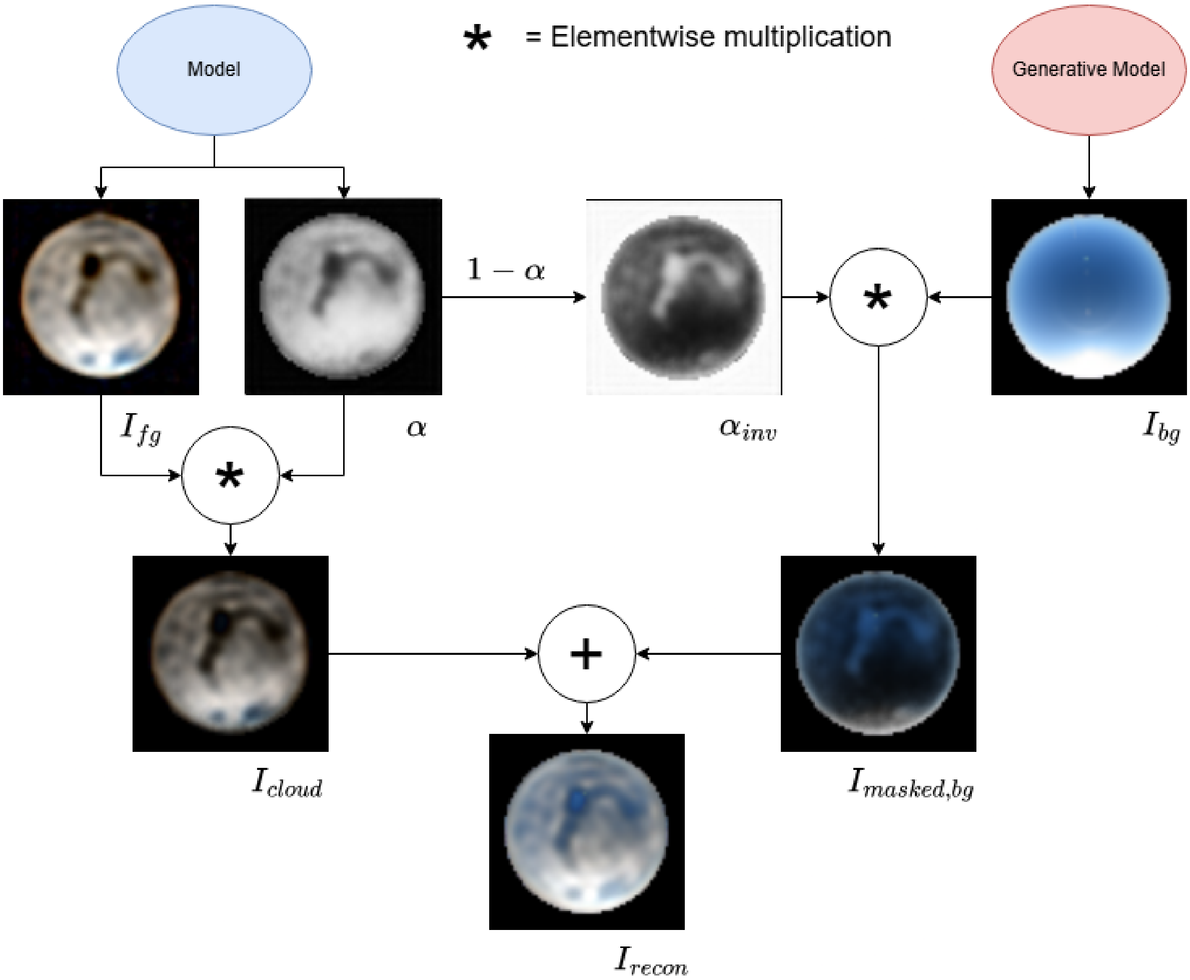


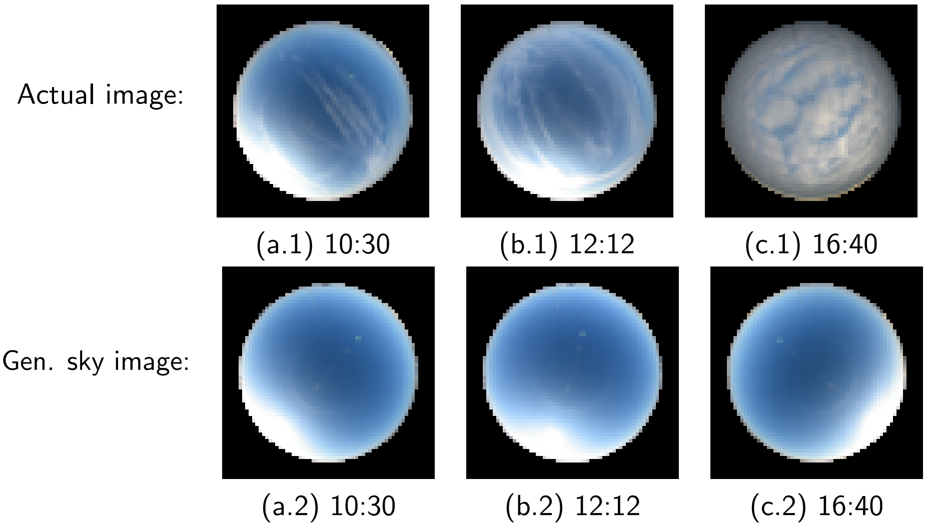
Figure 3.5: Flowchart representing reconstructed image process (option comb-off)

3.4.2 Clear-sky image generation

The clear-sky image I_{bg} is generated using a text-to-image generation [23], which is an extension of a denoising diffusion probabilistic model [24] to receive the input text as the condition for image generation. To generate a sky images, we provide the following text inputs to the model, which is

$$\{\text{hour}, \text{minute}, \text{day of year}, \text{cloud coverage}, \text{clear sky irradiance}, \text{solar irradiance}\} \tag{3.4}$$

where cloud coverage is obtained from [25], which is the cloud coverage estimated using Elifan algorithm [26]. The clear sky irradiance is obtained from the Ineichen and Perez clear sky model in terms of the Linke turbidity [27, 28]. The solar irradiance is obtained from the SIRTA dataset [10]. During deployment, cloud coverage is set to 5% and the solar irradiance is set to clear sky irradiance $+20$. *Figure 3.6* provide the example of actual sky images and generated clear sky images.



(a.1) 10:30 (b.1) 12:12 (c.1) 16:40

(a.2) 10:30 (b.2) 12:12 (c.2) 16:40

Figure 3.6: Examples of actual sky images and generated clear sky images on 29 Jan 2023 at 10:00 (a.1)-(a.2); 12:00 (b.1)-(b.2) ; and 16:00 (c.1)-(c.2).

3.5 Hyperparameter settings for generative omnimatte

We follow the parameter tuning steps guided by generative omnimate [22]. The training process began with a **base model**, which was optimized to provide the prediction of soft-decision mask α. This is achieved using only the mask loss L_{mask} and sparsity loss $L_{sparsity}$. Once the network achieved stable performance in learning the cloud masks, a **transferred learning** is performed by employing the reconstruction loss L_{recon} in addition to the previous two losses. *Table 3.3* provides the settings for each training.

3.6 Hyperparameter settings for solar irradiance forecasting

This study trains SUNSET, Unet, and Solarnet on 3 datasets such as CUEE, SKIPP'D, and SIRTA to find the best parameter setting as follows:

- **Learning rate.** the step size that determines how much the neural network's weights are adjusted during training.

Parameters	**Base Model**	**Transferred Model**
Loss function	L_{mask}, $L_{sparsity}$	L_{recon}, L_{mask}, $L_{sparsity}$
α_{mask}	0.001	0.001
$\alpha_{sparsity}$	0.001	0.001
α_{recon}	-	1
Learning rate	0.0001	0.0001
Comb	-	Yes, No

Table 3.3: U-net based neural network hyperparameter

- **Batch size.** the number of data samples used in one iteration of training the model.
- **Drop rate.** the probability of deactivating neurons to prevent overfitting.
- **Dense size.** the number of neurons in a dense (fully connected) layer.
- **Div num filter.**a divisor used to determine the number of parameters in hidden layers of a deep learning model.
- **Loss function.** a function that calculates the error of the model, which the model optimizes to minimize during training.
- **Max epoch.** the maximum number of iterations over the entire dataset during training.
- **Early stop epoch.** The number of consecutive epochs without improvement in the loss function, after which training is stopped to prevent overfitting.

3.6.1 SUNSET

Varying 24 SUNSET parameters for solar irradiance forecasting using 15 lagged sky images and irradiance data as inputs on the CUEE, SKIPP'D, and SIRTA datasets. *Table 3.4* provides the hyperparameter settings for SUNSET.

Parameters	**CUEE**	**SKIPP'D**	**SIRTA**
Learning rate	0.001	0.000003	0.001
Batch size	512	128	64
Drop rate	0.001	0.4	0.001
Dense size	64, 128, 256, 1024	64, 128, 256, 1024	64, 128, 256, 1024
Div num filter	1, 2, 4	1, 2, 4	1, 2, 4
Loss function	AE, SE	AE, SE	AE, SE
Max epochs	200	200	200
Early stop epochs	10	10	10

Table 3.4: SUNSET hyperparameters varying on the CUEE, SKIPP'D, and SIRTA datasets

3.6.2 Unet

Varying with 18 Unet parameters to predict solar irradiance from sky image on the CUEE, SKIPP'D, and SIRTA datasets. *Table 3.5* provides the hyperparameter settings for Unet.

Parameters	**CUEE**	**SKIPP'D**	**SIRTA**
Learning rate	0.001	0.0001	0.0001
Batch size	128, 256, 512	128, 256, 512	128, 256, 512
Drop rate	0.001, 0.1, 0.4	0.001, 0.1, 0.4	0.001, 0.1, 0.4
Loss function	AE, SE	AE, SE	AE, SE
Max epochs	200	200	200
Early stop epochs	10	5	5

Table 3.5: Unet hyperparameters varying on the CUEE, SKIPP'D, and SIRTA datasets

3.6.3 Solarnet

Varying 30 Solarnet parameters for solar irradiance forecasting using 15 lagged sky images as inputs on the CUEE, SKIPP'D, and SIRTA datasets. *Table 3.6* provides the hyperparameter settings.

Parameters	**CUEE**	**SKIPP'D**	**SIRTA**
Learning rate	0.00001	0.0001	0.0001
Batch size	16, 64, 128	16, 64, 128	16, 64, 128
Drop rate	0.001	0.001	0.001
Number of layers	0, 1, 2, 3, 4	0, 1, 2, 3, 4	0, 1, 2, 3, 4
Loss function	AE, SE	AE, SE	AE, SE
Max epochs	200	200	200
Early stop epochs	10	5	5

Table 3.6: Solarnet hyperparameters varying on the CUEE, SKIPP'D, and SIRTA datasets

3.7 Hyperparameter settings for training with processed data

This study enhances sky images by performing RoI extraction and distortion correction on the SKIPP'D and SIRTA datasets and trains SUNSET, Solarnet, and U-Net on the processed SIRTA data with the hyperparameters provided in *Table 3.7*.

Parameters	SUNSET	Solarnet	Unet
Learning rate	0.001	0.0001	0.0001
Batch size	64	16	128
Drop rate	0.001	0.001	0.001, 0.1, 0.4
Dense size	1024	-	-
Div num filter	1, 2, 4	-	-
Number of layers	-	0, 1, 2, 3, 4	-
Loss function	AE, SE	AE, SE	AE, SE
Max epochs	200	100	200
Early stop epochs	10	5	5

Table 3.7: SUNSET, Solarnet, and Unet hyperparameters varying on the SIRTA processed data

3.8 Training of solar irradiance forecasting with cloud mask and foreground and hyperparameter settings

This study aims to enhance the model performance by extracting additional features from sky images, specifically cloud masks. SUNSET model was trained using the first six months of the SIRTA dataset, incorporating an additional cloud mask channel from [16] and generative omnimatte eq. (3.3).

- Experiment 1 : adding 1 channel of cloud mask from section 3.3.1.
- Experiment 2 : adding 3 channels of cloud foreground from section 3.3.2.
- Experiment 3 : adding 1 channel of soft-decision mask α from comb-on eq. (3.3).
- Experiment 4 : adding 3 channels of cloud foreground I_{cloud} from the comb-on eq. (3.3).

These experiments were conducted to improve model performance, utilizing consistent hyperparameters across all experiments. Learning rate = 0.001, batch size = 64, drop rate = 0.001, dense size = 1024, div num filder = 1, loss function = AE/SE, max epochs = 200, early stop epochs = 10.

Chapter 4

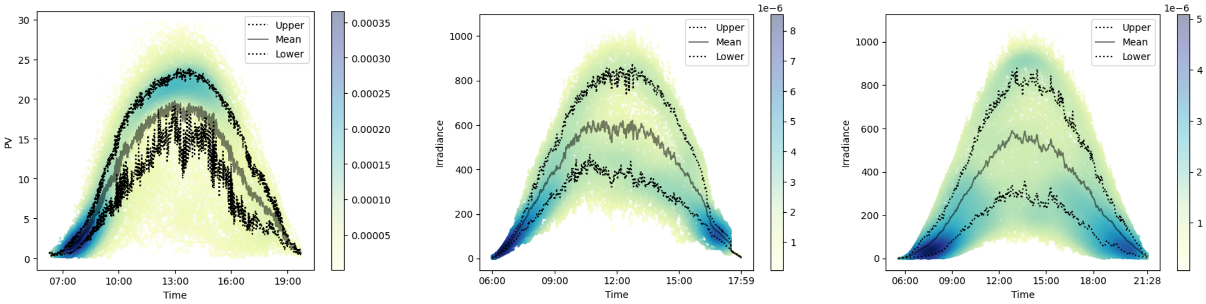
Results and discussion

4.1 Dataset analysis

SKIPP'D, CUEE, and SIRTA comprise sky images along with the corresponding solar irradiance or photovoltaic (PV) measurements, each collected from distinct geographical locations. They exhibit unique characteristics in terms of data acquisition methods, temporal resolution, data distribution, and other factors, as detailed in *Table 4.1* and *Figure 4.1*.

Dataset	SKIPP'D	CUEE	SIRTA
Camera Model	DS-2CD6365G0E-IVS	DS-2CD1021G0-I	EKO SRF-02
Resolution	64 × 64	1920 x 1080	64 × 64
Lens Type	Auto focus lens	DSLR lens	Auto focus lens
Capture Interval	1 minute	1 minute	1-2 minute
Start Date	09/03/2017	15/03/2023	01/01/2023
End Date	26/10/2019	03/11/2023	31/12/2023
Start Time	06:00	06:00	05:00
End Time	Not over 20:00	18:00	Not over 22:00

Table 4.1: Data properties



(a) SKIPP'D [KW] (b) CUEE [Watt/m^2] (c) SIRTA [Watt/m^2]

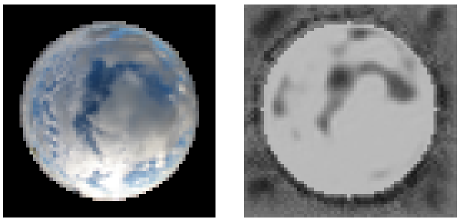
Figure 4.1: Distribution of solar irradiance [Watt/m^2] and PV [KW]

From *Figure 4.1*, three datasets SKIPP'D, CUEE, and SIRTA exhibit distinct distributions of solar irradiance or PV output. SKIPP'D shows a stable and narrow distribution with low variability, suggesting clear sky conditions and consistent output throughout the day. CUEE presents moderate variability with slightly wider uncertainty bands, indicating more dynamic weather patterns. SIRTA exhibits the highest variability and the widest spread, likely due to complex environmental factors or urban influences. In general, SKIPP'D provides the most stable profile for modeling, while SIRTA offers a more challenging dataset to evaluate model robustness under uncertain conditions.

4.2 Generative omnimatte results

4.2.1 Cloud mask and cloud foreground extraction

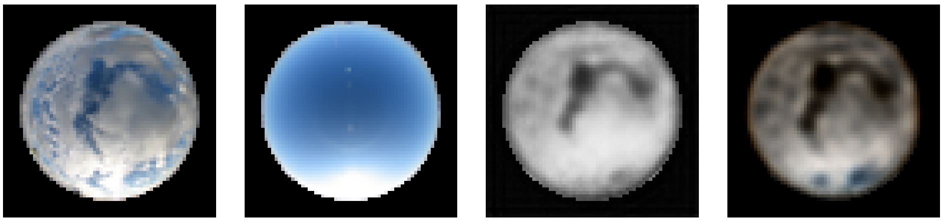
Base model was trained using the mask loss and sparsity loss. It is capable of generating cloud masks; however, the grayscale accuracy remains limited. In particular, when thin clouds are present, the model often predicts values close to one (highest value) across the region as in *Figure 4.2*.



(a) Sky image (b) Cloud mask

Figure 4.2: Cloud extraction with based model on SIRTA dataset

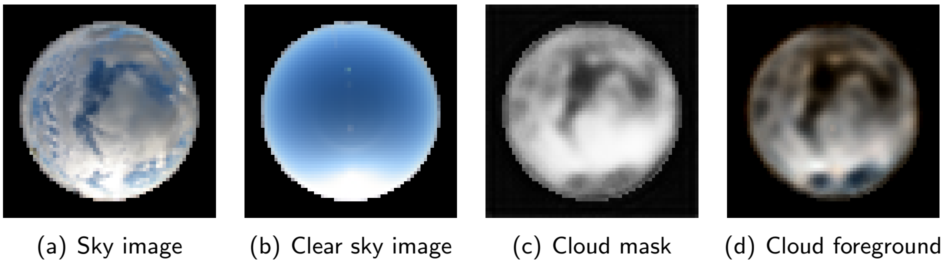
Comb-off model is a model transferred from the base model and is trained using the reconstruction loss (L_{recon}) in addition to the masking and sparsity loss. In this version, the predicted mask is applied directly to the cloud foreground to compute the reconstructed image (eq. (3.2)). This model improves grayscale performance by allowing the network to better distinguish between different cloud densities. It marks a significant improvement over the base model, especially in detecting thinner clouds more accurately. The example results are shown in *Figure 4.3*.



(a) Sky image (b) Clear sky image (c) Cloud mask (d) Cloud foreground

Figure 4.3: Cloud extraction with comb-off model on SIRTA dataset

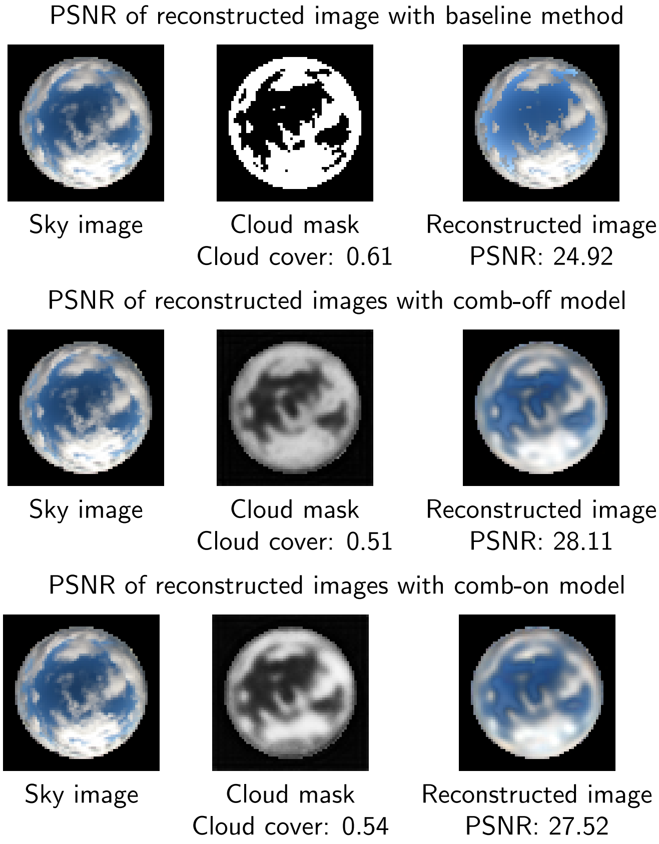
Comb-on model also uses L_{recon}, but with a different reconstruction method. Before applying the predicted mask, the cloud foreground is first combined with the clear-sky image (eq. (3.3)). This allows the network to distinguish between actual cloud regions and bright areas caused by sunlight. By including the clear sky image in the reconstruction process, the model is encouraged to circumvent the sunlight effect and avoid mistakenly labeling the sun as a cloud. The example results are shown in *Figure 4.4*.



(a) Sky image (b) Clear sky image (c) Cloud mask (d) Cloud foreground

Figure 4.4: Cloud extraction with comb-on model on SIRTA dataset

4.2.2 Quality of the cloud image reconstruction



PSNR of reconstructed image with baseline method

Sky image

Cloud mask
Cloud cover: 0.61

Reconstructed image
PSNR: 24.92

PSNR of reconstructed images with comb-off model

Sky image

Cloud mask
Cloud cover: 0.51

Reconstructed image
PSNR: 28.11

PSNR of reconstructed images with comb-on model

Sky image

Cloud mask
Cloud cover: 0.54

Reconstructed image
PSNR: 27.52

Figure 4.5: PSNR of reconstructed images between baseline, comb-off, and comb-on models

This section compares the cloud image reconstruction results from generative omnimattes, both comb-on and comb-off models. This is to ensure that the training of generative omnimattes are achieving the desired goal. For generative omnimattes, the reconstructed cloud image is obtained by eq. (3.1). Then, the PSNR is calculated with respect to the sky image. These results are compared with the image reconstructed by baseline detection [16], which is defined in a similar manner, but α is a hard decision cloud mask. *Figure 4.5* and *Figure 4.6* provides the example results. The baseline detection [16] provides a clean reconstruction result in *Figure 4.5*; however, this is not the case in *Figure 4.6*. The image reconstructed by the baseline detection is overly simplified and does not capture the mist in the cloud. Meanwhile, reconstruction by generative omnimattes, both comb-on and comb-off models, can capture the mist in both *Figure 4.5* and *Figure 4.6*.

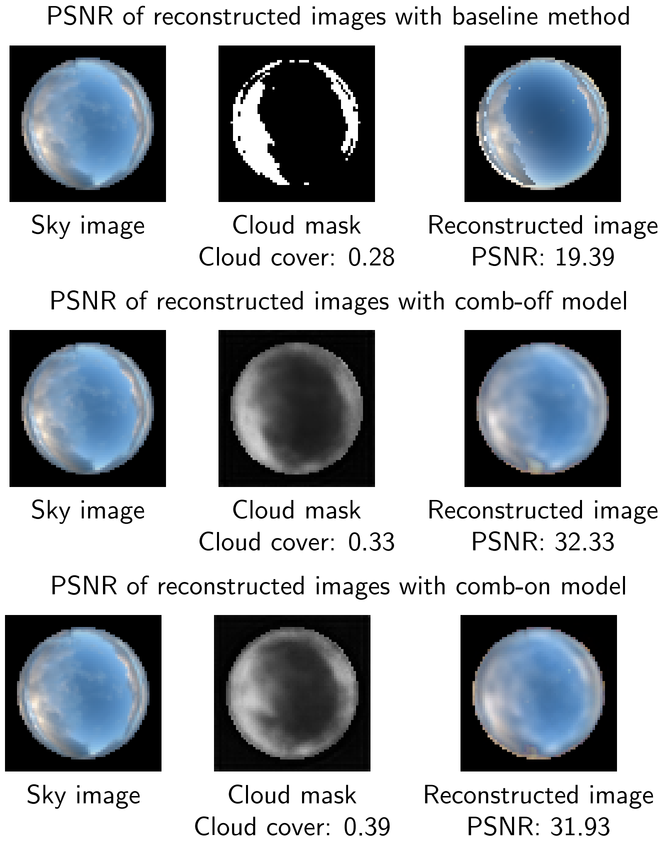


Figure 4.6: PSNR of reconstructed images between baseline, comb-off, and comb-on models

4.2.3 Quality of the predicted soft-decision mask

This section studies the quality of soft-decision cloud masks by generative omnimatte. Performance was validated by the correlation between the soft decision cloud mask with respect to two ground-truth binary cloud coverages, which are the cloud coverage of the Elifan dataset and the baseline method [16], as seen in *Figure 4.7*. The comparison indicates that both the comb-on and comb-off models perform similarly in generating cloud masks. However, the comb-on model was chosen for further exploration, since it can separate sunlight from cloud regions.

4.3 Model performance

The evaluation of model performance will involve the measurement of error according to two metrics.

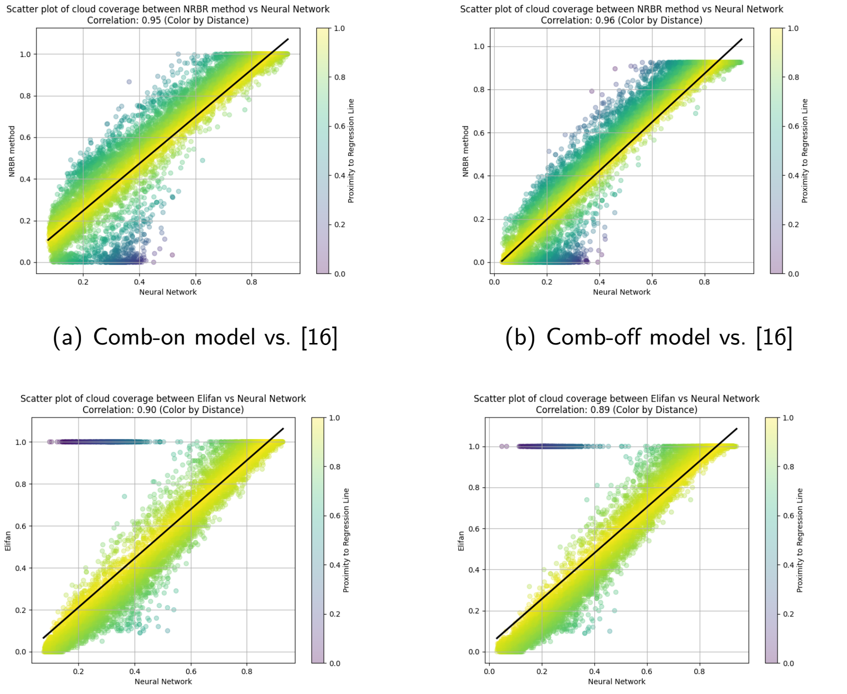
- Mean absolute error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.1}$$

- Root mean square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.2}$$

where n is the total number of data points.



(a) Comb-on model vs. [16]

(b) Comb-off model vs. [16]

(c) Comb-on model vs. Elifan

(d) Comb-off model vs. Elifan

Figure 4.7: Correlation between the cloud mask of generative omnimatte versus two baselines: [16] and Elifan dataset. **Correlation wrt. [16] are (a-b)**: (a) comb-on model, (b) comb-off model; and **Correlation wrt. Elifan (c-d)**: (c) comb-on model, (d) comb-off model.

The performance comparison of the models in forecasting solar irradiance for the next 15 minutes is shown in the following table. The parameters for the SUNSET, Solarnet, and Unet models were adjusted to find the parameters that resulted in the most accurate predictions on validation as measured by MAE and RMSE.

4.3.1 SUNSET, Solarnet, and Unet benchmarks

Validation dataset

This section reports the performance of SUNSET, Solarnet, and Unet on the validation datasets of CUEE, SKIPP'D, and SIRTA. The results are provided in *Table 4.2-Table 4.4.* The table confirms the lowest possible error that leads to the best model selection, which will be used in evaluation of testing dataset in the next section.

Model	**Learning loss.**	**Parameter setting**	**Model parameters**	**Train on AE**		**Train on SE**	
				MAE	RMSE	MAE	RMSE
SUNSET	MAE	Dense Size: 128, Div num filter: 1	3227185	102.00	160.26	-	-
	RMSE	Dense Size: 256, Div num filter: 1	6424497	-	-	103.98	155.04
Unet	MAE	Batch Size: 128, Drop rate: 0.4	306120	65.27	106.54	-	-
	RMSE	Batch Size: 256, Drop rate: 0.1	306120	-	-	66.19	109.32
Solarnet	MAE	Batch Size:64, Num layer: 3	7790098	127.50	178.41	-	-
	RMSE	Batch Size:16, Num layer: 4	14867974	-	-	129.35	172.27

Table 4.2: Performance on **CUEE validation dataset** with the lowest error in [Watt/m^2].

Model	**Learning loss.**	**Parameter setting**	**Model parameters**	**Train on AE**		**Train on SE**	
				MAE	RMSE	MAE	RMSE
SUNSET	MAE	Dense Size: 1024, Div num filter: 1	26296497	1.57	2.82	-	-
	RMSE	Dense Size: 1024, Div num filter: 2	13671897	-	-	1.66	2.73
Unet	MAE	Batch Size: 128, Drop rate: 0.1	306120	1.12	2.06	-	-
	RMSE	Batch Size: 128, Drop rate: 0.4	306120	-	-	1.21	2.08
Solarnet	MAE	Batch Size: 16, Num layer: 4	14867974	1.78	3.00	-	-
	RMSE	Batch Size: 16, Num layer: 4	14867974	-	-	1.91	2.94

Table 4.3: Performance on **SKIPP'D validation dataset** with the lowest error in [KW].

Model	**Learning loss.**	**Parameter setting**	**Model parameters**	**Train on AE**		**Train on SE**	
				MAE	RMSE	MAE	RMSE
SUNSET	MAE	Dense Size: 1024, Div num filter: 1	26296497	39.13	81.97	-	-
	RMSE	Dense Size: 1024, Div num filter: 4	7367373	-	-	45.61	85.91
Unet	MAE	Batch Size: 128, Drop rate: 0.4	306120	25.31	44.36	-	-
	RMSE	Batch Size: 128, Drop rate: 0.4	306120	-	-	25.41	43.04
Solarnet	MAE	Batch Size: 64, Num layer: 4	14867974	72.86	120.15	-	-
	RMSE	Batch Size: 16, Num layer: 4	14867974	-	-	65.31	105.05

Table 4.4: Performance on **SIRTA validation dataset** with the lowest error in [Watt/m^2].

Test dataset

The results demonstrate that the SUNSET model significantly outperforms the Solarnet model in the forecasting task across all three datasets. The results are provided in *Table 4.5*-*Table 4.7*. This performance advantage is primarily due to SUNSET's ability to utilize both sky images and time series data as input, whereas Solarnet relies solely on sky images.

In contrast, for the regression task, the Unet model achieves the lowest loss. This outcome is expected, as Unet predicts the irradiance at the current timestamp t, rather than forecasting it 15 minutes ahead, which inherently reduces the complexity of the prediction.

Figure 4.8 and *Figure 4.9* provide the example of the result comparison between the predicted value versus the ground truth on CUEE and SKIPP'D, respectively.

Model	**Learning loss.**	**Parameter setting**	**Model parameters**	**Train on AE**		**Train on SE**	
				MAE	RMSE	MAE	RMSE
SUNSET	MAE	Dense Size: 128, Div num filter: 1	3227185	92.06	152.52	-	-
	RMSE	Dense Size: 256, Div num filter: 1	6424497	-	-	94.39	146.86
Unet	MAE	Batch Size: 128, Drop rate: 0.4	306120	62.45	102.42	-	-
	RMSE	Batch Size: 256, Drop rate: 0.1	306120	-	-	62.12	95.95
Solarnet	MAE	Batch Size: 64, Num layer: 3	7790098	123.59	172.94	-	-
	RMSE	Batch Size: 16, Num layer: 4	14867974	-	-	123.95	167.75

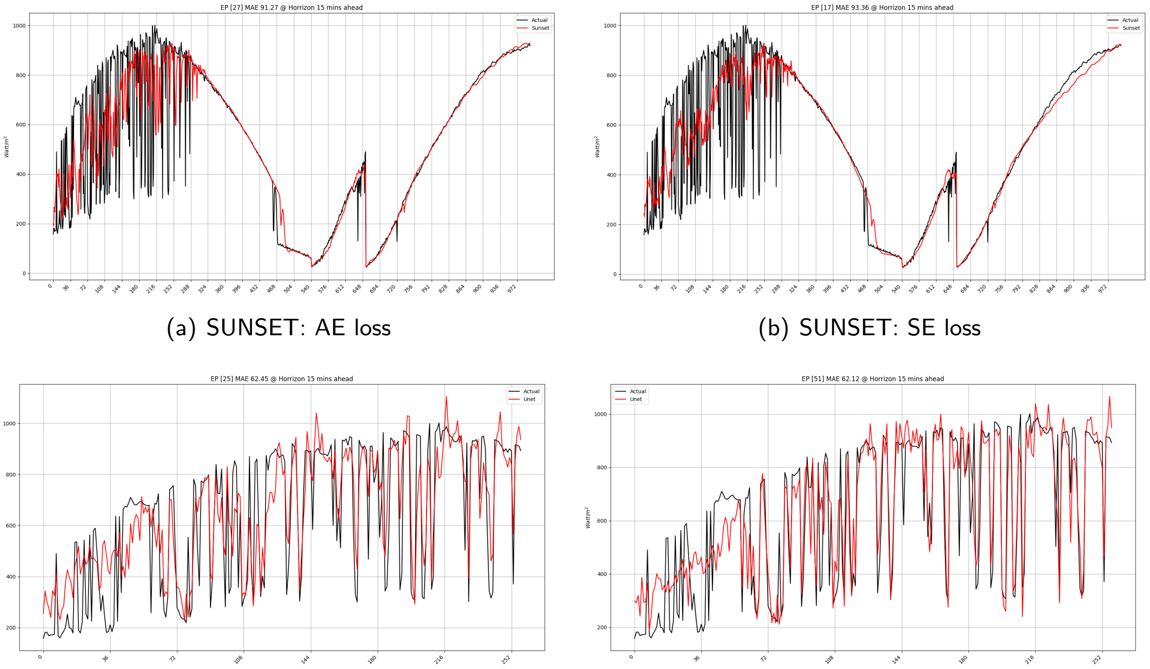
Table 4.5: Performance on **CUEE testing dataset** with error in Watt/m^2.

Model	Learning loss.	Parameter setting	Model parameters	Train on AE		Train on SE	
				MAE	RMSE	MAE	RMSE
SUNSET	MAE	Dense Size: 1024, Div num filter: 1	26296497	2.67	3.91	-	-
	RMSE	Dense Size: 1024, Div num filter: 2	13671897	-	-	2.77	3.89
Unet	MAE	Batch Size: 128, Drop rate: 0.1	306120	2.19	3.15	-	-
	RMSE	Batch Size: 128, Drop rate: 0.4	306120	-	-	2.27	3.20
Solarnet	MAE	Batch Size: 16, Num layer: 4	14867974	2.91	4.06	-	-
	RMSE	Batch Size: 16, Num layer: 4	14867974	-	-	3.03	4.08

Table 4.6: Performance on **SKIPP'D testing dataset** with the error values in KW.

Model	Learning loss.	Parameter setting	Model parameters	Train on AE		Train on SE	
				MAE	RMSE	MAE	RMSE
SUNSET	MAE	Dense Size: 1024, Div num filter: 1	26296497	38.97	84.41	-	-
	RMSE	Dense Size: 1024, Div num filter: 4	7367373	-	-	45.04	86.24
Unet	MAE	Batch Size: 128, Drop rate: 0.4	306120	24.99	39.89	-	-
	RMSE	Batch Size: 128, Drop rate: 0.4	306120	-	-	24.08	41.02
Solarnet	MAE	Batch Size: 64, Num layer: 4	14867974	72.65	121.16	-	-
	RMSE	Batch Size: 16, Num layer: 4	14867974	-	-	65.03	106.86

Table 4.7: Performance on **SIRTA testing dataset** with the error values in Watt/m^2.



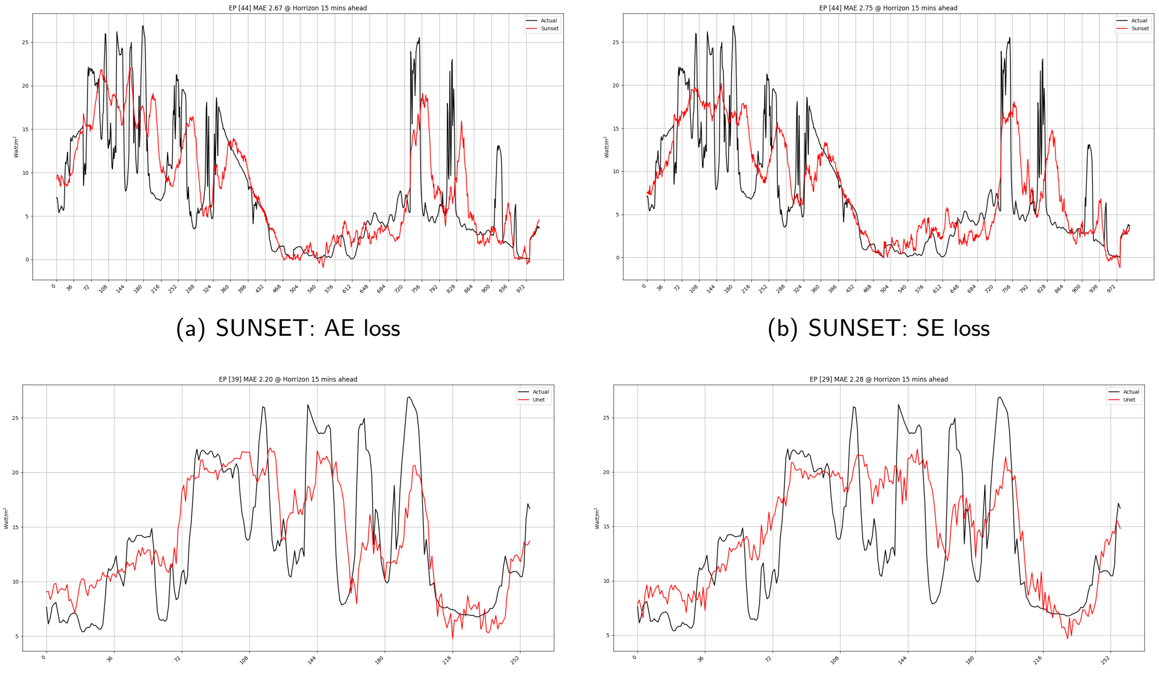
(a) SUNSET: AE loss

(b) SUNSET: SE loss

(c) Unet: AE loss

(d) Unet: SE loss

Figure 4.8: **CUEE Dataset**: Solar irradiance forecasting results compared to actual values using the **SUNSET (a-b)**: (a) trained with AE loss, (b) trained with SE loss; and using the **Unet model (c-d)**: (c) trained with AE loss, (d) trained with SE loss.



(a) SUNSET: AE loss

(b) SUNSET: SE loss

(c) Unet: AE loss

(d) Unet: SE loss

Figure 4.9: **SKIPP'D Dataset**: Solar irradiance forecasting results compared to actual values using the **SUNSET model (a-b)**: (a) trained with AE loss, (b) trained with SE loss; and **Unet model (c-d)**: (c) trained with AE loss, (d) trained with SE loss.

4.3.2 Training with processed data

This study trained SUNSET, Solarnet, and U-Net on SIRTA-processed data, which has undergone RoI extraction and correction of distortion effects from the fisheye lens. We choose the best model from the validation in *Table 4.8*, and evaluate the performance of the model in the testing in *Table 4.9*. The experimental results indicate that the implemented approach did not lead to a significant enhancement in model performance. A plausible explanation for this outcome is the utilization of low-resolution images (64×64 pixels) during the training phase. Such low-resolution inputs may result in the loss or distortion of critical features necessary for accurate model predictions.

Validation dataset

Model	**Learning loss.**	**Parameter setting**	**Model parameters**	**Train on AE**		**Train on SE**	
				MAE	RMSE	MAE	RMSE
SUNSET	MAE	Dense Size: 1024, Div num filter: 1	26296497	43.07	86.38	-	-
	RMSE	Dense Size:1024, Div num filter: 4	3237081	-	-	46.20	84.76
Unet	MAE	Batch Size:128, Drop rate:0.4	306120	30.06	50.83	-	-
	RMSE	Batch Size:128, Drop rate:0.4	306120	-	-	30.40	51.10
Solarnet	MAE	Batch Size:16, Num layer: 4	14867974	67.24	115.49	-	-
	RMSE	Batch Size:16, Num layer: 4	14867974	-	-	67.67	108.31

Table 4.8: Performance on **SIRTA validation dataset** with the lowest error values.

Test dataset

Model	Learning loss.	Parameter setting	Model parameters	Train on AE		Train on SE	
				MAE	RMSE	MAE	RMSE
SUNSET	MAE	Dense Size: 1024, Div num filter: 1	26296497	42.91	88.45	-	-
	RMSE	Dense Size:1024, Div num filter: 4	3237081	-	-	45.86	85.65
Unet	MAE	Batch Size:128, Drop rate:0.4	306120	29.01	48.46	-	-
	RMSE	Batch Size:128, Drop rate:0.4	306120	-	-	29.66	48.12
Solarnet	MAE	Batch Size:16, Num layer: 4	14867974	66.79	116.81	-	-
	RMSE	Batch Size:16, Num layer: 4	14867974	-	-	67.50	110.16

Table 4.9: Performance on **SIRTA testing dataset**.

4.3.3 Training with cloud mask and foreground

This study aims to enhance model performance by extracting cloud masks using various approaches and training the SUNSET model on the first six months of the SIRTA dataset because of limitation of CPU RAM. The experiments are conducted using the only one configuration hyperparameters outlined in section 3.8 and are based on four different cloud masking strategies.

- Experiment 1 : adding 1 channel of cloud mask from section 3.3.1.
- Experiment 2 : adding 3 channels of cloud foreground from section 3.3.2.
- Experiment 3 : adding 1 channel of soft-decision mask α from comb-on eq. (3.3).
- Experiment 4 : adding 3 channels of cloud foreground I_{cloud} from the comb-on eq. (3.3).

Results.

- *Table 4.10* demonstrates that none of the experiments resulted in an improvement in MAE. However, with respect to RMSE, Experiments 2 and 3 appear to show a slight decrease.

Evaluation	Learning loss: MAE				
	Original	Experiment 1	Experiment 2	Experiment 3	Experiment 4
MAE	32.74	36.51	33.93	33.28	36.79
RMSE	76.25	78.80	75.77	75.22	80.22

Table 4.10: Comparison of MAE across different experiments

- *Table 4.11* illustrates that all experiments led to improvements in both MAE and RMSE, with Experiment 1 showing particularly notable enhancements.

Evaluation	Learning loss: RMSE				
	Original	Experiment 1	Experiment 2	Experiment 3	Experiment 4
MAE	42.91	34.76	38.83	36.82	39.12
RMSE	79.03	73.32	77.06	74.08	76.69

Table 4.11: Comparison of RMSE across different experiments

The results of Experiment 1 indicate the most significant improvement in model performance with respect to RMSE, achieving approximately a 3% reduction in RMSE loss compared to the optimal RMSE observed in the original dataset.

However, in terms of MAE, adding cloud masking channels did not lead to significant improvement. This may be because the additional channels help reduce large spikes in error (as reflected by the RMSE), but the model still produces consistent small deviations, which MAE is more sensitive to.

Chapter 5

Conclusion

This study explored the CUEE, SKIPP'D, and SIRTA datasets, each with its unique characteristics. Various models, including SUNSET, Unet, and Solarnet, were trained on the CUEE and SKIPP'D datasets to compare the performance for solar irradiance forecasting. We have also done data preparation to enhance models, including: RoI extraction to filter out irrelevant image parts, lens distortion correction to fix the disproportionate of image sizes, extraction of the sun's position in sky images, cloud extraction, and cloud masking from sky images to be used as input variables for model training.

The results from training the forecasting models demonstrate that SUNSET consistently achieves the best performance in the forecasting task across all three datasets. This superior performance can be attributed to its ability to leverage both spatial information from sky images and temporal patterns from time-series data. To further enhance the SUNSET model's performance on the SIRTA dataset, this study incorporated an additional cloud mask channel method from [16], resulting in an approximate 3% improvement in RMSE compared to the original sky image data, but in terms of MAE, the addition of cloud mask and cloud foreground channels did not lead to a significant improvement. This suggests that the benchmark system needs to be further improved, as the current training and evaluation in this part can handle no more than 6 months of data due to high CPU RAM requirement.

5.1 Operation plan

Work process	2024					2025			
	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
1. Exploratory data analysis	░	░	░						
	▓	▓	▓						
2. Processing sky images		░	░						
		▓	▓						
3. Extracting additional data such as cloud volume and cloud marking			░	░					
			▓	▓					
4. Train and evaluate neural network architectures on three datasets		░	░	░	░				
		▓	▓	▓	▓				
5. Train and evaluate neural network architectures on processed SIRTA dataset					░	░	░		
					▓	▓	▓		
6. Train and evaluate neural network architectures on SIRTA with additional data					░	░	░	░	
					▓	▓	▓	▓	
7. Write an academic report								░	░
								▓	▓

▓: Current ░: Plan

5.2 Problems, obstacles, and solutions

Solar irradiance forecasting is challenging due to the variability of cloud formations, atmospheric conditions, and differences in geographical locations. This study aims to introduce deep learning techniques to enhance performance, particularly by processing images and extracting more additional data.

One obstacle is that the limited memory on the CPU and GPU RAM poses a significant challenge in the handling of high-resolution images. Consequently, the original images, stored in the database at full resolution, must be downscaled to 64×64 pixels for model training. This reduction in resolution can lead to a loss of image quality and critical features, which may affect the accuracy of the results and complicate their interpretation. To overcome this limitation, an on-the-fly data generation approach can be employed. By leveraging GPU batch processing, this method can be further improved for real-time image handling and cloud detection, optimizing resource usage while preserving essential details during training.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to Dr. Suwichaya Suwanwimolkul, my advisor, for her invaluable guidance, continuous support, and insightful feedback throughout the course of this research. Her expertise and encouragement have played a crucial role in the successful completion of this project.

I would also like to thank all faculty members, peers, and collaborators who contributed their time and knowledge to enrich my learning experience.

Lastly, I am sincerely thankful to my family and friends for their unwavering support and motivation during every stage of this academic journey.

Respectfully,
Kanawut Suwandee
Kongpob In-odd

Bibliography

[1] S. Janjai, I. Masiri, S. Pattarapanitchai, and J. Laksanaboonsong, "Mapping global solar radiation from long-term satellite data in the tropics using an improved model," *International Journal of Photoenergy*, vol. 2013, 2013.

[2] Q. Paletta, G. Arbod, and J. Lasenby, "Benchmarking of deep learning irradiance forecasting models from sky images – an in-depth analysis," *Solar Energy*, vol. 224, pp. 855–867, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X21004266

[3] ——, "Omnivision forecasting: Combining satellite and sky images for improved deterministic and probabilistic intra-hour solar energy predictions," *Applied Energy*, vol. 336, p. 120818, 4 2023.

[4] C. Feng and J. Zhang, "Solarnet: A sky image-based deep convolutional neural network for intra-hour solar forecasting," *Solar Energy*, vol. 204, pp. 71–78, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X20303285

[5] S. Boonprasert, "Development of a low-cost sky camera system with solar irradiance estimation and very short-term forecast capabilities," 2023.

[6] S. Mesakulroong, "Optical-flow learning for sky image generation," 2023.

[7] L. Julian and A. C. Sankaranarayanan, "Precise forecasting of sky images using spatial warping," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021.

[8] Q. Paletta, A. Hu, G. Arbod, and J. Lasenby, "Eclipse: Envisioning cloud induced perturbations in solar energy," *Applied Energy*, vol. 326, p. 119924, 2022.

[9] Y. Nie, X. Li, A. Scott, Y. Sun, V. Venugopal, and A. Brandt, "Skipp'd: A sky images and photovoltaic power generation dataset for short-term solar forecasting," *Solar Energy*, vol. 255, pp. 171–179, 5 2023.

[10] M. Haeffelin, L. Barthès, O. Bock, C. Boitel, S. Bony, D. Bouniol, H. Chepfer, M. Chiriaco, J. Cuesta, J. Delanoë, P. Drobinski, J.-L. Dufresne, C. Flamant, M. Grall, A. Hodzic, F. Hourdin, F. Lapouge, Y. Lemaitre, A. Mathieu, Y. Morille, C. Naud, V. Noël, W. O'Hirok, J. Pelon, C. Pietras, A. Protat, B. Romand, G. Scialom, and R. Vautard, "Sirta, a ground-based atmospheric observatory for cloud and aerosol research," *Annales Geophysicae*, vol. 23, pp. 253–275, 2005. [Online]. Available: https://angeo.copernicus.org/articles/23/253/2005/

[11] Y. Sun, V. Venugopal, and A. R. Brandt, "Short-term solar power forecast with deep learning: Exploring optimal input and output configuration," *Solar Energy*, vol. 188, pp. 730–741, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X19306164

[12] Y. Nie, E. Zelikman, A. Scott, Q. Paletta, and A. Brandt, "Skygpt: Probabilistic ultra-short-term solar forecasting using synthetic sky images from physics-constrained videogpt," *Advances in Applied Energy*, vol. 14, p. 100172, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666792424000106

[13] S. Song, Z. Yang, H. Goh, Q. Huang, and G. Li, "A novel sky image-based solar irradiance nowcasting model with convolutional block attention mechanism," *Energy Reports*, vol. 8, pp. 125–132, 2022, iCPE 2021 - The 2nd International Conference on Power Engineering. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484722004127

[14] S. Suwanwimolkul, N. Tongamrak, N. Thungka, N. Hoonchareon, and J. Songsiri, "Developing a thailand solar irradiance map using himawari-8 satellite imageries and deep learning models," 2024. [Online]. Available: https://arxiv.org/abs/2409.16320

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

[16] Y. Nie, Y. Sun, Y. Chen, R. Orsini, and A. Brandt, "Pv power output prediction from sky images using convolutional neural network: The comparison of sky-condition-specific sub-models and an end-to-end model," *Journal of Renewable and Sustainable Energy*, vol. 12, no. 4, p. 046101, 08 2020. [Online]. Available: https://doi.org/10.1063/5.0014016

[17] A. Da Rosa and Ordóñez, *Fundamentals of renewable energy processes*. [Online]. Available: https://www.sciencedirect.com/book/9780128160367/fundamentals-of-renewable-energy-processes

[18] R. Chauvin, J. Nou, S. Thil, A. Traoré, and S. Grieu, "Cloud detection methodology based on a sky-imaging system," *Energy Procedia*, vol. 69, pp. 1970–1980, 2015, international Conference on Concentrating Solar Power and Chemical Energy Systems, SolarPACES 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1876610215005044

[19] Y. Nie, X. Li, Q. Paletta, M. Aragon, A. Scott, and A. Brandt, "Open-source sky image datasets for solar forecasting with deep learning: A comprehensive survey," *Renewable and Sustainable Energy Reviews*, vol. 189, p. 113977, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032123008353

[20] Y. Nie, Q. Paletta, A. Scott, L. M. Pomares, G. Arbod, S. Sgouridis, J. Lasenby, and A. Brandt, "Sky image-based solar forecasting using deep learning with heterogeneous multi-location data: Dataset fusion versus transfer learning," *Applied Energy*, vol. 369, p. 123467, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030626192400850X

[21] E. Lu, F. Cole, T. Dekel, W. Xie, A. Zisserman, D. Salesin, W. T. Freeman, and M. Rubinstein, "Layered neural rendering for retiming people in video," in *SIGGRAPH Asia*, 2020.

[22] Y.-C. Lee, E. Lu, S. Rumbley, M. Geyer, J.-B. Huang, T. Dekel, and F. Cole, "Generative omnimatte: Learning to decompose video into layers," *arXiv preprint arXiv:2411.16683*, 2024.

[23] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, D. Nair, S. Paul, W. Berman, Y. Xu, S. Liu, and T. Wolf, "Diffusers: State-of-the-art diffusion models," https://github.com/huggingface/diffusers, 2022.

[24] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[25] U. de Lille – Cité Scientifique Bâtiment M3 extension Avenue Carl Gauss 59650 Villeneuve d'Ascq Cedex (France), "Aeris/icare data and services center – uar 2877," https://dataviz.icare.univ-lille.fr/elifan_dataviz, 2025.

[26] M. Lothon, P. Barnéoud, O. Gabella, F. Lohou, S. Derrien, S. Rondi, M. Chiriaco, S. Bastin, J.-C. Dupont, M. Haeffelin, J. Badosa, N. Pascal, and N. Montoux, "Elifan, an algorithm for the estimation of cloud cover from sky imagers," *Atmospheric Measurement Techniques*, vol. 12, no. 10, pp. 5519–5534, 2019. [Online]. Available: https://amt.copernicus.org/articles/12/5519/2019/

[27] P. Ineichen and R. Perez, "A new airmass independent formulation for the linke turbidity coefficient," *Solar Energy*, vol. 73, no. 3, pp. 151–157, 2002.

[28] A. R. Jensen, K. S. Anderson, W. F. Holmgren, M. A. Mikofski, C. W. Hansen, L. J. Boeman, and R. Loonen, "pvlib iotools—open-source python functions for seamless access to solar irradiance data," *Solar Energy*, vol. 266, p. 112092, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X23007260