

Improving irradiance prediction by using classification of sky images and deep learning

Gijs van Urk:
MSc Sustainable
Energy
Technology

IMPROVING IRRADIANCE PREDICTION BY USING CLASSIFICATION OF SKY IMAGES AND DEEP LEARNING

A thesis submitted to the Delft University of Technology in partial fulfillment
of the requirements for the degree of

Master of Science in Sustainable Energy Technology

by

Gijs van Urk

13 December 2022

Gijs van Urk: *Improving irradiance prediction by using classification of sky images and deep learning* (2022)

© This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

ISBN

The work in this thesis was made in the:



PVMD group
Department of Electrical Sustainable Energy
Faculty of Electrical Engineering, Mathematics & Computer Science
Delft University of Technology

Supervisors: Dr. ir. Jochem Cremer
Dr. ir. Hesan Ziar
Ir. Victor Arturo Martinez Lopez
Chairman : Prof. dr. ir. Rene van Swaaij

PREFACE

Before starting this research, it had to figure out what seemed interesting to me to start exploring. I became interested in a number of research subjects within the Photovoltaic Materials and Devices department, so I started reading up on them to get a better idea of what I was most interested in. Then finally I decided to go for the project named 'cloud models v2.0', where I would elaborate on the previous research of Pim Doodkorte.

In the beginning it was difficult for me to continue the work of Pim. But after some interesting discussions with Pim and my Supervisors, I slowly started to get into this research topic more and more. My research abilities have significantly improved as a result of the MSc thesis, and I now have a greater understanding of the problem-solving potential of deep learning. In addition, I gained experience working with a variety of enormous volumes of data and a lot in image segmentation.

ACKNOWLEDGEMENTS

First of all, I want to thank my MSc Thesis Committee Rene van Swaaij, Hesan Ziar, Jochem Cremer and Arturo Martinez Lopez for assessing my work. Then, I want to yet again thank my supervisors Hesan Ziar and Arturo Martinez Lopez for guidance during the research. Furthermore, once again a special thanks to my daily supervisor Arturo who have guided my through this research and help me a lot in gaining knowledge to bring this research to a higher academic level. On top of that, Arturo has taught me some great life lessons through quotes, such as: 'When there's doubt ... there's no doubt.' At last, I would like to thank Pim Doodkorte for all the preliminary work he has done for me, otherwise I would not have been able to achieve this research either.

Gijs van Urk

13 December 2022

1

ABSTRACT

Short-term solar forecasting is essential for the large-scale application of solar energy and is necessary for the operation of power plants, energy trading, and grid balancing. The main cause of uncertainty in solar forecasting is cloud movement, which can be observed by All-Sky Images. The spatial layout and temporal dynamics of clouds cannot be extracted by conventional cloud modelling approaches utilising image analysis techniques, leading to inaccurate predictions of the interaction with solar radiation. In this study an categorization method is made based on sky conditions and from that five classes are made. The goal of this classification method is to improve the 21-minute irradiance predictions which are made with an deep learning model. The outcome of the 21-minute deep learning model will get compared with the Persistence, Smart Persistence and ARIMA model. This classification method and irradiance prediction are applied for location Folsom, California and Delft, Netherlands. The irradiance predictions based on sky classification showed an improvement of 8.6% for location Delft and 29.3% for location Folsom when the same dataset sizes are used. The irradiance prediction were also done for various dataset sizes and compared per location. Finally, it provides a way forward to improve the classification method and deep learning models to anticipate short-term irradiance in the future.

CONTENTS

1	ABSTRACT	vii
2	INTRODUCTION	1
3	BACKGROUND	3
3.1	Meteorology	3
3.2	Sun	4
3.2.1	Position of the Sun	4
3.2.2	Solar radiation	5
3.2.3	Clear Sky Irradiance	7
3.3	Clouds	8
3.4	Irradiance forecasting	8
3.4.1	Statistical models	10
3.4.2	All sky images	12
3.5	Weather classification methods	13
3.5.1	Image thresholding methods	15
3.5.2	Clear-sky modeling	18
3.6	Machine learning	20
3.7	Deep Learning	24
3.7.1	Artificial Neural Network	24
3.7.2	Activation functions	25
3.7.3	Deep learning layers	26
3.7.4	Deep learning model training	28
4	METHODOLOGY	31
4.1	Pre-processing data	31
4.2	Classification process	32
4.2.1	Roadmap through the classification	32
4.2.2	Performance evaluation	35
4.3	Prediction model architectures	37
4.3.1	Roadmap through the prediction models	37
4.3.2	Performance evaluation	39
5	RESULTS	43
5.1	Classification results	43
5.1.1	Classification sensitivity analysis	45
5.2	Prediction results	47
5.2.1	Results without classification	47
5.2.2	Results with classification	48
5.2.3	Prediction sensitivity analysis	49
6	DISCUSSION	53
6.1	Classification comparison	53
6.1.1	All Sky Camera	54
6.2	Prediction comparison	55
6.3	Future work	56
6.4	Conclusion	57
A	APPENDIX	67
A.1	Minutely MAE and Quantile 95% per Class	67

LIST OF FIGURES

Figure 3.1	The five layers of the Atmosphere, with the given height and temperature range per layer. Image from Annnsky [2022]	3
Figure 3.2	Illustration on how to find the location of the sun in relation to the earth. The Observer is the specific location at Earth and the Object is the sun. In order to do this the Altitude (a) angle and Azimuth angle (A) must be known Smets et al. [2015]	4
Figure 3.3	The sun path of two specific days the at the location Delft, the Netherlands. The inner circles represents the radius at a specific Zenith angle (θ) and the circles turned clockwise represents the Azimuth angle (A). Figure is made with Python 3.9	5
Figure 3.4	The Direct Normal Irradiance (DNI), which is the irradiance perpendicular to the sun and the Diffuse Horizontal Irradiance (DHI) which is the reflected irradiance on that same point. Combine the DNI and DHI and the Global Horizon Irradiance (GHI) can be found and measured by the pyranometer. Figure obtained from Hukseflux [2006]	6
Figure 3.5	The Clear Sky Irradiance (CSI) for the first of August 2021 in Delft projected against the real GHI during that day. Figure is made with Python 3.9.	7
Figure 3.6	The steps that are made to become a cloud from a water vapor. Air becomes cooler and adhere to dust, then cloud droplets are made. The light cloud droplet accumulate and clouds are created Novati [2022]	8
Figure 3.7	The ten different cloud classification illustrated. The classification is done based on height and cloud characteristics. Palubicki [2020]	9
Figure 3.8	The four forecast horizon for irradiance forecasting with the time step and the used application. Kumari and Toshniwal [2020]	9
Figure 3.9	Overview of the prediction of solar irradiance methods with there spatial coverage and time horizon made by V.A. Martinez Lopez. Each time horizon has there own goal. Image based on work of Blanc [2017] , Kazantzidis [2017] and NREL [2016]	10
Figure 3.10	The visualized difference between (a) the RGB pixel and (b) the HSV pixel, image from Popov et al. [2018]	13
Figure 3.11	The ASI that gets converted into an RGB matrix which can be used for programming in Python.	13
Figure 3.12	The mask layer of the ASI at the TU Delft. These mask layer will be used on top of the ASI to calculate the final CC. The mask layer will not be taken into account for calculating the final CC.	14
Figure 3.13	Overview of the different cloud covers expressed in oktas, from Skull [2017]	15

Figure 3.14	(A) The threshold value calculated for an fish-eye ASI and the outcome of the binary ASI. Note that the mask background is not yet taken into account. (B) The threshold value calculated for an flattened ASI and the outcome of the binary ASI. Note that the mask background is not yet taken into account. (C) Example of an gray scaled image where the maximum inter-class variation is found to find the specific threshold value for this image when using the Otsu threshold value Szelag [2020]	17
Figure 3.15	The Superpixel segmentation method explained in 5 steps. First will the ASI gets flattened. Second will the SPS method by applied. Third the mask applied. Fourth the final threshold image is made. Finally the CC % can get calculated. . . .	18
Figure 3.16	The visual method of how to find the X and Y coordinate of the sun in an All Sky Image.	19
Figure 3.17	Global overview of the difference between Artificial Intelligence (AI), Machine Learning (ML) and Deep learning (DL). Jawahar and Anoop [2022]	20
Figure 3.18	Visually explanation of the Support Vector Machine method, image from Dabakoglu [2018]	21
Figure 3.19	Example of the SVR, to show how it works and visualize the used terms. Image taken from Sharp [2020]	22
Figure 3.20	An example of how the k-means segmentation method works. It is an unsupervised learning method, so non-labeled data comes in and the method labels the data. Above images from Chhabria [2021] . The lower images are example of the applied k-means clustering method on an ASI.	23
Figure 3.21	An example of how the Artificial Neural Networks (ANN) looks like when it would be drawn. Part of the Figure taken by Dertat [2017]	24
Figure 3.22	An example of how the zoomed in Node looks like when it would be drawn and when an Activation function is added. Figure taken by Dertat [2017]	25
Figure 3.23	Activation functions in comparison. Red curves stand for, respectively,(a) Sigmoid,(b) hyperbolic tangent, (c) ReLU, and (d) Softplus functions. Their first derivative is plotted in blue. Figure taken from Roffo [2017]	27
Figure 3.24	An Example of how the convolutional layers works for an RGB image with 6 by 6 pixel size that goes through six 2D kernel (filters), from these multiplication two feature maps are created. Than feature map values can be passed through the activation function, which in this example is the ReLU function with an bias (b). From this an 3 by 3 by 2 output is given. Image from Ng [2018]	27
Figure 3.25	An example of the two different pooling layers techniques, maximum pooling (left part) and average pooling (right part). Image from Ng [2018]	28
Figure 3.26	An example of the pooled that gets flattened to a flatten layer. This is needed to find the final 1D output value from the model. Image from Biswal [2022]	28
Figure 4.1	All the equipment needed for making the all sky images and the extra information, located at the TU Delft, the Netherlands.	31
Figure 4.2	(a) Overview of the amount of data per month that is used for the location Delft. (b) Overview of the amount of data per month that is used for the location Folsom.	32

Figure 4.3	The real sunpath of the ASI's for the day 1st April in 2021. In figure 4.2(A), the sunpath of one day is shown for the images of that day. Figure 4.2(b) shows how the sun path would actually look like. In Figure 4.2(c) the both images are merged together so one could see the difference between the real sun path and the theoretical.	33
Figure 4.4	Four examples of the cropped image of where the sun in the image is located and the thresholded cropped image, to see how the sun gets detected.	34
Figure 4.5	The overview of how the cloud cover is calculated for all the ASI's. First one needs to find out whether the sun is blocked or not, then is there are cloud pixels in the image or not. Finally, the CC can be found for all the images. In this example the most two left images will be considered as blocked sun and the most right two images as sun not blocked with clouds.	35
Figure 4.6	Overview of how the ARIMA FIFO system works to make the 21-min GHI predictions. For every 21-min prediction, a 500 historical dataset is used which shifts throughout the dataset via the FIFO method.	38
Figure 4.7	This graphic shows the convolutional neural network that was utilised in the deep learning model to extract visual features. Figure made by Doodkorte [2021] and slightly adjusted.	38
Figure 4.8	This graphic shows the auxiliary dense neural network that was utilised in the deep learning model. Figure made by Doodkorte [2021] and slightly adjusted.	39
Figure 4.9	This graphic shows the total overview of the deep learning prediction mode that was utilised in the deep learning model. Figure made by Doodkorte [2021] and slightly adjusted.	39
Figure 5.1	Overview of the distributions of the classification over the 5 classes for location Delft & Folsom.	43
Figure 5.2	Difference distribution (in oktas) of CC algorithm and with the manual observation for (a) location Delft and (b) location Folsom.	44
Figure 5.3	The two example where the CC algorithm misclassified the CC %. (a) ASI is taken 4:19 (UTC) in the morning of the fifth of August in 2021. (b) ASI is taken 4:40 (UTC) in the morning of the 19th of August in 2021.	45
Figure 5.4	The Variability Index (VI) shown as box-and-whiskers plot for all the data and per class for location Delft on the left and location Folsom on the right.	46
Figure 5.5	Overview how the classes are classified based on the K_c and CC.	46
Figure 5.6	The 21-minute predictions of the 4 models namely: Persistence model, Smart Persistence model, ARIMA model & CNN model. The final outcome of the MAE per minute for the four prediction models for location Delft.	48
Figure 5.7	The 21-minute predictions of the 4 models namely: Persistence model, Smart Persistence model, ARIMA model & CNN model. The final outcome of the quantile 95% per minute for the four prediction models for location Delft.	48
Figure 5.8	The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class E is not applied, so afterwards classified.	51

Figure 5.9	The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class E is not applied, so afterwards classified.	51
Figure 6.1	The outcome of the Hartmann [2020] classification applied for location Delft and Folsom.	54
Figure A.1	The 21-minute MAE of the baseline model (SP) against the CNN model when the classification for Class A is applied beforehand against when the classification is not applied, so afterwards classified.	67
Figure A.2	The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification for Class A is applied beforehand against when the classification is not applied, so afterwards classified.	67
Figure A.3	The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class B is not applied, so afterwards classified.	68
Figure A.4	The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class B is not applied, so afterwards classified.	68
Figure A.5	The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class C is not applied, so afterwards classified.	68
Figure A.6	The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class C is not applied, so afterwards classified.	69
Figure A.7	The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class D is not applied, so afterwards classified.	69
Figure A.8	The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class D is not applied, so afterwards classified.	69
Figure A.9	The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class E is not applied, so afterwards classified.	70
Figure A.10	The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class E is not applied, so afterwards classified.	70

LIST OF TABLES

Table 3.1	overview of the cloud percentage can be converted to oktas. . .	14
Table 4.1	The image's pixel intensity values, excluding the region surrounding the sun among the 200 pictures.	34
Table 4.2	The overview of the five ASI classification based on the Cloud Cover and Clear Sky Index.	35
Table 5.1	The final outcome of the classification for both location Delft & Folsom.	43
Table 5.2	The final classification performance based on the recall, precision & F-score for both locations.	44
Table 5.3	The prediction results for the four different models in comparisons based on the MAE, RMSE, nMAE, nRMSE and FS for location Delft.	47
Table 5.4	The prediction results for the four different models in comparisons based on the MAE, RMSE, nMAE, nRMSE and FS for location Folsom.	49
Table 5.5	The results of the MAE per classification for the four prediction models for location Delft. For the CNN model 2 predictions were made. The first one 'CNN: class afterwards' is when the classification is done after the prediction. The second one 'CNN: class forwards' is when the classification is done before the prediction.	49
Table 5.6	The total improvement for making classifications forwards and afterwards based on a dataset of 19000 at the location Delft.	49
Table 5.7	The total improvement for making classifications forwards and afterwards based on a dataset of 9900 at the location Folsom.	50
Table 5.8	Results of the total improvement when all data per classification is used.	50
Table 5.9	Results of the total improvement when making classes of three instead of five for location Delft, Folsom and combined.	50
Table 5.10	The performance of the CNN-Main model compared to the CNN-Main models with the ablation of inputs is depicted. . .	52
Table 6.1	Comparing between the studies of Xie and Yiren [2019] and Kazantzidis et al. [2012a] and this research based on the difference in oktas with the CC algorithm compared to manual observation.	53
Table 6.2	How the best performing classification method of Hartmann [2020] is made	54

LIST OF ACRONYMS

UV	Ultra Violet	3
DNI	Direct Normal Irradiance	6
DHI	Diffuse Horizontal Irradiance	6
GHI	Global Horizontal Irradiance	6
AM	Air Mass	5
a	Altitude angle	4
A	Azimuth angle	4
CSI	Clear Sky Irradiance	7
K_c	Clear Sky Index	7
PV	PhotoVoltaic	8
NWP	Numerical Weather Prediction	9
GFS	Global Forecast System	10
ECMWF	European Centre for Medium-Range Weather Forecasts	10
MM5	Fifth-Generation Penn State/NCAR Mesoscale Model	10
WRF	Weather Research and Forecasting	10
ASI	All Sky Images	10
PM	Persistence Model	37
SPM	Smart Persistence Model	37
AR	Auto Regressive	11
I	Integrated	11
MA	Moving Average	11
ARIMA	Auto Regressive Integrated Moving Average	2
SVR	Support Vector Regression	21
CC	Cloud Cover	14
RGB	Red-Green-Blue	12
HSV	Hue-Saturation-Value	12
SYNOP	(surface synoptic observations)	15
SPS	Supapixel Segmentation	17
DL	Deep Learning	1
ML	Machine Learning	20
AI	Artificial Intelligence	20
SVM	Support Vector Machine	21
SVR	Support Vector Regression	21
NN	Neural Networks	24
ANN	Artificial Neural Networks	24
CNN	Convolutional Neural Networks	1
kNN	k-Nearest Neighbor	22
ReLU	Rectified Linear Unit	26
SGD	Stochastic Gradient Descent	29
UTC	Coordinated Universal Time	32
ADF	Augmented Dickey Fuller	11
TP	True Positive	36
FN	False Negative	36
FIFO	First In First Out	37
MAE	Mean Average Error	38
STC	Standard Test Conditions	6
ADF	Augmented Dickey Fuller	11
MSE	Mean Squared Error	29
RMSE	Root Mean Squared Error	40

nMAE normalized Mean Average Error	40
nRMSE normalized Root Mean Squared Error	40
FS Forecast Skill	40
Adam Adaptive moment estimation	29
NWS National Weather Service	46
CI Clearness Index	47
VI Variability Index	36

2

INTRODUCTION

Reason of research: Over the last ten years, the capacity of photovoltaic installations has increased exponentially. In 2012, the total worldwide capacity was around 101 GW of electricity capacity. In the end of 2022, the total worldwide capacity is almost 1,200 GW [IEA-PVPS \[2017\]](#). The global electricity generation capacity is expected to be around 14,700 GW by 2050, from which is expected that around 45% of the renewable energy generation is from solar [EERE \[2022\]](#). This indicated that the growth is already phenomenal, but expected to increase much more in the upcoming years. Due to the increase in photovoltaic generated electricity, the demand for making accurate irradiance predictions also increases. This is mainly due to the intermittency of the solar energy that gets generated. The solar energy is at every location on Earth different and does not generate energy all day long on a specific location. On top of that, the biggest factor apart from the location of the sun that can change the outcome of the solar energy at a specific point on earth are clouds. It is critical to know how much solar energy can be generated to electricity in order to maintain grid stability, optimise energy control tactics, and create energy trading algorithms.

Since the solar energy is location dependent and can change abruptly, it is necessary to make very short term irradiance prediction and on a small spatial (few hundred metres) and temporal scale. All sky images can be used to make short term irradiance prediction on a small spatial and temporal scale. The use of all sky images to make short term irradiance has gained popularity over the last recent years due to promising prediction results.

In order to train Deep Learning (DL) models for irradiance predictions, patterns must be extracted from both single point data and sky images. Convolutional Neural Networks (CNNs) are a particular class of Neural Network that can handle 2D inputs like images [LeCun et al. \[1989\]](#). In addition to Convolutional Neural Networks (CNN), Convolutional Neural Network + Long Short-Term Memory network (CNN+LSTM) [Siddiqui et al. \[2019\]](#) and 3D convolutional neural network (3D-CNN) are two more advanced model designs that can be used to extract relevant data from a series of all sky images [Wei et al. \[2019\]](#).

Apart from the short term irradiance predictions, the development and deployment of all sky image devices has made it possible to continuously gather data on the cloud cover state. [Kazantzidis et al. \[2017\]](#) presented a method for the detection of clouds and the calculation of the entire cloud coverage, given that colour is the main characteristic that enables the visual recognition of clouds on the sky. [Xie and Yiren \[2019\]](#) used an CNN model with an labeled training set of the cloud cover for the all sky images to made an indication of the cloud cover in the test set. The calculated cloud cover can be used as an direct input in the DL model ([Zuo et al. \[2022\]](#)) or can be used to classify the ASI in to groups based on sky conditions [Kazantzidis et al. \[2012a\]](#). When the location of the clouds in the image are known, [Chow et al. \[2011\]](#) measures the temporal dynamics of the clouds that the camera sees, calculates their current trajectory, and predicts where the clouds will be in the future.

This study made use of the prior work of [Doodkorte \[2021\]](#), which made an CNN prediction model for 21-minutes. From this research became clear that predicting the interaction between solar irradiance and clouds in the sky remains a complex

problem, especially for partly clouded conditions. To obtain more insight and a possible improved solution to this problem the following question will be answered:

What is the maximum improvement of the CNN 21-minute irradiance predictions model when making classification based on sky conditions first?

Aim of the research: The goal of this research is to first classify the all sky images into five separate groups based on sky conditions. These separate groups will then make 21-minute irradiance predictions. The reason for 21-minute predictions is that for energy trading, the electricity can be traded 5 minutes before delivery of a 15-minutes contract. Since there is some expected time to make the prediction, the total prediction time is chosen to be 21-minutes. These irradiance predictions will be done for the location Delft in the Netherlands and for Folsom in the United States, which is made publicly available by [Pedro et al. \[2019\]](#). Four prediction models will be used to make the 21-minute predictions. The first model is the best performing (lowest Mean Average Error over 21-minutes) DL model from the work of [Doodkorte \[2021\]](#). This model is an CNN model with as input the original measured irradiance and the clear sky irradiance. The other three models are the Persistence model, the Smart Persistence model and the Auto Regressive Integrated Moving Average (ARIMA) time series model. Apart from the research question various other contributions are:

- A comparison of four different 21-minute prediction models outcomes for two different locations.
- A classification algorithm that can make five separate groups based on the cloud cover and clear sky index.
- The 21-minute predictions will get minutely projected and compared based on the MAE and quantile 95% based on the MAE.
- The 21-minute CNN model will get compared with: 1. the CNN model with cloud cover as extra input, 2. the CNN model but then extra images added to the dataset and 3., when HSV images are used instead of the RGB images as input.

Structure of the research: The structure of this research is as follows. First in Chapter 3, the needed background information will be explained in order understand the classification process and the prediction models that are used. After that in Chapter 4, the classification process and the final prediction models are shown. On top of that, a performance evaluation is made for the classification method as the prediction models. For those two methods the results will get presented in Chapter 5. Finally, this research gets discussed and the conclusion will be given in Chapter 6.

3 | BACKGROUND

3.1 METEOROLOGY

The layers of gases that surround a planet or other celestial body are referred to as atmosphere. The atmosphere guards life on Earth from incoming Ultra Violet (UV) radiation, maintains the Earth warm through insulation, and suppresses temperature extremes between day and night [Brown \[2022\]](#). The study of the atmosphere, atmospheric phenomena, and atmospheric influences on our weather is known as meteorology. Meteorology, focuses on atmospheric factors connected to present or near-future situations, up to multiple days ahead [Coleman and Law \[2015\]](#). There are multiple meteorological elements describing the atmosphere such as solar energy, air pressure & temperature, humidity, precipitation amount and type, wind direction and strength, and cloud cover. The Earth's atmosphere is composed of around 78% nitrogen, 21% oxygen, and 1% other gases. These gases are present through five different layers: troposphere, stratosphere, mesosphere, thermosphere, and exosphere, as shown in Figure 3.1. These five layers are characterised by distinct characteristics such as temperature and pressure. The Exosphere is the layer furthest from the earth, with a distance between 700 to 10,000 km. After that, the thermosphere, mesosphere and stratosphere follows. The layer that starts at ground level with the earth is called the troposphere. Practically all influences on solar radiation before reaching earth in the troposphere. It contains 99 percent of the water vapour of the atmosphere, hence most clouds form here [Russel \[2015\]](#).



Figure 3.1: The five layers of the Atmosphere, with the given height and temperature range per layer. Image from [Annsky \[2022\]](#).

3.2 SUN

The Sun is a star at the center of our solar system. Without the Sun, there would be no life on earth. Scientists studying the beginnings of life on Earth refer to Earth's optimal positioning in relation to the Sun as a necessary component in creating the conditions that allow life on Earth. The presence of the Sun is important due to the radiation it releases [Buis \[2020\]](#).

3.2.1 Position of the Sun

The Earth makes one rotation around its own axis every 23 hours and 56 minutes while it also makes one full rotation around the sun in 365 days, 5 hours, 59 minutes and 16 seconds [May \[2021\]](#). This means that the position of the Sun in relation to the earth changes all the time. As shown Figure 3.2, the Sun is the object that needs to get located from the observer point. The observer point is a specific point on earth, for Example Delft, the Netherlands. From observer (location on earth) point of view, the object (the Sun) is moving over the Celestial sphere during the day. The celestial sphere is an imaginary sphere with the earth at its center. When this celestial sphere goes specifically through the North and South it is called the Meridian. In order to find the location of the object from the observer point, two angles must be used. The first one is the 'Altitude angle (a)' (also called the elevation angle), which is the angular elevation of the sun above the horizontal plane. The angle range is between 90° and minus 90° , where the negative values are below the horizon so not visible for that specific location at the observer point. When a is 90° it is at Zenith, which corresponds to the point in the sky where the vertical intersects the celestial sphere. The other angle is the 'Azimuth angle (A)', which is the angle formed by the projected line of sight on the horizontal plane [Smets et al. \[2015\]](#). The angle range is between 0° and 360° . The angle increases clockwise and start North at 0° . So, 90° , 180° and 270° , represents the East South and West.

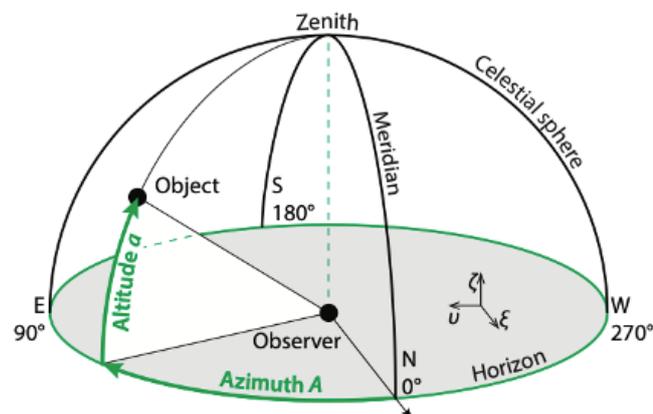


Figure 3.2: Illustration on how to find the location of the sun in relation to the earth. The Observer is the specific location at Earth and the Object is the sun. In order to do this the Altitude (a) angle and Azimuth angle (A) must be known [Smets et al. \[2015\]](#).

When the location of the sun can be found, the total sun path for a whole year on a specific location can be found. In Figure 3.3, the path of the Sun with respect to the A angle and Zenith angle (θ) is shown. The moment of the year has influence on the amount of daylight received along a specific location during a finite period of time. For the location Delft, is shown that the sun path in the summer (2022-07-01), is much longer compared to the sun path in the winter (2022-01-01). Apart from that, the analemma loops are shown. A Solar analemma loop is a plot that depicts

the position of the Sun in the sky at a single place and time of day over the course of a year, shaped like the number 8 [Kher \[2022\]](#).

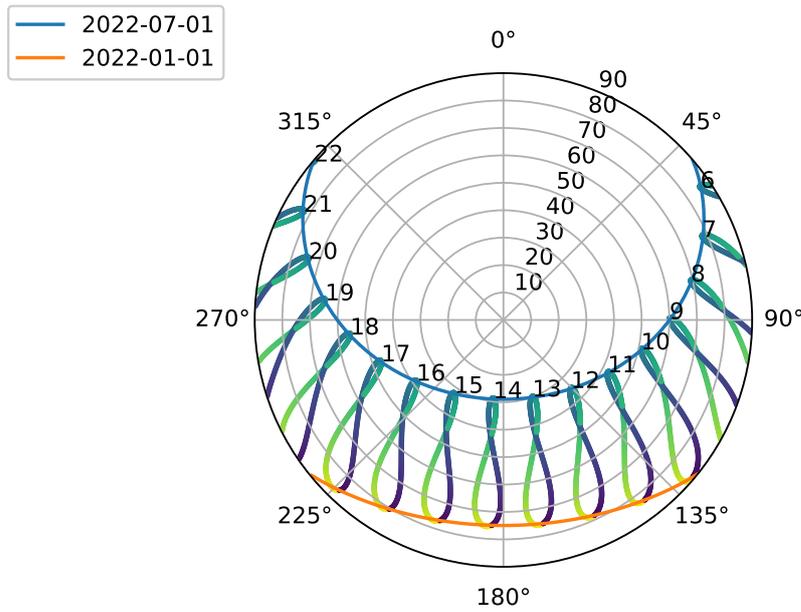


Figure 3.3: The sun path of two specific days the at the location Delft, the Netherlands. The inner circles represents the radius at a specific Zenith angle (θ) and the circles turned clockwise represents the Azimuth angle (A). Figure is made with Python 3.9

3.2.2 Solar radiation

Solar radiation is the energy released by the Sun and transmitted as electromagnetic waves across space. The Sun emits energy in the form of short-wave radiation, which is attenuated in the atmosphere by clouds and absorbed by gas molecules or suspended particles. The spectral irradiance is the power density at a particular wavelength. Combining the whole range of wavelengths with this spectral irradiance is the solar radiation. Solar radiation reaches land surfaces after travelling through the atmosphere and is reflected or absorbed [EERE \[2021\]](#). The atmosphere influences the final irradiance outcome that reaches Earth. The distance that sunlight must travel through the atmosphere to reach the surface of the earth is the most crucial factor that controls the solar irradiance under clear sky circumstances [Smets et al. \[2015\]](#). When the Sun is at its zenith (directly overhead), this distance is the shortest. The optical Air Mass (AM) is defined as the ratio of the sunlight's actual travel length to this close distance. The spectrum is known as the AM_1 spectrum when the Sun is at its zenith ($\theta = 0$) and the optical air mass is unity. The spectrum outside the atmosphere is called the AM_0 spectrum, since no atmosphere is traversed there [Smets et al. \[2015\]](#). In Equation 3.1, the AM spectrum formula is shown.

$$AM := \frac{1}{\cos \theta} \quad (3.1)$$

This Equation, however, does not take the curvature of the Earth into account [Smets et al. \[2015\]](#). This Equation indicates that solar radiation varies in strength and spectral distribution depending on the location on the Earth and the Sun's position in the sky. The reference solar spectral distribution is the $AM_{1.5}$ spectrum, as described in International Standard IEC 60904-3 [60904-3 \[2008\]](#). This $AM_{1.5}$ is used as

the Standard Test Conditions (STC), together with the irradiance of $1000\text{W}/\text{m}^2$ and a cell temperature of 25°C Smets et al. [2015].

This atmospheric interaction results in two different forms of irradiance. The first one is the Direct Normal Irradiance (DNI), which is the measured irradiance at the surface perpendicular (normal) from the sun, so only the irradiance from a straight line of the sun to the surface. The second form of irradiance is Diffuse Horizontal Irradiance (DHI), which is all the irradiance measured due to scattering in the atmosphere Vignola et al. [2017]. A pyranometer is an instrument used to measure the GHI, as shown in Figure 3.4. The pyranometer has multiple thermopile temperature sensors, which are sensors that can measure temperature from distance. Due to the difference in temperature between the sensors, the radiation intensity can get calculated. With these sensors the pyranometer can measure the incoming GHI at the specific location Stoffel et al. [2010].

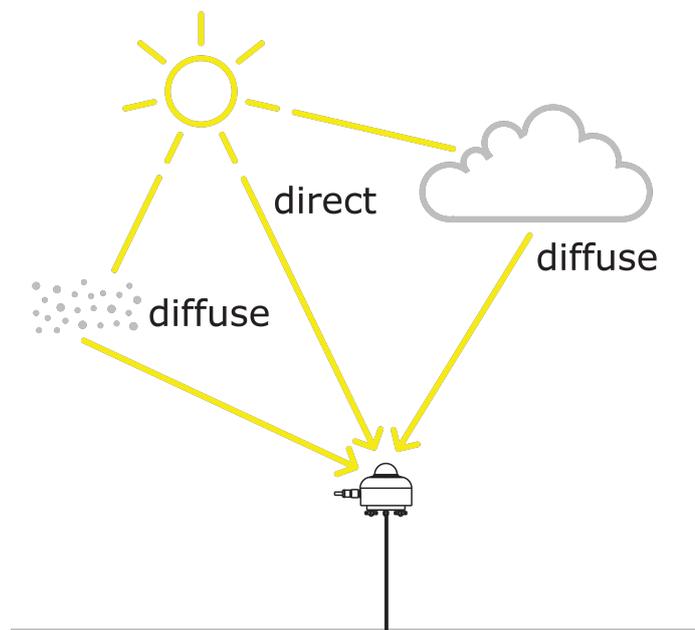


Figure 3.4: The Direct Normal Irradiance (DNI), which is the irradiance perpendicular to the sun and the Diffuse Horizontal Irradiance (DHI) which is the reflected irradiance on that same point. Combine the DNI and DHI and the Global Horizontal Irradiance (GHI) can be found and measured by the pyranometer. Figure obtained from Hukseflux [2006]

When combining the DNI with the altitude of the sun and the DHI, the Global Horizontal Irradiance (GHI) can be calculated as shown in Equation 3.2.

$$GHI = DNI * \cos(a_s) + DHI \quad (3.2)$$

The GHI is the total irradiance from the sun coming at a specific point from a horizontal surface on the earth. The biggest factor apart from the location of the sun that can change the outcome of the GHI at a specific point on earth are clouds. The measured GHI can change up to 70 % in a time range of only 5 seconds Barbieri et al. [2017]. Therefore, it is discussed in more detail in the following section 3.3. First, the situation when the clouds are neglected will get further explained in the next subsection 3.2.3.

3.2.3 Clear Sky Irradiance

The absence of visible clouds throughout the entire sky dome is expressed as a clear sky condition, and Clear Sky Irradiance (CSI) is the irradiance (GHI) that occurs during these conditions [Reno and Hansen \[2015\]](#). The CSI is in research widely used since it can be used as a reference value to the real GHI and to make irradiance predictions, which will be explained in Section 3.4. The most common CSI models are empirical models, which indicates that it uses solar geometry inputs and atmospheric parameter inputs to model the expected CSI at a specific location [Antonanzas-Torresa et al. \[2019\]](#). There are many methods that calculate the expected CSI. That is why [Antonanzas-Torresa et al. \[2019\]](#) reviewed seventy CSI models and verifies the results based on the irradiance in clear sky condition of two meteorological stations. One of the best performing models is the Simplified Solis [Ineichen \[2008\]](#), which will be used in this research. The input for this model consist of the location and the time frame. The output then gives the predicted CSI at the given location for that given time frame. In Figure 3.5, an example is given of the CSI outcome in comparison with the actual GHI for that day.

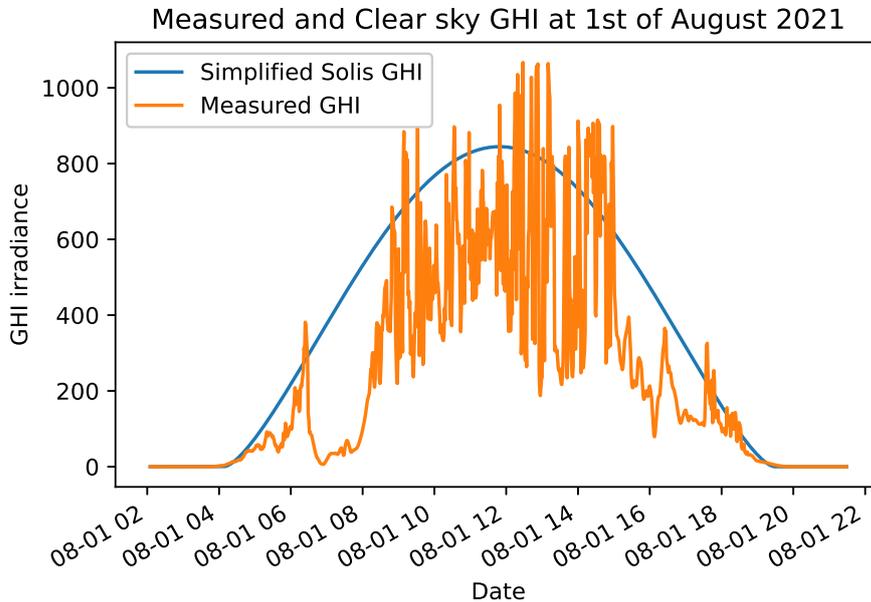


Figure 3.5: The Clear Sky Irradiance (CSI) for the first of August 2021 in Delft projected against the real GHI during that day. Figure is made with Python 3.9.

In Figure 3.5, one can see that the Measured GHI is sometimes higher than the CSI. This can happen when there are partly cloudy conditions. This means that the sky is partly clear and partly clouded. The influence of the clouds can increase the measured GHI in combination with the solar elevation angle. When photons scatter from water droplets in clouds close to the direct irradiance route, enhanced irradiance zones are created. This phenomena is called cloud enhancement and when this occurs the measured GHI can give higher outcomes than the clear sky irradiance [Jarvela and Kari Lappalainen \[2018\]](#). With the CSI and the measured GHI, one can calculate the Clear Sky Index (K_c), which is the measured GHI at ground level divided by the estimated GHI in clear sky conditions at ground level [Kato \[2016\]](#), as shown in Equation 3.3.

$$K_c(t) = \frac{GHI(t)_{measured}}{GHI(t)_{CSI}} \quad (3.3)$$

The K_c will become important in irradiance forecasting, which will get explained in Section 3.4.

3.3 CLOUDS

Clouds are formed of floating water drops or ice crystals in the sky. The water drops are too little to see. They have decomposed into a gas known as water vapour. The air becomes cooler as the level of **water vapour** in the atmosphere rises. The colder air encourages water droplets to adhere to substances like **dust**, ice, or sea salt [May \[2017\]](#). The development of small globules is called the creation of **cloud droplets**, which occurs when water droplets attach to dust. Cloud droplets are light in weight and accumulate as they drift, merging with air to form the frothy shapes we see hung in the sky known as **clouds** [Zuckerman \[2019\]](#). The steps involved in turning water vapour into a cloud are also shown in Figure 3.6.

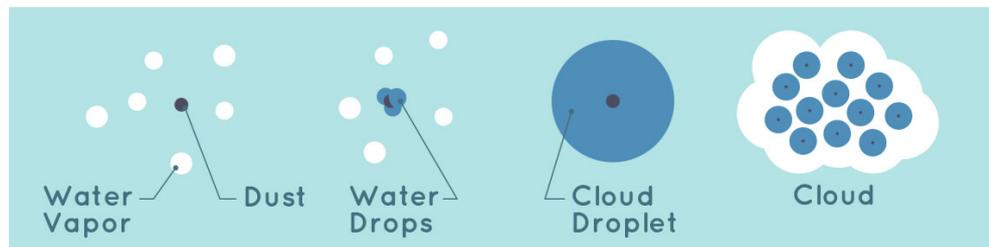


Figure 3.6: The steps that are made to become a cloud from a water vapor. Air becomes cooler and adhere to dust, then cloud droplets are made. The light cloud droplet accumulate and clouds are created [Novati \[2022\]](#).

Clouds exist in all shapes and sizes. Clouds get either classified based on the height of the clouds above the Earth's surface (high, intermediate or low), or classified (and named) on their distinguishing features. The group of high clouds are between 5 to 13 km above the Earth's surface, the middle group between 2 to 7 km and the low group between 0 and 2 km above the surface [UCAR \[2019\]](#). As already stated, the names of the clouds refers to the type of clouds. From this four types can be divined. The first variety is called 'Cirro,' which is a Latin term that means "hair curl." Cirro-form clouds are whitish and hair-like clouds consisting largely of ice crystals that are usually high clouds. Strato-form clouds are named after the Latin word for "layer" because they are frequently broad and spread out, resembling a blanket. The third variety is referred to as a 'Nimbus,' which is the Latin word for rain. The final variety is called 'Culumo,' which means 'pile up,' and it resembles white fluffy cotton bottom balles and demonstrate vertical motion or thermal uplift of air in the atmosphere [NWS \[2022b\]](#). Based on the height and the four cloud characteristics, the clouds can be classified into ten groups, as shown in Figure 3.7.

3.4 IRRADIANCE FORECASTING

Solar irradiance forecasting is an essential topic in renewable energy generation. Forecasting of solar irradiance increases the photovoltaic system development and operation while providing numerous economic benefits to energy companies [Alzahra-
nia et al. \[2017\]](#). Since it is expected that in 2050 around 45% of the renewable electricity generation is from solar, the importance of solar irradiance forecasting increases as well [EERE \[2022\]](#). The forecast horizon can be categorized into four groups, as shown in Figure 3.8. Long term irradiance prediction is from a couple of month until years ahead. Long term GHI prediction is crucial for PhotoVoltaic (PV) power plant site selection and installation [Coimbra and Kleiss \[2013\]](#). Medium-term

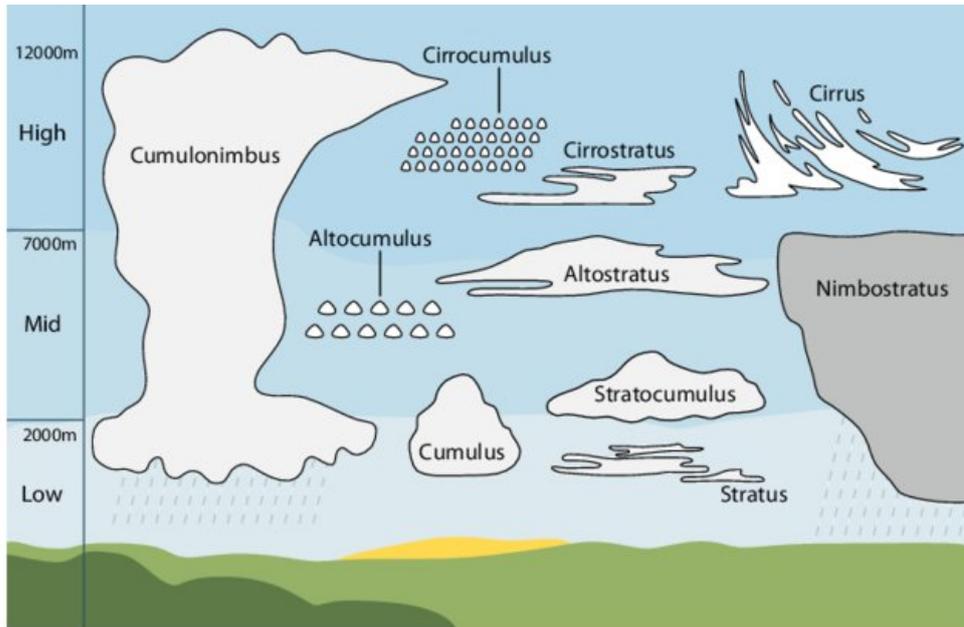


Figure 3.7: The ten different cloud classification illustrated. The classification is done based on height and cloud characteristics. Palubicki [2020]

forecasting, which predicts for around one week in advance, assists in making power plant maintenance schedules. Short-term forecasting, with a time step of 48 to 72 hours, is critical for decision making in areas such as demand, supply balance and unit commitment. The final group is very short term irradiance predictions, which is essential for system monitoring, so these predictions are a few seconds up to a couple of minutes Kumari and Toshniwal [2021]. In this research the focus lays on the very short term irradiance predictions.

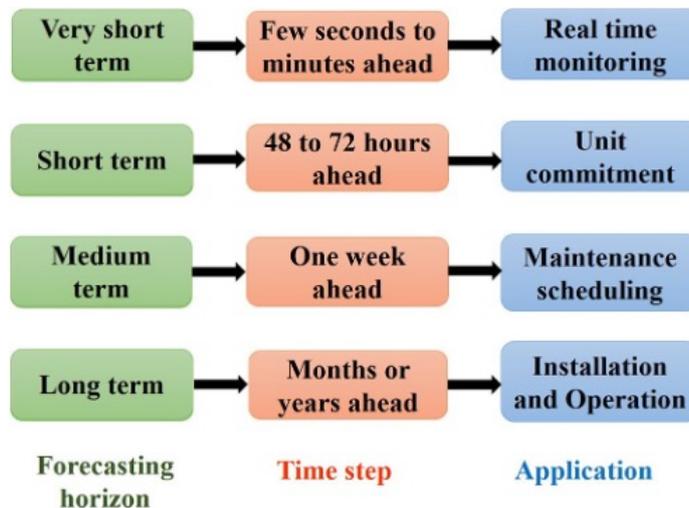


Figure 3.8: The four forecast horizon for irradiance forecasting with the time step and the used application. Kumari and Toshniwal [2020]

Apart from the duration of the prediction, the spatial scale of the area also needs to get established. Different methods can be used regarding the spatial scale. For global scale, Numerical Weather Prediction (NWP) is the most common method. NWP models explain the main physical processes in the atmosphere, surface, and in the soil. They account for their impact on the temporal evolution of model variables such as pressure, temperature, wind, water vapour, clouds, and precipitation

Visser et al. [2022]. Two examples are the Global Forecast System (GFS) and the European Centre for Medium-Range Weather Forecasts (ECMWF). On mesoscale (1 to 100 km) level, two NWP methods are commonly used. The Fifth-Generation Penn State/NCAR Mesoscale Model (MM5) employs a terrain-following coordinate, solves finite-difference equations using a time-split approach, and supports multiple nesting. The other one is the Weather Research and Forecasting (WRF), which is supported as a community model under ongoing development and incorporates characteristics from other mesoscale models, notably MM5. WRF can be thought of as a follow-up model to the MM5 Diagne et al. [2013]. Another method for mesoscale predictions is the use of satellite images and meteorology data Perez et al. [2010]. The final scale is microscale (0 to 1 km) and the two most common methods for making short term irradiance predictions are statistical models or All Sky Images (ASI) based models, in combination with meteorological data. The final overview is shown in Figure 3.9. Since the goal of this research is to make short term irradiance prediction on microscale, the time series models and all sky images will get a broader explanation in respectively subsection 3.4.1 and 3.4.2.

Prediction of solar irradiance

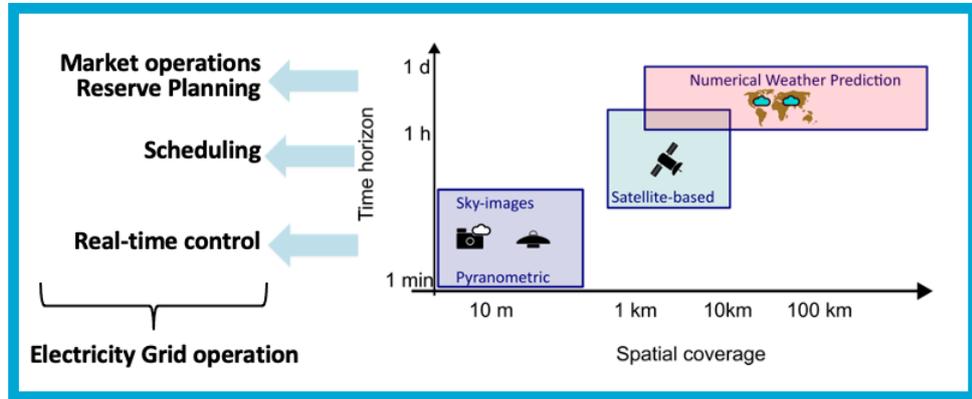


Figure 3.9: Overview of the prediction of solar irradiance methods with their spatial coverage and time horizon made by V.A. Martinez Lopez. Each time horizon has their own goal. Image based on work of Blanc [2017], Kazantzidis [2017] and NREL [2016].

3.4.1 Statistical models

A statistical model is a collection of assumptions about the probability distribution that produced some observed data Taboga [2021]. Statistical models use historical time series solar irradiance data to develop forecasting models Aguiar and Díaz [2018]. Besides the historical time series of the solar irradiance, also other time series data could be added which are correlated to the future outcome of the solar irradiance. These correlated time series are meteorological parameters that have influence on the measured solar irradiance. The models that only take the historical solar irradiance into account will be handled. Statistical time series models are regression lines based on the historical data points it received. The most straightforward is the simple linear regression model, with the forecast variable y_t and a single predictor variable x_t .

$$y_t = \alpha + \beta_1 x_t + \varepsilon_t \quad (3.4)$$

The coefficients α and β_1 represent the line's intercept and slope and the ε_t the random error Hyndman and Athanasopoulos [2018]. When taking more than one

predictor variables, the regression becomes a multiple regression model and when the relationship between the forecast variable and predictor variable becomes non-linear, the regression model becomes non-linear.

A baseline model is required for any time series forecasting situation. The performance baseline offers a sense of how well all other models will perform on the problem. The guidelines of the baseline model is that it should be simple (does not require much training time or intelligence), fast (to implement and requires little computational effort) and repeatable (generates the same predicted output given the same input) [Brownlee \[2016\]](#).

On top of the question of whether time series is a linear or non-linear line and a multiple regression model or not, it is important to consider whether the time series is stationary or differencing. A stationary time series is one whose statistical characteristics are independent of the observation time. A stationary time series will not exhibit any long-term predictable trends [Hyndman and Athanasopoulos \[2018\]](#). Stationarity indicates that the mean and variance do not change over time. When the time series is stationary, it is expected that the prediction accuracy will increase [Hyndman and Athanasopoulos \[2018\]](#). In reality, this is not always the case for the time series. For that reason, when the time series is non-stationary, differencing will be used. With differencing the differences between successive observations can make a non-stationary time series stationary [Hyndman and Athanasopoulos \[2018\]](#).

A common used model which implements differencing for time series analysis is the [ARIMA](#) model. The final ARIMA model has three variables. $ARIMA(p,d,q)$, where p is the number of lag observations, d is the number of times the observations are differentiated and the q is the size of the moving average window [Brownlee \[2017\]](#).

The three parts of the $ARIMA(p,d,q)$ will get explained one by one. The first one is the Auto Regressive ([AR](#)) model, which checks if the variable in the time series gets regressed by itself over time, as shown in equation [3.5](#).

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p} + \epsilon_t \quad (3.5)$$

Where α is the constant and ϵ is white noise. White noise is a discrete signal with zero mean and finite variance [Carter and Bruce \[2009\]](#). The Y_t is a multiple regression but with lagged values of as predictors. It can be written as an $AR(p)$ model, an AR model of order p . t is the number of time steps [Hyndman and Athanasopoulos \[2018\]](#).

The second part of the model is the Integrated ([I](#)). $I(p)$ is the amount of times that the $ARIMA$ model gets differentiated. The purpose of differencing it to make the time series stationary. These variable p can be found with the Augmented Dickey Fuller ([ADF](#)) method. The ADF method is a common statistical test used to test whether a given Time series is stationary or not [Prabhakaran \[2019\]](#). The null hypothesis of the ADF is that the time series is non-stationarity. If the null hypothesis is rejected, the $ARIMA$ model will get differentiated and the ADF test will be done again. If it is concluded that the time series is stationary, the test's p -value (exceedance probability) is less than the significance level (0.05).

The last part is the Moving Average ([MA](#)), which uses the average over the historical time series. $MA(q)$ is the amount of historical time 'q' that will be taking into consideration as used moving average for making predictions [Hyndman and Athanasopoulos \[2018\]](#). The formula is shown in Equation [3.6](#).

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \dots + \phi_q \epsilon_{t-q} \quad (3.6)$$

With ϕ the weight of the specific time step and each number of MA(q) can be viewed as a weighted moving average of the previous several predicted errors. Now that the three parts of the total model are explained separately, the final ARIMA(p,d,q) formula can be seen in equation 3.7.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p} + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \dots + \phi_q \epsilon_{t-q} \quad (3.7)$$

The outcome of this method is the future values that are getting estimated with the forecasted errors (AR) and a linear combination of previous measured values (MA) [Hans \[2019\]](#). This method is non-stationary, which means that the mean and variance changes over time.

The ARIMA model is also used for making irradiance predictions. In the paper of [Kanagasundaram et al. \[2019\]](#), they compared the long short-term memory (LSTM) method which is a Deep Learning method to the outcome of the ARIMA time series for long term irradiance predictions. The outcome was that the LSTM approach outperformed the ARIMA time series with only 2% when the Root Mean Squared Error (RMSE) values gets compared. Comparing the RMSE is a common method to calculate and compare the error metric for numerical predictions. Moreover, [Hans \[2019\]](#) did a comparison for the short term irradiance prediction among other the ARIMA model. The prediction were for 15 minutes, 1 hour and 3 hours. The outcome was that the Persistence Model (PM) outperforms the ARIMA model for those three time frames. The PM uses the most recent time step to make predictions over future time steps. So, for the irradiance predictions it uses the most recent GHI time step to make future GHI predictions. In this paper a temporal resolution of 1 minute was used.

3.4.2 All sky images

All sky images (ASI), in other papers sometimes called whole sky images, are images that are being made with a fish-eye camera directed perpendicular to the sky. Normally, photographs are then taken at a fixed time interval. ASI can be used for making irradiance forecasting and to make weather classification based on the ASI's, which will get further explained in Section 3.5.

To be able to use the ASI, it is important to understand the basic principles of images and how they are interpreted by a program. An image is build up into pixels. A pixel is in digital imaging the smallest 2-dimensional (squared) grid item that can hold some specific information of an image [Zhang and Gourley \[2009\]](#). The value of that specific pixel is dependent on the color representation method. Two of the most widely used color representation methods in computer graphics are Red-Green-Blue (RGB) color space or the Hue-Saturation-Value (HSV) color space. The RGB pixel can be visualized as a 3D-square which consist of 3 values between [0 : 255] in the Red, Green and Blue spectrum as shown in Figure 3.10(a). This would indicate that a red pixel is (255, 0, 0), a green pixel (0, 255, 0) and a fully blue pixel (0, 0, 255). For a gray scale (black-white) image the pixel would range between [0 : 255] in only one dimension, namely the gray dimension. The HSV scale gives a numerical reading of the image that matches the color names it contains [Chin \[2021\]](#). Hue is measured in degrees from 0 to 360, Value and Saturation range both from 0 : 100, which gives the visualized pixel shown in 3.10(b). So, both colored pixels are 3 Dimensional. When the saturation value is zero, the image would appear as a grayscale image.

All the pixels combined form the ASI. The final implementation of the ASI that gets converted to an RGB matrix for programming in Python is shown in Figure 3.11. Note that the X-direction 'Width' is from left to right and the Y-direction 'Length'

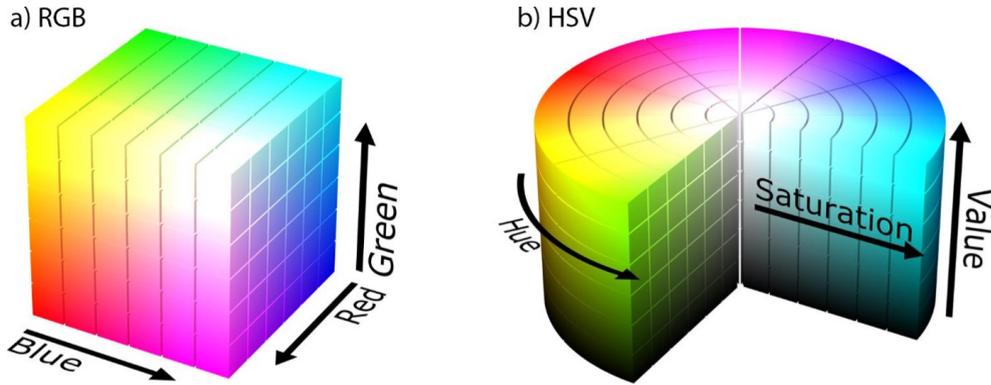


Figure 3.10: The visualized difference between (a) the RGB pixel and (b) the HSV pixel, image from Popov et al. [2018].

from top to bottom. The final matrix of an image is dependent on the size of the image. The matrix structure is namely: (Number of pixels in row, Number of pixels in column, Number of dimensions), which is in the ASI example of Figure 3.11: (1536, 1536, 3). In this example the number of dimensions in an RGB image is equal to three.

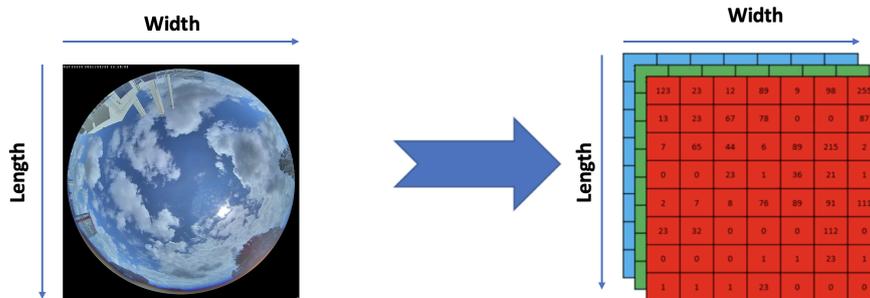


Figure 3.11: The ASI that gets converted into an RGB matrix which can be used for programming in Python.

The solar irradiance forecast is more difficult on cloudy days because of the large fluctuations in the solar irradiance at very low levels, that is why implementing ASI in irradiance predictions are interesting Gao and Liu [2022]. In these ASI the clouds can get detected and improve the model's predictions. On top of that, the movement of clouds from the ASI can also be found by employing digital image processing technologies and computing the displacement of clouds in the photographs of the sky Kunert et al. [2016]. All these implementations of the ASI's are done with various deep learning techniques which will get further explained in Section 3.7.

3.5 WEATHER CLASSIFICATION METHODS

There are many methods to classify the sky conditions in specific groups. As explained in Section 3.3, the clouds can be classified into ten different groups based on their height and cloud characteristics only. Heinle and A. Macke [2010] found a way to make the classification of the ten different clouds based on ASI's, to combine some classification in the same group, for that reason consisting of seven groups instead of ten. This was done by choosing twelve specific image features to find the differences between the classified groups. The features are the Red and Blue mean of the RGB image. Furthermore the Blue (from the RGB image) Skewness, Energy,

Entropy, Contrast, Homogeneity. Then the difference in Red-Blue, Red-Green and Green-Blue. Finally the last feature is the Cloud Cover. This method was also applied in the paper of Kazantzidis et al. [2012a], where they improved the Cloud Cover (CC). The CC is the percentage of detected clouds in an image, as shown in Equation 3.8.

$$CC = \frac{\text{Cloud Pixels}}{\text{Cloud Pixels} + \text{Sky Pixels}} \quad (3.8)$$

The background mask of the image is not taken into account when calculating the CC. The background mask are the pixels in the ASI which are neither sky or cloud. This could be all other things that disturb the ASI and the mask of the location at TU Delft is shown in Figure 3.12.



Figure 3.12: The mask layer of the ASI at the TU Delft. These mask layer will be used on top of the ASI to calculate the final CC. The mask layer will not be taken into account for calculating the final CC.

Based on the CC, the ASI's can get separated into clear sky images, partly sky images and overcast images. Apart from the CC percentage, another method to classify the amount of clouds in the sky is via oktas. Okta is a measurement method broadly used in meteorology which can describe the amount of cloud cover in the sky at a specific moment. 0 okta indicates a zero % CC and 8 oktas means a fully clouded sky so 100% Skull [2017]. The overview is shown in Figure 3.13

The reason for the use of oktas is that this can get classified well by human observation, whereas the exact percentage of the clouds in the sky becomes difficult. The total overview of how the CC can get converted to oktas is shown in Table 3.1.

Percentage of cloud	Okta
0	0
$0 < \% < 12.50$	1
$12.50 \leq \% < 25$	2
$25 \leq \% < 37.5$	3
$37.5 \leq \% < 50$	4
$50 \leq \% < 62.5$	5
$62.5 \leq \% < 75$	6
$75 \leq \% < 87.5$	7
$87.5 \leq \% < 100$	8

Table 3.1: overview of the cloud percentage can be converted to oktas.

Sky Cover (oktas)	Sym- bol	Name	Abbr.	Sky Cover (tenths)
0	○	Sky Clear	SKC	0
1	◐	Few* Clouds	FEW*	1
2	◑			2 to 3
3	◒	Scattered	SCT	4
4	◓			5
5	◔	Broken	BKN	6
6	◕			7 to 8
7	◖			9
8	◗	Overcast	OVC	10
(9)	⊗	Sky Obscured		un- known
(/)	☉	Not Measured		un- known

Figure 3.13: Overview of the different cloud covers expressed in oktas, from [Skull \[2017\]](#).

Different techniques can be used to calculate the CC. Common methods for finding the CC are image thresholding, clear-sky modeling and some machine learning methods like k-means segmentation, support vector machine and k-Nearest Neighbor. The first two methods will get explained one by one in this section, while the last three methods will be explained in Section 3.6.

3.5.1 Image thresholding methods

In order to study the outcome of an image, thresholding is commonly used. Thresholding is a sort of image segmentation method in which the pixels of an image are converted into binary pixels. This indicated that the outcome of the image is in black (for finding the CC this would be a no cloud pixel) and white (for finding CC this would be a cloud pixel). As previously explained in subsection 3.4.2, the pixel range is between $[0 : 255]$ for an RGB image. The value that makes the distinction between a white and black pixel when thresholding is called the threshold value. Multiple threshold techniques can be used for this. One of those threshold techniques is **fixed threshold**. For RGB images, a common fixed threshold method is to look at the differences between the Red-Green-Blue dimensions. [Long et al. \[2006\]](#) found the red-to-blue ratio of 0.6 as the fixed threshold value to find the CC in an ASI. More specifically, pixels are categorized as cloudy if their red-to-blue ratio (R/B) > 0.6 and as cloud-free if it is lower. There were about 100 ASI's used in the training image set to detect this ratio. [Kreuter et al. \[2009\]](#) founded a similar ratio of 1.3 between blue-to-red (B/R instead of R/B). So, $B/R > 1.3$ means cloud, lower no cloud. For this analysis 3903 images were used, and the outcome was that 73% of the images were in agree within one okta compared with the (surface synoptic observations) (SYNOP) observations. SYNOP are 3-hourly weather observation made by the World Meteorological Organization were they check the sky cover, state of the sky, cloud height and other meteorological data. Weather observations made by both manned and automated weather stations are reported using the numerical

code SYNOP. The calculated CC in the paper of [Heinle and A. Macke \[2010\]](#) was done with a fixed threshold of: $R - B > 30$, so clouded when higher than 30 and no cloud when lower than 30. Another technique that can be used is to normalize the image pixels according to their Red and Blue and then find a threshold value [Yamashita et al. \[2014\]](#). Normalization means that the RGB values are converted to values between zero and one instead of [0:255]. The used index is called the *sky index* and shown in Equation 3.9. Values less than 0.23 indicate the presence of clouds led to the creation of the clouds detection.

$$\text{Sky index} = \frac{B_{\text{pixel}} - R_{\text{pixel}}}{B_{\text{pixel}} + R_{\text{pixel}}} \quad (3.9)$$

By calculating the *Haze index* (Equation 3.10), [Crisosto \[2019\]](#) combined information from the green channel with that from the red and blue channels to improve cloud detection of the Sky index.

$$\text{Haze Index} = \frac{((B + R)/2) - G}{((B + R)/2) + G} \quad (3.10)$$

Fixed threshold techniques can also be used for HSV images. [Souza-Echer et al. \[2006\]](#) used the saturation value to find a mean and variance threshold value for detecting a cloud pixel. With this method a lower and upper threshold value was found to only detect ASI which are completely clear from clouds or completely clouded. The findings indicate that the method has an accuracy of better than 94% for clear sky ASI and 99% for overcast ASI in determining the CC state. Whereas [Jayadevan et al. \[2015\]](#) used the findings of [Yamashita et al. \[2014\]](#) to calculate the normalized saturation value for HSV images.

It is also possible to apply multiple thresholds in one image. This method is called the *multi color criterion*, since all the three image pixel ranges are taken into account. [Kazantzidis et al. \[2012b\]](#) The cloud pixel gets detected when:

$$R < 140 \& R > G + 70 \& R > B + 120.$$

It could be possible that the desired threshold is per ASI different, since the weather situation differs throughout the year. When instead of one ASI, there is a whole dataset for a year of ASI's available from which the CC will need to get calculated, other methods where the threshold value automatically differs per ASI could be used. One method that automatically calculates the threshold value per ASI is the **Otsu's threshold method**. The algorithm returns a single intensity threshold in its most basic form, dividing pixels into the foreground and background classes, or in the case of ASI into clouds and sky. This threshold is established by maximising inter-class variation or, alternatively, minimising intra-class intensity variance [Otsu \[1979\]](#). The classes refer to the colour distribution created in the image by the threshold technique. Finding this maximum inter-class variation is shown in Figure 3.14 (c). The maximum inter-class variation can be found by finding the optimal value based on the range of pixels and the amount of pixels per side. When for both sides this is maximized, the maximum inter-class variation is found and the threshold value can be calculated. This threshold technique is only usable for gray scaled images and was used by [Zhena et al. \[2017\]](#) to detect the clouds in the ASI, so the cloud forecasting could be predicted. Before using the Otsu method, it is important that the fish-eyed ASI gets flattened first. The fisheye image is a projection of a dome onto a plane, so the image is circular and distorted. Flattening an image means to remove such distortion leading to a standard rectilinear image. When this is not done, the pixel count of the black mask has a large influence on the on the distribution of the pixel intensity, which influences the final Otsu threshold

value. Due to this influence of the mask, the Otsu threshold value will appear lower compared to the flattened image where the influence of the mask is minimal. For that reason, the fish-eye ASI's first needs to get flattened. In Figure 3.14(a & b), one can see the difference between the outcome for an fish-eye ASI compared with an flattened ASI.

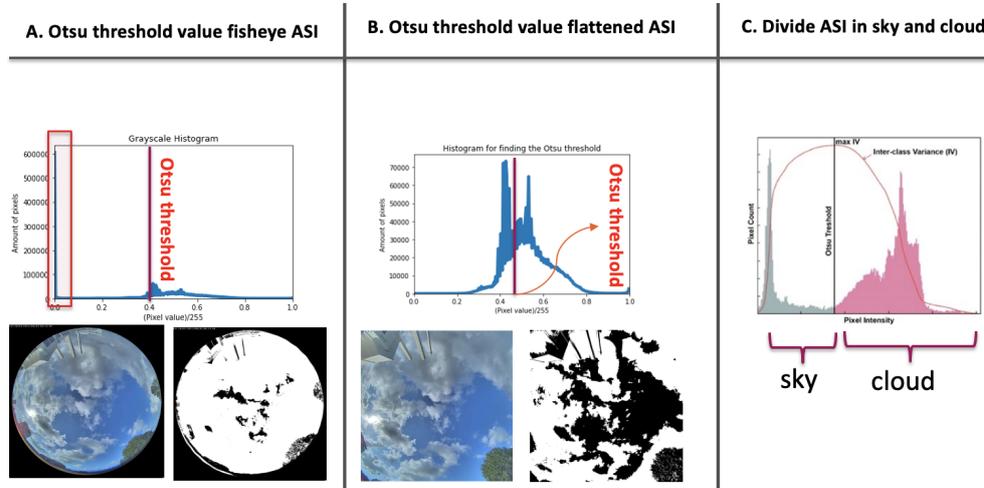


Figure 3.14: (A) The threshold value calculated for an fish-eye ASI and the outcome of the binary ASI. Note that the mask background is not yet taken into account. (B) The threshold value calculated for an flattened ASI and the outcome of the binary ASI. Note that the mask background is not yet taken into account. (C) Example of an gray scaled image where the maximum inter-class variation is found to find the specific threshold value for this image when using the Otsu threshold value Szelag [2020].

All the methods discussed up until now could be classified as **global threshold** methods, which indicates that the threshold method is the same value for the whole image. Apart from global thresholding, the threshold value can also differ throughout the image. The method when multiple threshold values will be used throughout the image is called the **local threshold** method. Liu et al. [2015] compared the global Otsu threshold method from Otsu [1979] with the fixed threshold method from Heinle and A. Macke [2010] and two local threshold methods. The first local threshold methods divided the image into 16 equally divided blocks and each local threshold will get calculated yang Jun et al. [2010]. The other technique is to divide an image into smaller blocks called superpixels. When applying local thresholding to each superpixel, results in the **Superpixel Segmentation (SPS)** method proposed by Liu et al. [2015]. This method will get explained in five steps and these steps are shown in Figure 3.15. The first step is that the fish-eyed ASI needs to get flattened first, for the same reason as previously explained for the Global Otsu threshold method. The second step is that the image will gets divided into local parts with the SPS technique. The SPS method splits an image into a number of intermittent blocks based on contour continuity, brightness similarity, and texture similarity Ren and Malik [2003]. For these 'superpixels' the threshold value needs to be found. For each superpixel the Otsu threshold value and the superpixel threshold value will be calculated. Since the Otsu threshold method only works for grayscale images, the third step is to grayscale the ASI before applying the threshold method. The final calculation for finding the threshold value is found with Equation 3.11.

$$T_{\text{local}} = (1/2)T_{\text{global}} + (1/2)T_{\text{Superpixel}} \quad (3.11)$$

So, for every superpixel that is made the final local threshold value is based on the superpixel threshold and the global Otsu threshold value of the whole image. The

fourth step is to add the flattened background mask on the thresholded ASI. This is to make sure that the background will not get categorised as cloud. The final step is to calculate the CC of the ASI.



Figure 3.15: The Superpixel segmentation method explained in 5 steps. First will the ASI gets flattened. Second will the SPS method by applied. Third the mask applied. Fourth the final threshold image is made. Finally the CC % can get calculated.

Each method has their own limitations. One of those limitations is that there is no difference being made between the thicknesses of the clouds, while the thickness can influence the final irradiance output arriving at the desired location. Furthermore, sometimes the clouds on the ASI appears somewhat more darker and for that reason that pixels could get classified as sky instead of clouds. On the other hand, the white sun could also get detected as clouds because of its white light. Also, the used camera's and aerosols differ per location when using this kind of techniques, for that reason, the specific optimal threshold values could differ per location/camera. Lastly, the pixel intensities change during the day as a result of shifting sun zenith angles, especially at the beginning and end of the sun when the sun comes up or goes down. Since the solar irradiance in that time frame is relatively low and the error high, it is common practice not to include this part in the final dataset in order to increase the accuracy.

When there is a dataset of multiple ASI's available, finding the CC for those images can be done with multiple method's. When multiple techniques are used for one dataset of images it is called **Hybrid thresholding**. Different threshold techniques can for example work better on ASI's where the sun is blocked or not [Zuo et al. \[2022\]](#). This final CC algorithm that is used in this research is explained in subsection [4.2.1](#).

3.5.2 Clear-sky modeling

Clear-sky modeling is another method in order to find the CC in ASI's. For clear-sky modeling, it is first necessary to create a library of clear sky background images with a range of solar elevation angles [Yang et al. \[2016\]](#). This would consist of images where only the sky would appear and no clouds. The next step is to add the location of the sun in the images. For that, first the position of the sun with the a & A angles needs to be calculated. Then the position of the sun needs to get translated to the X & Y coordinate in the ASI. The fisheye lens that is used for making the ASI's adopts an angular projection, which means the distance from the center of the image is proportional to the angle around the projection sphere [Schneider et al. \[2009\]](#). For that reason, the following Equation can be applied to find the X and Y coordinate.

$$\begin{aligned} X_{sun} &= X_{center} - R_{image} * \sin(a) * \cos(A) \\ Y_{sun} &= Y_{center} + R_{image} * \cos(a) * \cos(A) \end{aligned} \quad (3.12)$$

With X_{center} & Y_{center} as the center of the image, R_{image} the radius of the image and the altitude (a) and Azimuth (A) angles. In order to apply this Equation 3.12, it is important that the ASI is first flipped because the output of the image is mirrored with the reality. If the ASI is not flipped, then the West would be at the right side of the image and the East at the left side. Then the ASI gets rotated with the North as the highest middle point of the image, in order to correctly apply the Equation's of 3.12. Finally, the time of information of the sun angles needs to correspond with the time of the ASI that is used and the sun can be located as shown in Figure 3.16. However, it can only give a rough estimate of the true sun location due to the optics and mirror surface distortion [Alessandro Niccolai \[2020\]](#), which are unknown in the current work as well as in many industrial settings.

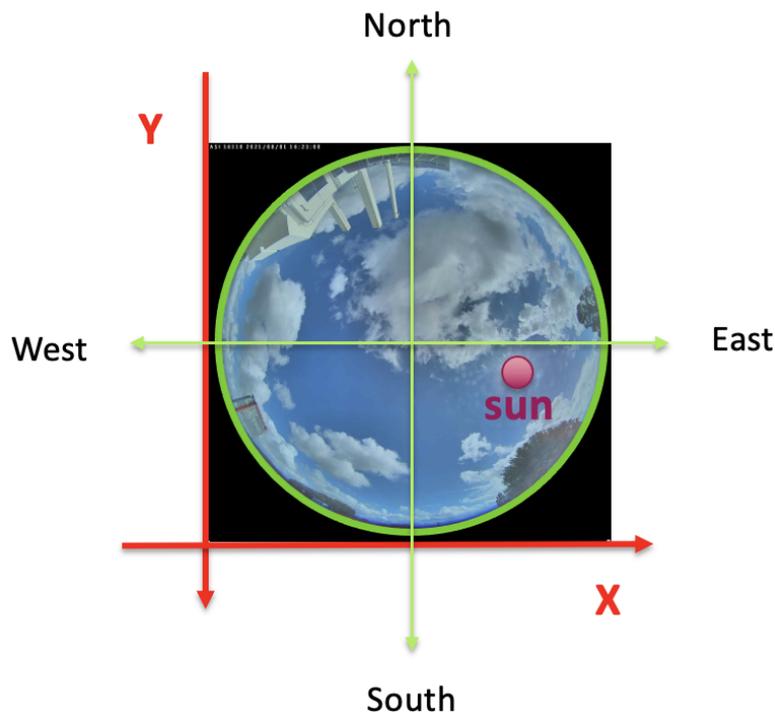


Figure 3.16: The visual method of how to find the X and Y coordinate of the sun in an All Sky Image.

When the location of the sun is found, one can start to create the final clear sky background library. This library consist of all the ASI's when there would be no clouds and for all the sun positions in the image throughout the day. When the library is created, the difference between the clear sky background and the real ASI at the same time can be compared. With the differences in the images the cloud cover can get calculated. [Chauvin et al. \[2015\]](#) worked with the clear-sky library and calculated the CC by comparing the clear-sky image from the library with the real ASI at the same moment. Subtracting the differences between the clear-sky image from the real ASI can detect the clouds in the image. This method performs well when there are less clouds in the sky, but the algorithm's success rate drops to less than 10% when there are more than 40% of clouds in the sky vault compared to the real situation. [Yabuki et al. \[2014\]](#) did a comparable study, but just looked at the R/B differences between the clear sky background and the ASI. On top of that, there is also an extra cloud classification applied. This indicated that the clouds can get

classified into three kind of clouds based on the R/B differences between the clear sky background and the ASI. The bigger the difference, the thicker the cloud. The advantage of this technique is that it is better to detect clouds in situations when there are not much or thick clouds. Also, it is now possible to make differences in the kind of cloud thickness in the air. The disadvantages is that it works less well when there are much clouds in the ASI and that the problem of the light at the beginning and end of the day is just like with thresholding still a problem. For that reason, this method will not be applied in this research.

3.6 MACHINE LEARNING

Artificial Intelligence (AI) is a technology that gives robots and systems the ability to mimic the problem-solving and decision-making capabilities of the human mind IBM [2020]. Machine Learning (ML) is a subset of AI, where in ML computers can learn from data without being explicitly programmed. Making a distinction between traditional programming and ML can be helpful. In conventional programming, the rules that specify how the programme should operate on a computer or other machine are written explicitly by the programmer. However, a ML model picks up the rules from the data. These guidelines do not require specific writing. The ML model's necessary code still needs to be written, though Pramoditha [2021]. DL is a subset of ML in which multilayered neural networks learn from data. So, an example of something that is ML but not DL is an model that learn from data without the use of an neural networks. This can be for example be a classification method like the Support Vector Machine which will get explained more broadly later on. The total overview of the relationships between these three terms are shown in Figure 3.17.

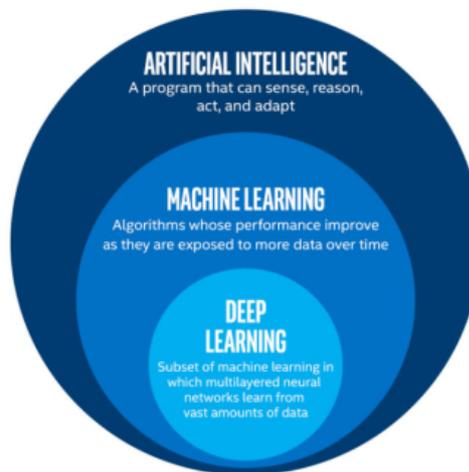


Figure 3.17: Global overview of the difference between Artificial Intelligence (AI), Machine Learning (ML) and Deep learning (DL). Jawahar and Anoop [2022]

The function of ML can be divided into three groups. It may be **descriptive**, which generally is used to understand what the system is and what it does, to find the structure and relationships between the data. The system may be **predictive**, in which case it will make forecasts of the outcome of the data based on the historical data. This can either be used for making a classification or regression model. The last option is **prescriptive**, in which case the system will utilise the information to recommend possible courses of action Malone et al. [2020].

The subcategories of ML can also be divided into three groups. The first category is **supervised** ML. In these models, labeled data sets are used to train ML models,

allowing them to make a correlation between the inputs and the labels in order to become more accurate over time. As an illustration, an algorithm might be trained with images of dogs and other objects that have all been identified by humans, and the computer would then learn how to recognise images of dogs on its own. The other other category is **unsupervised** ML, a programme scans unlabeled data for patterns. Unsupervised machine learning has the ability to identify patterns or trends that individuals are not consciously seeking. For instance, by examining online sales data, an unsupervised ML software could determine the various customer groups making purchases. The last category is **reinforcement** ML. By setting up a reward system, reinforcement ML teaches computers through trial and error to choose the best action. By letting the machine know when it made the appropriate choices, reinforcement learning can train models to play games or train autonomous vehicles to drive. Over time, the machine will understand what actions to take [Malone et al. \[2020\]](#).

Some of these ML methods are also used in ASI classification or irradiance prediction. One of these methods for making classifications is called the Support Vector Machine (*SVM*). The SVM's primary goal is to draw a optimal line (called a hyperplane) that divides the data into classes. The SVM looks for lines between the classification from which the margin line is maximum, ensuring that the classification is maximally isolated from one another. This method is visually shown in [Figure 3.18](#).

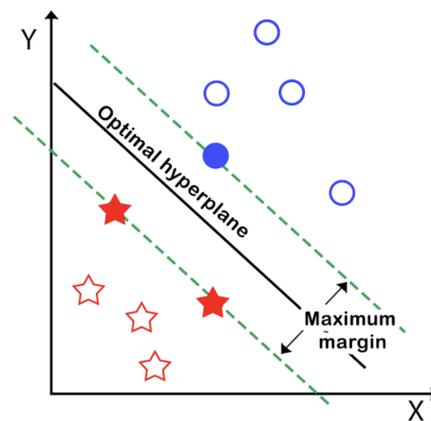


Figure 3.18: Visually explanation of the Support Vector Machine method, image from [Dabakoglu \[2018\]](#)

The SVM is a supervised ML method, so for this method it is important that the used data is already labeled. This would mean that when SVM will be used for ASI's, then these images first needs to get classified into groups. This same method can be used for making prediction instead of classifications, named the Support Vector Regression (*SVR*) method. The SVR seeks to match the best line within a threshold value, in contrast to other Regression models that aim to reduce the error between the real and predicted value. The flexibility of SVR allows us to decide how much error in our model is acceptable, and it will locate a suitable line (or hyperplane in higher dimensions) to fit the data. The hyperplane in [Figure 3.19](#) is illustrated as $w_i x_i$. The boundary lines (or margin lines) are the decision boundaries of the hyperplane. The boundary lines can be adjusted to give our model the required level of accuracy [Sharp \[2020\]](#). The distance between the boundary line and the hyperplane is the threshold value (ϵ). The hyperplane is the actual prediction line, so forecasts continuous output [Raj \[2020\]](#). When a data point falls outside the boundary lines, the deviation from the boundary lines gets denoted as ζ . These deviation will be taken into consideration for the final objective function to create the best fitted line in the end. An example of a linear SVR is shown in [Figure 3.19](#).

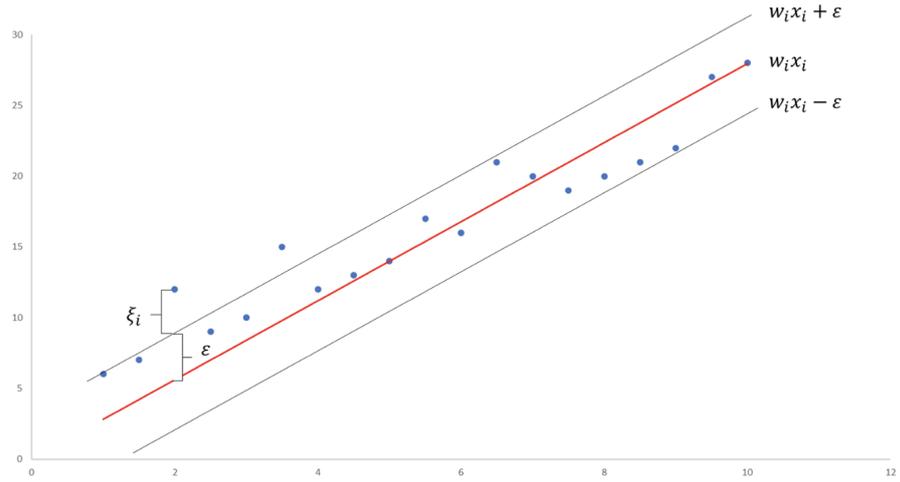


Figure 3.19: Example of the SVR, to show how it works and visualize the used terms. Image taken from Sharp [2020]

A similar kind of method compared to the SVM is called the k-Nearest Neighbor (kNN), which calculates the likelihood that a data point will belong to one group or another based on which group the data points closest to it do Christopher [2021]. When the images are already classified, is it possible to use the SVM and kNN methods to classify new images in the already existing groups. This was done by Li et al. [2021], where they selected 5863 images for training the classifier into four groups and used 1465 images for testing the effectiveness of the methods. The accuracy for both these methods is above the 95% overall. It is however more common that ASI are not labeled, so unsupervised methods can be used better. In that case it is better to use the **k-means segmentation** method.

k-means segmentation

K-means segmentation is relatively simple unsupervised ML and image segmentation method. Image segmentation refers to the splitting of a digital image into a number of separate areas, each having a set of superpixels (groups of pixels) and other related pixels Chauhan [2019]. The K-means segmentation method is an unsupervised learning method, which is an algorithm that learns from unlabeled data by identifying patterns among examples and classifying them appropriately because it lacks prior labelled material to draw on.

The K-means algorithm is a widely used image processing technique for ASI to separate sections of clouds or clear sky in a given image based on their similarity in colour Blazek and Pata [2015]. The method that is used for this is algorithm is shown in Equation 3.13.

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2 \quad (3.13)$$

In the Equation J refers to the objective function, k to the number of clusters, n to the number of used cases, x_i the specific case (or pixel) and finally the centroid of cluster c_j . The $\left\| x_i^{(j)} - c_j \right\|$ is the distance function, so the distance between one specific pixel and the centroid of one cluster. In the case of an image, the Euclidean distance formula is employed to determine the separation between two points on a 2D plane, which can be seen in Equation 3.14.

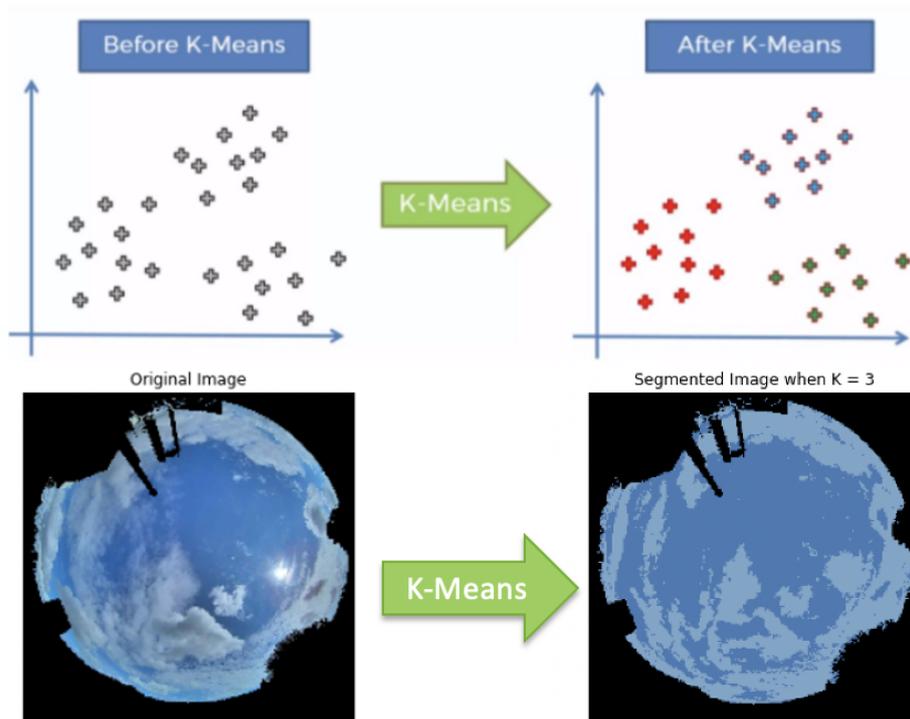


Figure 3.20: An example of how the k-means segmentation method works. It is an unsupervised learning method, so non-labeled data comes in and the method labels the data. Above images from Chhabria [2021]. The lower images are example of the applied k-means clustering method on an ASI.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.14)$$

After calculating the distance, the centroid of all the objects in the clusters needs to get calculated. An example of the k-means segmentation method is shown in Figure 3.20. In this example the ASI with $K=3$ is projected, which indicates that the image gets classified into three groups (so K is the amount of clusters). Apart from the k-means clustering method there is also the k-means++ approach. The only difference is that the centroids are placed in a more intelligent manner with the k-means++ approach Arthur and Vassilvitskii [2007]. The k-means++ removes the drawback of K means which is it is dependent on initialization of centroid. For that reason the k-means++ will get further explained, as all the previous steps of the k-means++ and k-means clustering method are the same. The beginning point of the centroid is at random when the k-means++ method is used. The results of the first centroids that were chosen at random are now used to determine the new centroid in the following scenario. The formula will iterate through the image numerous times in order to find the global lowest sum of squares. It is written as the number of cases ' n ' in the Equation 3.13. The better the centroid is chosen, the higher the compactness of the data points (in case of images, the more pixels with the same values), the lesser the distance from the centre to the data points.

Blazek and Pata [2015] used this method to classify the clouds in the ASI into the ten different cloud classifications as explain in Section 3.3. Other studies mainly used the k-means ++ method in satellite images instead of ASI's Goswami and Bhandari [2011].

These methods are not DL methods, since for DL methods the use of Neural Networks (NN) are used, which will get explained in the next section 3.7.

3.7 DEEP LEARNING

As previously shown in Figure 3.17, DL is subset of ML, while ML is a subset of AI. A model is called a DL model when there are multiple NN layers in the model. Information in a human's brain is transmitted via neurons. They communicate between various brain regions and between the brain and the rest of the nervous system through electrical impulses and chemical signals NIOH [2022]. NN is a mathematical model that mimics the operating principle of neurons in the human brain. In computer models these NN's are called **Artificial Neural Networks (ANN)** and will be explained in the next subsection 3.7.1.

3.7.1 Artificial Neural Network

Every ANN model start with an input layer, then a number of hidden layers, and finally an output layer. Each node in one layer is linked to the other nodes in the layers around. A node is a specific data point and is working as a single neuron. In an ANN network, these nodes are connected via edges. As shown in Figure 3.21, ANN figures commonly shows the node as a small circle, while the edge, which is the connections between those nodes, are usually drawn as arrows. The data goes from input to output, this is also shown with the arrow direction. By adding more hidden layers, which are the layers between the input and output layers, the network is made deeper Dertat [2017]. The hidden layers is the reason for calling this kind of models a DL model.

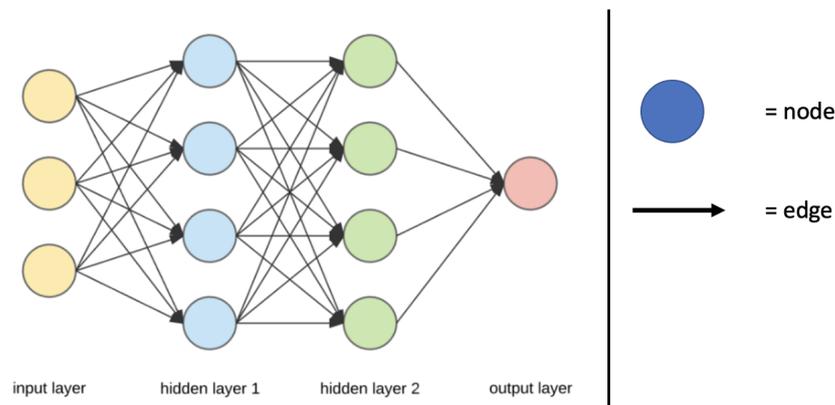


Figure 3.21: An example of how the Artificial Neural Networks (ANN) looks like when it would be drawn. Part of the Figure taken by Dertat [2017].

In this example shown in Figure 3.21, the data is only able to move through the network from the input to the output. This are called feed-forward NN systems. On the other hand, for feedback systems is it possible for data to move both sides inside the network. Insight those systems it will become important to give certain weights to the nodes between the input and the output, as shown in Figure 3.22.

A specific node runs a nonlinear **activation function** on the weighted sum of its inputs. An activation function is a function that is incorporated into an ANN to aid in the network's learning of complicated patterns in the input data. It receives all the output signals from the nodes and transforms it into a form with a upper and lower limit that is used as the input for the node after it Sharma [2017]. The activation

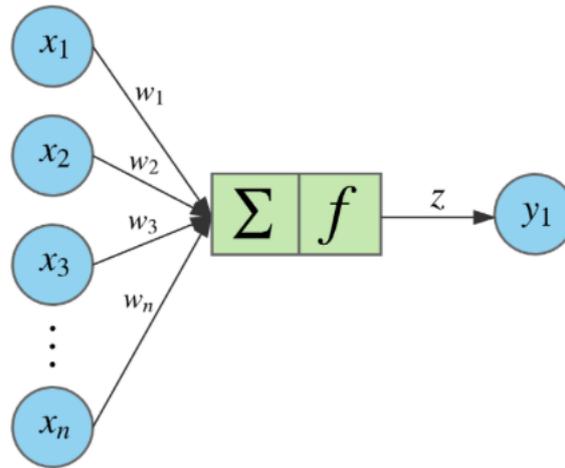


Figure 3.22: An example of how the zoomed in Node looks like when it would be drawn and when an Activation function is added. Figure taken by Dertat [2017].

function determines what signals should be sent to the following neuron. Without activation functions, a NN is essentially just a linear regression model Gupta [2020].

This process is carried out for each node, and the ultimate output—the signal flowing from left to right—is determined. Learning the weights connected to each edge in this deep neural network requires training Dertat [2017]. The equation for a particular node is shown in Equation 3.15. The inputs were combined and processed by a nonlinear activation function.

$$z = f(b + x \cdot w) = f\left(b + \sum_{i=1}^n x_i w_i\right) \quad (3.15)$$

With x the node, w the weighted value, n the number of inputs for the node and b the bias. In Equation 3.15 it is important to acknowledge the added bias to the node. Along with the weighted total of the inputs to the neuron, the bias is an extra parameter in the neural network that is implemented to modify the output. Therefore, bias is a parameter that improves the model in achieving the best fit for the provided data Sharma [2017].

3.7.2 Activation functions

There are different kinds of activation functions used in DL models. Before going into detail about the different kinds of activation functions, it is first important to find out what the desired features (or not) are for an activation function.

The first thing that is desired is that the activation function is **zero-centered**. This is because then the gradients do not shift to a particular (positive or negative) direction. Also, when the activation is zero-centered it would indicate that the mean of the activation is also around zero which creates normalized data that creates faster convergence Jain [2019]. Apart from that, since the activation function is applied millions of times throughout the model, it is desired that it is **computational inexpensive**. In addition, since gradient descent is a method used to train neural networks, the model's layers must be **differentiable**. At last, the feature which is not wanted is the **Vanishing Gradient problem**. The problem that occurs when the weights between the layers become really small and are not able to change the values in the upcoming layers. This influences the training of the NN. When the models wants to decrease the loss in the model, it adjusts the weights in the layers to

improve the outcome. When the weights can barely change the values in the layers, it will become more difficult to optimize the output values. Since the loss function is first calculated and then the weights get adjusted from the output until the input (so backwards going through the model), it is called backwards propagation. Since the layers gets multiplied with each other, the Vanishing Gradient problem is especially a problem for the beginning layers. On the other hand, the same problem can occur the other way around, this is called **Exploding Gradient problem**. The earlier in the layers that one wants to change the weights, the bigger the values can get multiplied with each other and creates too much change in the values. This also results in non optimal results.

The first activation function that will be discussed is the **Sigmoid function**, which is of the most popular non-linear activation functions. The Sigmoid function changes values between 0 and 1, as shown in Figure 3.23(a). The problem with this method is that it results in vanishing gradient problems, since it saturates at zero for large negative values and at one for large positive values. Furthermore, it is not zero-centered which makes it computationally demanding. This technique is typically applied to binary classification issues. This is because the Sigmoid function returns values in the range of 0 and 1, which can be interpreted as probability that a data point belongs to a specific class.

The second one is the **Tanh function**. The tanh function and the sigmoid function are quite similar, the only difference is the symmetry around the origin, which solves the problem of not being zero-centred (as shown in Figure 3.23(b)), but does not solve the vanishing gradient problem.

When solving multiclass classification, the **Softplus function** can be employed, that is why the Softmax function is frequently referred to as a concatenation of different Sigmoids. The chance that a data point belongs to each distinct class is returned by this function. The output layer of a network created to solve a multiclass problem would contain the same number of neurons as classes in the goal [Gupta \[2020\]](#).

Finally, the last activation function that will be discussed is the **Rectified Linear Unit (ReLU) function**. The ReLU function's primary advantage over other activation functions is that it does not simultaneously trigger all of the neurons, which makes it far more computationally efficient than the Sigmoid and Tanh functions [Gupta \[2020\]](#). As shown in Figure 3.23(c), it only triggers its positive values. Also, since the partial derivative is either 0 or 1 it will not cause the Vanishing Gradient problem. The only problem is that since it is not zero centred but the activation function will not learn from the negative input, so the change will only be in the positive direction. For that reason the only used activation function in this research is the ReLU activation function.

3.7.3 Deep learning layers

Up until now, it was common that neurons of the layer are connected to every neuron of its preceding layer. This is called a **dense layer**. However, when images want to be used in the DL model, the dense layers are not sufficient to use. A more common method to extract sufficient data from images are with **convolutional layers**.

Convolutional layers

Convolutional layers detect the important features and patterns due to certain filters from which the model can learn. The name for these kind of models are **CNN**. The idea of how the convolutional layer works is shown in Figure 3.24. The input data that goes into the model is an RGB image (so 3 dimensions) with the size of 6 by 6. The RGB image gets multiplied with the filters and from this the feature maps gets

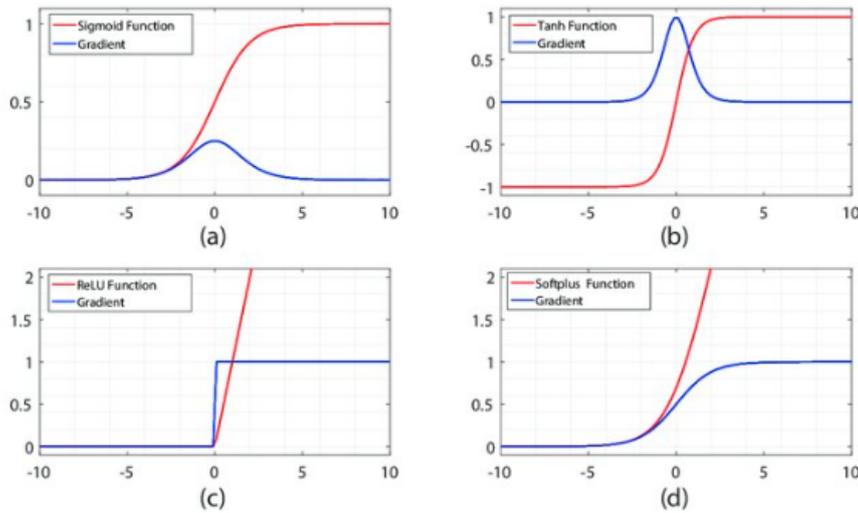


Figure 3.23: Activation functions in comparison. Red curves stand for, respectively, (a) Sigmoid, (b) hyperbolic tangent, (c) ReLU, and (d) Softplus functions. Their first derivative is plotted in blue. Figure taken from [Roffo \[2017\]](#).

created. To be more precise, the filter is consistently applied from top to bottom, left to right, to each overlapping portion or filter-sized patch of the incoming data. If the filter is intended to find a certain kind of feature in the input, then applying it consistently throughout the entire input image gives the filter the chance to find that feature wherever in the image [Brownlee \[2020\]](#). Finally, with these filters the model can find certain patterns and similarities in images. On these feature maps an activation filter can be applied, the use of an activation function is previously explained and in the example of Figure 3.24 the ReLU function is used.

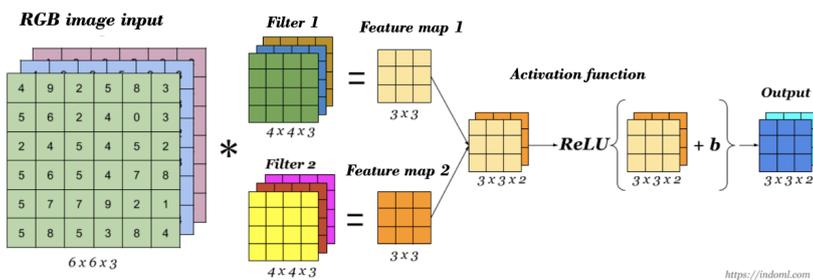


Figure 3.24: An Example of how the convolutional layers works for an RGB image with 6 by 6 pixel size that goes through six 2D kernel (filters), from these multiplication two feature maps are created. Then feature map values can be passed through the activation function, which in this example is the ReLU function with an bias (b). From this an 3 by 3 by 2 output is given. Image from [Ng \[2018\]](#)

In order to improve the final accuracy of the model, it can be expected that the RGB input images are usually of bigger size. As a results, it can be expected that the amount of filters (and size), amount of feature maps (and size) and the amount of output blocks also increases. In order to increase the training speed and make the model more efficient **pooling & flatten layers** can be used [Jeong \[2019\]](#).

Pooling & Flatten layers

First, the **Pooling layer** will be discussed. The pooling layer is a method that decrease the size of a feature map. A pooling layer often comes after a convolutional layer in a CNN. Typically, a pooling layer is added to speed up computation and strengthen some of the identified features. There are various forms of pooling. The two most popular pooling techniques for CNN's are max pooling and average pooling. In max pooling, the highest value of the chosen window is used to build a reduced map from each patch of a feature map. In average pooling, the average value of the chosen window is used from each patch of a feature map [Pokhrel 2019]. These two different pooling methods are shown in Figure 3.25.

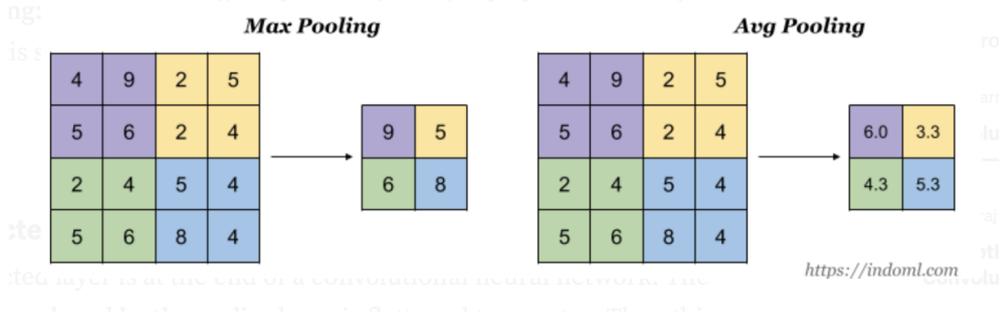


Figure 3.25: An example of the two different pooling layers techniques, maximum pooling (left part) and average pooling (right part). Image from Ng [2018]

After the pooling layer, it is common to use a **flatten layer**. The flatten layer makes the data inside the model flattened, so the output is 1D. This is important since the final output of the model should also be in 1D. Figure 3.26 shows how the Pooled 2D layer then gets flattened. With this flattened layer the dense layers can be used to come to the final output value which one wants to receive from the model.

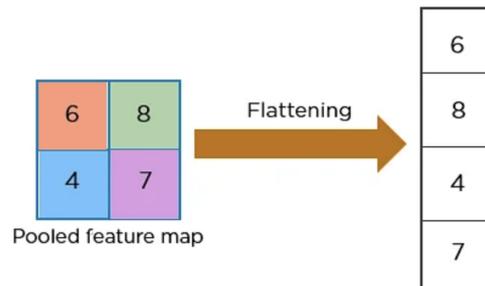


Figure 3.26: An example of the pooled that gets flattened to a flatten layer. This is needed to find the final 1D output value from the model. Image from Biswal [2022]

3.7.4 Deep learning model training

After the model is made, it is finally possible to start training it. The goal of training the model is to get the most accurate outcome of the model. For a prediction model, the difference between the actual values and the predicted values should be as small as possible. This could be checked with the Mean Average Error (MAE), Root Mean Squared Error (RMSE) or the 95% quantile. How these performances are specifically evaluated will get further explained in subsection 4.3.2. In this subsection the emphasis lays on how the model gets trained in order to improve the final outcome. This is done in five steps.

Step 1 is to split the data into a **training & test dataset**. The training data is the data that is used to train the model. The test data is to check how the model performs on data is not used for the training. With the test dataset the model outcome can get validated. How the data gets split between training & test data is for every model different. According to empirical research, using 70–80% of the data for training and the remaining 20–30% for testing yields the most sufficient outcomes. [Gholamy et al. \[2018\]](#) Within this 20–30% test data, sometimes an extra split is made for testing the outcome and after that validate the outcome.

Step 2 is the **Forward propagation**. At first, a linear combination of inputs and weights per step are made. Secondly, a non-linear function is applied in the form of an activation function, which introduced non-linearity into the NN. This can go through various deep layers before finally get the output.

Step 3 is calculating the **loss function**. The loss can be determined by calculating the difference between the output of the model and the actual results from the test data. Different loss function's can be used for this dependent on which kind of model is used. In this research the Mean Squared Error (**MSE**) is used as loss function, since it penalizes larger errors more than smaller errors.

Step 4 is the minimization of the loss function. This can be done with the **back-propagation technique**, the chain rule is applied to compute gradients all the way to the inputs of the network and then update the weights [Brownlee \[2021\]](#). The gradients can then be used by an optimization algorithm to update the model weights. A method to do this is **Stochastic Gradient Descent (SGD)**, which is an optimization method for reducing the models predictive loss with respect to the training dataset. The problem of SGD is that it has a constant learning rate, which is the tuning parameter that that specifies the step size at each iteration. For that reason, the Adaptive moment estimation (**Adam**) optimization of the loss function method will be used. Adam computes adaptive learning rates, is computational efficient, requires little memory and compared to other optimizer methods almost always performs best [Kingma and Ba \[2014\]](#).

Step 5 is to repeat this process until the loss function is minimized. The amount of times this process is done is called **epochs**. It is important to find the optimal numbers of epochs in order to make the most accurate predictions. To less epochs would underfit the dataset, while to many epochs would overfit the dataset. The right amount of epochs is found when the training set outcome of the model is in the same outcome range as the test outcome. The right amount of epochs can be found when controlling the validation set loss with the outcome of the training loss. When the amount of epochs have been reached that the validation and training set reaches the same outcome, the `tf.keras.callbacks.EarlyStopping()` function in Python can be used to stop the model from running more epochs, in order to prevent overfitting.

4 | METHODOLOGY

4.1 PRE-PROCESSING DATA

First, all the data was obtained in Delft, a city in the Netherlands. More specifically, the equipment's that were needed to do the measurements were all located at the rooftop of 'Combined heat power plant' (Warmtekrachtcentrale) at the Technical University of Delft. The latitude and longitude of this location are (52.011, 4.357). In Figure 4.1, one can see the used equipment of the brand Schreder. The first one is the ASI-16 which makes the All-Sky Images (ASI) every minute of the day. A single image is saved as a 3D matrix with a width of 1536 pixels, a height of 1536 pixels, and a depth of 3 colour channels. The second equipment is the pyranometer, which measured the GHI for every 15 seconds at this location. The last equipment is the temperature and humidity sensors, which data will not be used. The irradiance incoming time step is not the same as the incoming ASI's. Also, the ASI's stops with making images when the sun goes down, while the irradiance appears every 15 seconds. In order to combine the information correctly, the irradiance input was converted into 1 minute data for all the moments that there were ASI's being made. There are two ways of doing this. Either the average over the whole minute is taken or a single measurement at the top of the minute. In this research the average over the 4 time steps within the 1 minute is used.

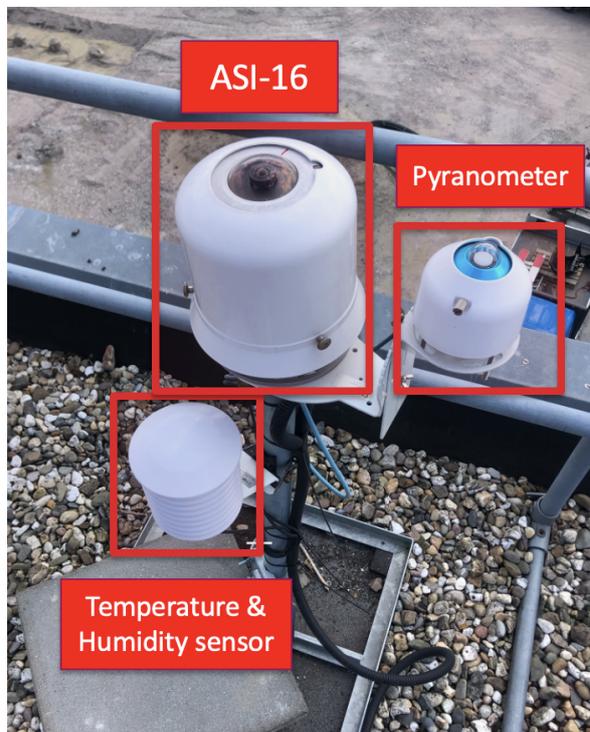


Figure 4.1: All the equipment needed for making the all sky images and the extra information, located at the TU Delft, the Netherlands.

Apart from the measured GHI, also the clear sky GHI was calculated at the location of the TU Delft. As previously explained in subsection 3.2.3, the Simplified Solis model was one of the best performing CSI models, so is used in this research. The 'get clearsky()' method found in the pvlib Python library was used to retrieve the 1-minute clear sky GHI Holmgren et al. [2018]. With the CSI and the measured GHI, the K_c at that same moment can also be calculated, which will be used to make the classification between the different weather conditions.

The last step of pre-processing the data is to make all the data correspond with the Coordinated Universal Time (UTC). This is done to prevent making errors between the winter and summer time. Also, it is easier to implement other dataset in the same model. The final amount of data time steps for the locations Delft is 313729. The data starts at 26th of March in 2021 and stops at the 8th of October in 2022. In between some data has been lost (24th of February 2022 until 9th of July) since the ASI-16 was not working properly. The overview of the amount of data that is used per month this is shown in Figure 4.2(a). The longest day of the year in Netherlands, Delft is 21th of June and the shortest 21th of December. Since only the data where potential irradiance could be measured is taken into consideration, the distribution for number of data points per month is clearly projected as the amount of sun minutes per month.

To validate the model, extra input information is used of the MSc thesis of Doodkorte [2021]. This is the irradiance and ASI for all 1 minute data from California, the Folsom area Pedro et al. [2019]. The clear sky irradiance is calculated with the same method as for the location Delft. The extracted data is from all the data of 2016 and consist totally of 260610 time steps. The overview of the amount of data that is used per month this is shown in Figure 4.2(b). The longest day of the year in California, Folsom is 21th of June and the shortest 21th of December (same as Delft). Also here, only the data where potential irradiance could be measured is taken into consideration. More detailed information about this dataset can be found in the MSc thesis of Doodkorte [2021].

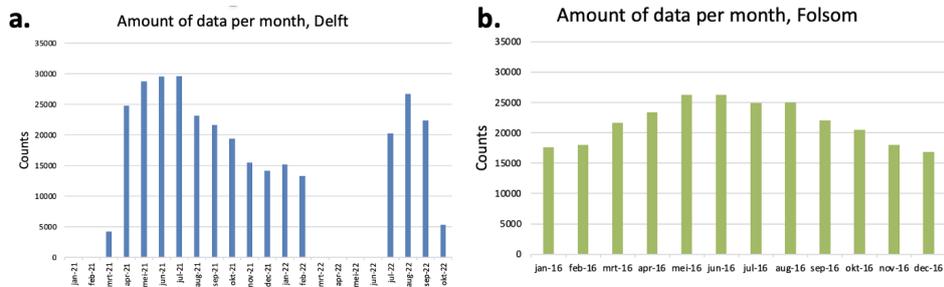


Figure 4.2: (a) Overview of the amount of data per month that is used for the location Delft. (b) Overview of the amount of data per month that is used for the location Folsom.

4.2 CLASSIFICATION PROCESS

4.2.1 Roadmap through the classification

For the final classification process it is first important that the CC will be found for all ASI's. The CC will get calculated for all the ASI's that are used in the dataset. The final algorithm that is used to calculate the CC is shown in Figure 4.5 and will be explained step by step.

For the original ASI, it is important that the location of the sun in the image will be found. As previously explained in subsection 3.5.2, the position of the sun at a specific location can be translated into X & Y coordinates with Equation 3.12. Before this Equation can be used, it is first important that the ASI first gets flipped and rotated with the North as the highest middle point of the image. For the data at the location Delft, the images needed to get flipped first and after that rotated with 24 Degrees. Apart from that, the camera is not exactly perpendicular directed to the sky. The camera is 2 degrees tilted towards the East. This means that the sun position in the sun is slightly different than the exact location according to Equation 3.12. This means that the sun path in the ASI's are different then how they would go when the ASI was pointed perpendicular to the sky. An example of the sun path actual sun path versus the expected sun path of April the 12th of 2021 is shown in Figure 4.3. For that reason, the center value of the sun path is not exactly the middle of the image, but slightly shifted in the West direction.

When the position of the sun in the image is found, the next step is to make a cropped image of only the location of the sun can be made. According to [Alessandro Niccolai \[2020\]](#), when the camera is projected correctly the average pixel error for the exact sun location is 7.87 pixels with a standard deviation of 5.80 pixels. This was based on 20,394 ASI's. This indicates a 99% accuracy in the range of 30 pixels (based on the confidence interval of 99%). On the other hand, in the paper of [Zuo et al. \[2022\]](#), a radius of 80 pixels around the center of sun was enough to detect sun locations of the 150 ASI's that where used. For that reason, the cropped image is made with a radius of the 80 pixels around the location of the sun.

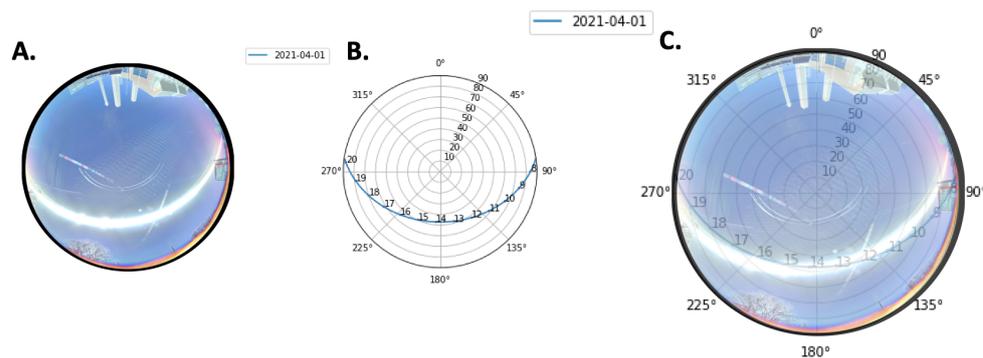


Figure 4.3: The real sunpath of the ASI's for the day 1st April in 2021. In figure 4.2(A), the sunpath of one day is shown for the images of that day. Figure 4.2(b) shows how the sun path would actually look like. In Figure 4.2(c) the both images are merged together so one could see the difference between the real sun path and the theoretical.

When the cropped image is made, the next step is to detect whether the sun is blocked by clouds or not. This is done with a fixed threshold value over the cropped ASI. The fixed threshold value will be at 210. The threshold value of 210 is based on the research of [Zuo et al. \[2022\]](#), where they looked for the minimum threshold value which still gets classified as a sun for 200 cropped ASI's. So, values above 210 will be detected as sun and values below 210 will be detected as not sun. Four examples of how the sun gets detected in the cropped images are shown in Figure 4.4. From the thresholded images one can now detect whether the sun is shining or not. The size of the sun changes throughout the day in an ASI, whereas the sun appears smaller in the beginning and end of the day. In order to always detect the sun throughout the day, the smallest sun will be considered as minimum range to get classified as sun, which is 5% of the cropped image (pixel size of around 18x18).

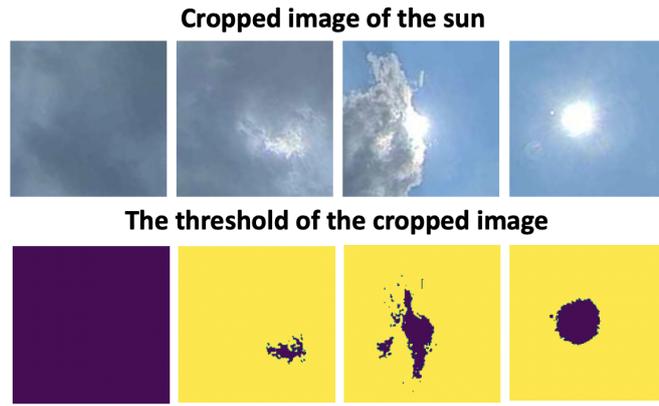


Figure 4.4: Four examples of the cropped image of where the sun in the image is located and the thresholded cropped image, to see how the sun gets detected.

For the situation that the sun is blocked by clouds, so there is no sun visible in the cropped image, one can use another threshold method for the whole ASI. With this threshold method the CC ratio can be calculated for the situation that the sun is blocked with clouds. This method used the red-to-blue ratio of 0.6 as the fixed threshold value to find the CC in ASI, which is based on the findings of Long et al. [2006]. More specifically, pixels are categorized as cloudy if their red-to-blue ratio (R/B) > 0.6 and as cloud-free if it is lower.

On the other hand, when the sun is not blocked with clouds, it is important to find out whether there are still cloud pixels detected in the ASI or not (see Figure 4.5). In order to answer this question it is important to look at the mean pixel intensity of the images with the area around the sun excluded. This means that the mean pixel intensity is calculated for images where the cropped sun location part is not taken into consideration. The mean pixel intensity can be calculated with Equation 4.1.

$$\text{Pixel intensity}_{mean} = \frac{\left(\text{Value}_{\text{pixel}} * N_{\text{pixels}} \right)}{\left(255 * N_{\text{pixels}} \right)} \quad (4.1)$$

Since the RGB values range between $[0 : 255]$, the mean pixel intensity gets divided by 255 to only get values between $[0 : 1]$. For this method, the mean pixel intensity for 200 ASI's were used. These images were already classified as sun not blocked by clouds, so either partly cloudy or clear sky images. For these images the mean pixel intensity is calculated and the values are shown in Table 4.1.

Statistical features	Clear sky	Partly cloudy
Average	0.345	0.420
Max	0.397	0.483
Min	0.297	0.354

Table 4.1: The image's pixel intensity values, excluding the region surrounding the sun among the 200 pictures.

When the mean pixel intensity is below 0.354, this would indicate that the ASI is classified as an clear sky image. Since that is the lowest value of the partly cloudy sky conditions. This would give an CC % of zero, since it is assumed that below the mean pixel intensity of 0.354, the ASI can be classified as a clear sky image. There are with this method always ASI's which are clear sky but does not get detected. For these images the CC can still be calculated via the last step.

The last step is when the sun is not blocked with clouds but there are still cloud pixels in the ASI present. For this situation the SPS method of Liu et al. [2015] is used, as previously explain in subsection 3.5.1. The final outcome of how the CC is calculated for all the ASI's is shown below in Figure 4.5.

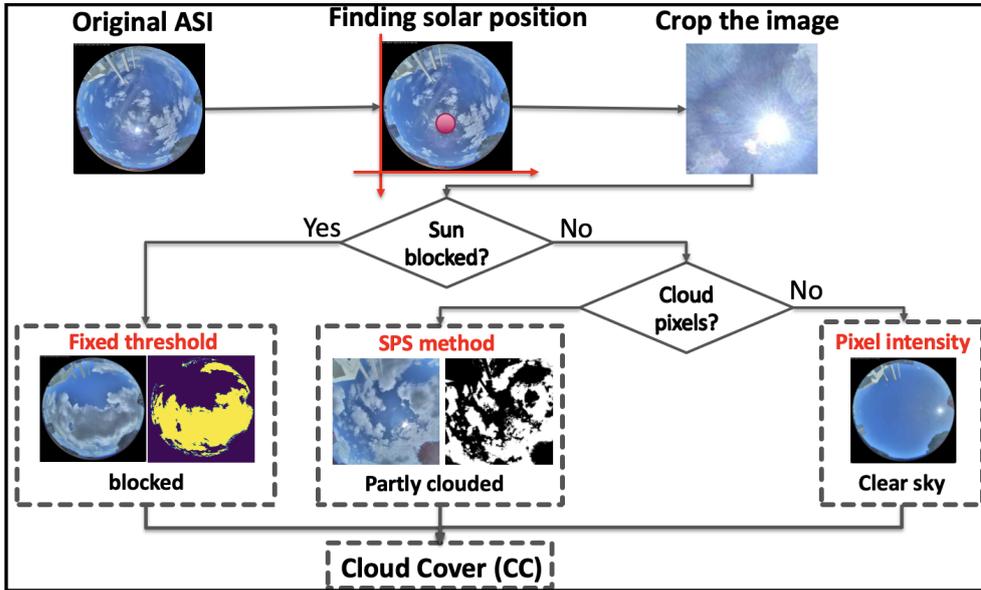


Figure 4.5: The overview of how the cloud cover is calculated for all the ASI's. First one needs to find out whether the sun is blocked or not, then is there are cloud pixels in the image or not. Finally, the CC can be found for all the images. In this example the most two left images will be considered as blocked sun and the most right two images as sun not blocked with clouds.

The cloud cover algorithm is partly based on the algorithm of Zuo et al. [2022]. The difference is that in this research when the sun is not blocked and there are cloud pixels the SPS method of Liu et al. [2015] instead of the global Otsu threshold.

Once the CC is calculated, it will become important to find the K_c at that same time moment. The K_c can be found by looking at the CSI and the measured irradiance at a given time moment (as explained in subsection 3.2.3. The ASI's will get classified into five categories and are based on the CC and K_c at the specific time moment. The overview of the classification is shown in Table 4.2.

Classification	Cloud Cover (CC)	Clear Sky Index (K_c)
A Clear sky	$0 < CC < 0.25$	$K_c > 0.75$
B Partly clouded high irradiance	$0.25 < CC < 0.75$	$K_c > 0.75$
C Partly clouded medium irradiance	$0.25 < CC < 0.75$	$0.4 < K_c < 0.75$
D Partly clouded low irradiance	$0.25 < CC < 0.75$	$K_c < 0.4$
E Overcast sky	$0.75 < CC < 1$	$K_c < 0.4$

Table 4.2: The overview of the five ASI classification based on the Cloud Cover and Clear Sky Index.

This classification method can finally be used to categories the data for final irradiance predictions. First the performance evaluation of this classification method will get explained in the next subsection 4.2.2.

4.2.2 Performance evaluation

The performance of the classification method will be evaluated by looking into the three different variables. The first variable is recall, which measures the quantity of

the classification. The question that can be answered with the recall variable in this case is: 'From all the images that could get classified, how many of those images will get classified correctly?' The answer on this question is a quantity, since with this answer one will know if a image is classified at all. The formula for the Recall is shown Equation 4.2.

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

For this classification variable the True Positive (TP) refers to the amount of images that are classified correctly, whereas the False Negative (FN) refers to the amount of images that are not classified but should be classified in a class.

The second variable that is the precision. The precision variable measures the quality of the classification. The question that can be answered with the precision variable in this case is: 'From the images that are classified, how many of those images are classified correctly?' The answer will give inside in the quality of the classification. The formula for the precision is shown in Equation 4.3.

$$Precision = \frac{TP}{TP + FP} \quad (4.3)$$

For this classification variable, the TP refers to the amount of images that are classified in the correct classification. The FP refers to the images that are classified, but in the wrong classification.

The last variable is the F-score. For a classification model is it important to measure both the quality and quantity, for that reason the F-score is introduced. The Equation 4.4 shown the formula that can be used for calculating the F-score.

$$F_{score} = \frac{1}{\frac{1}{2} \left(\frac{1}{Precision} + \frac{1}{Recall} \right)} = \frac{2 \times \text{precision} \times \text{recall}}{(\text{precision} + \text{recall})} \quad (4.4)$$

The outcome of the Recall and Precision can be used for the outcome of the F-score. The F-score is the harmonic mean of those two variables [Harikrishnan \[2019\]](#). With the F-score, a final outcome can be given on how well the classification model has performed. All the three variables can have a value between zero (worst classifier) and one (best classifier).

Furthermore, in order to look at the differences of the classification for multiple locations, an extra variable will be explained which is the Variability Index (VI). The "length" of the measured irradiance plotted against time divided by the "length" of the clear sky irradiance plotted against time can be thought of as the variability index [Stein et al. \[2012\]](#). The VI is the difference of the measured GHI between a given time range divided by the difference of the CSI for that same time range, as shown in Equation 4.5.

$$VI_t = \frac{|(GHI_t^{\text{Measured}} - GHI_{(t+1)}^{\text{Measured}})|}{|(GHI_{(t+1)}^{\text{CSI}} - GHI_{(t+1)}^{\text{CSI}})|} \quad (4.5)$$

In this research, the VI is chosen to be done on a time range of 1 minute. Since the data is processed by the minute, the VI for every minute can be made. With the VI one can see what the minutely difference is between the time steps. With this variable one can compare if the sky conditions for different locations and classification are changing fast or not.

4.3 PREDICTION MODEL ARCHITECTURES

4.3.1 Roadmap through the prediction models

For the final prediction model, four different methods are used. These models will be explained one by one. In order to evaluate the prediction models by one another it is first important to find a baseline model. As explained in subsection 3.4.1, the guidelines of a baseline model is that it should be simple to implement, fast and repeatable. A commonly used baseline model for irradiance predictions is the Persistence Model (PM).

Baseline model

The PM uses the most recent time step to make predictions over the future time steps. So, for the irradiance predictions it uses the most recent GHI time step to make future GHI predictions, as shown in equation 4.6.

$$GHI(t + \Delta t)_{predicted} = GHI(t = 0)_{measured} \quad (4.6)$$

So, the persistence model uses the GHI input at time t to make irradiance predictions for all the future GHI prediction steps. On top of the PM, there is also an Smart Persistence Model (SPM). The SPM adjusts the PM's output by applying the K_c (see equation 3.3) to future time steps, which can be seen in equation 4.7.

$$GHI(t + \Delta t)_{predicted} = K_c(t = 0) * GHI(t + \Delta t)_{clear} \quad (4.7)$$

For the SPM the clear sky index at time $t = 0$ will be multiplied with the given clear sky irradiance. The SPM and PM are in line with the guidelines that are needed for a baseline model. However, since the sky condition at time t will be taken into account, it is expected that the SPM outperforms the PM. For that reason, the SPM will be used as the final baseline model.

Auto Regressive Integrated Moving Average (ARIMA) model

The third prediction model which will be used is the ARIMA model. The ARIMA model is a common used time series prediction model which is already explained in subsection 3.4.1. For this model it is important to have a set of historical GHI data that can be used to make the 21-min GHI predictions. The historical dataset in order to make this 21-min predictions is chosen to contain 500 historical GHI data points. This is historical data set of 500 is based on the finding of Hans [2019], which stated that little improvement in forecast predictions can be obtained by increasing the dataset with more than one day of data for 15 minute and 1 hour time series predictions. The minutely dataset of 500 corresponds on average to one day of data. When using this 500 historical GHI data points for multiple prediction over time, the First In First Out (FIFO) method will be used. This means that when making a new prediction at a given time t , the $t - 500$ will be removed from the dataset and the 'new' $t - 1$ will be added. This is shown in Figure 4.6.

Apart from the dataset, it is also important to find the right ARIMA(p,d,q) variables for making the predictions. This will be explained in subsection 4.3.2. In this research, the variables are found with the `pmdarima.auto_arima()` function in Python, which automatically discover the optimal order for an ARIMA model.

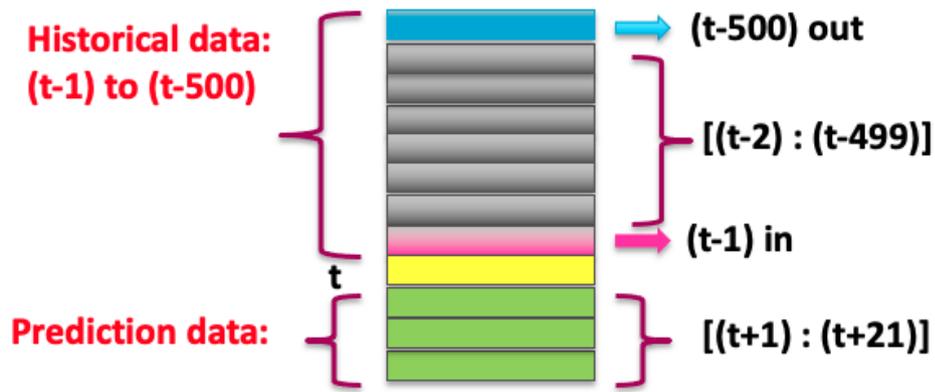


Figure 4.6: Overview of how the ARIMA FIFO system works to make the 21-min GHI predictions. For every 21-min prediction, a 500 historical dataset is used which shifts throughout the dataset via the FIFO method.

Convolutional Neural Networks (CNN) model

The fourth prediction model that will be used is the CNN model. For this model the input of the ASI's are needed to make future predictions. The used CNN model is based on the MSc thesis of Doodkorte [2021]. First the pixels in the ASI gets rescaled, so normally the values of an RGB image will go from $[0:255]$, but in the CNN model the values will vary between $[-1:1]$. Then, convolutional blocks are used to take the ASI images and extract useful information from them. Four CNN blocks make up the entire convolutional network, with the number of filters increasing and the filter size decreasing. This is done to extract more data from the ASI. Including more filters extracts more detailed information from the ASI. The final layer is a flatten layer, which converts the three dimensional layers to one. To extract more complexity from the image, the flattened output from the CNN blocks is then fed into two dense layers. The final overview of the used CNN network is shown in Figure 4.7.

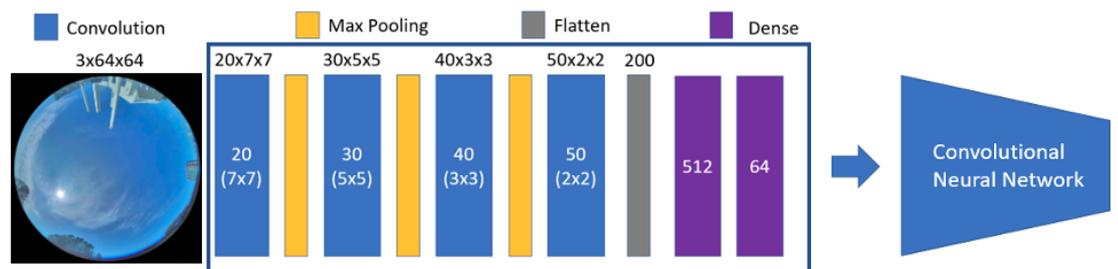


Figure 4.7: This graphic shows the convolutional neural network that was utilised in the deep learning model to extract visual features. Figure made by Doodkorte [2021] and slightly adjusted.

Apart from that, the GHI and CSI are used as input variables and is the auxiliary data of this model. In the MSc thesis of Doodkorte [2021] multiple input variables were used to check the performance outcome of the CNN prediction model. The results showed that the CNN model with the input variables of only the ASI, GHI and CSI resulted in the lowest Mean Average Error (MAE), which is a function that is used for finding the average magnitude of error which will further analysed in subsection 4.3.2. For that reason this research uses the GHI and CSI as auxiliary input data.

So, the measured GHI and CSI comprise the auxiliary data. The auxiliary data first gets normalized, this is done by scaling and shifting all the input data into a distribution centered around 0 with a standard deviation of 1. As shown in Figure 4.8, a dense NN with two layers is employed to accomplish this. The first layer consist of 16 units, while the second layer consist of 8 units.

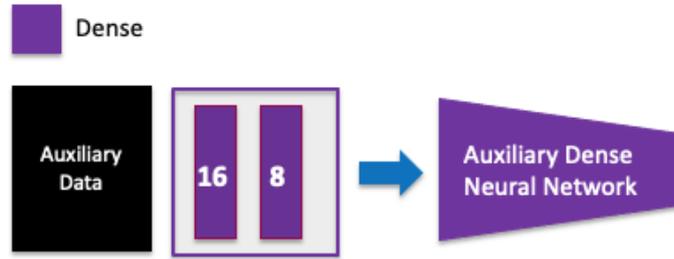


Figure 4.8: This graphic shows the auxiliary dense neural network that was utilised in the deep learning model. Figure made by Doodkorte [2021] and slightly adjusted.

So, the final dense layer consist of 8 units. The auxiliary data with size (1X8) will get concatenated with the CNN input (1x64). This results in an concatenated layer of (1X72). The final forecast is based on the concatenated vector. Then two dense layers will be used, the first dense layer consist of 64 units and the second one consist of 32 units. The vector has 22 distinct dense layers with a single unit each, representing the forecast horizon of 0 to 21 minutes. Total overview of all the layers are displayed in Figure 4.9 and below in chronological order.

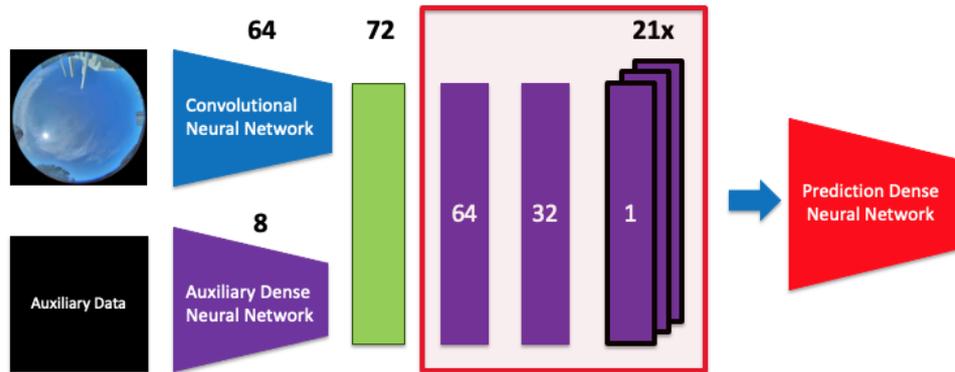


Figure 4.9: This graphic shows the total overview of the deep learning prediction mode that was utilised in the deep learning model. Figure made by Doodkorte [2021] and slightly adjusted.

4.3.2 Performance evaluation

For the performance evaluation of the prediction models, first the loss function of the CNN model must be chosen. The loss function is used to update the internal weights in the model in order to improve the final prediction. The used loss function is the MSE and is shown in Equation 4.8.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (4.8)$$

Where Y_i refers to the measured GHI, \hat{Y}_i to the predicted GHI and n to the number of predictions taken into consideration. The squared difference between the pre-

dicted and measured values causes the MSE loss function to penalise greater errors more severely than tiny errors. Additionally, it has already been established that MSE loss functions outperform other loss functions and display promising results when used with ASI images for short-term GHI predictions [Paletta et al. \[2021a\]](#). The problem with the MSE is the unit, which is W^2/m^4 and not the standard International System of Units (SI). Measurements that are in the correct SI are the MAE and the Root Mean Squared Error (RMSE), with both the unit of W/m^2 . The Equations 4.9 & 4.10 shows how the MAE and RMSE can get calculated.

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (4.9)$$

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{Y}_i - Y_i)^2}{n}} \quad (4.10)$$

The MAE and the RMSE can both be used to diagnose the variation in the errors in a set of forecasts. The RMSE will always be larger or equal to the MAE. The greater difference between them, the greater the variance in the individual errors in the sample. If the RMSE=MAE, then all the errors are of the same magnitude.

In order to better compare the relative difference between predictions for other locations and with other papers, the normalized Mean Average Error (nMAE) & normalized Root Mean Squared Error (nRMSE) are calculated and shown in Equations 4.11 & 4.12.

$$nMAE = \frac{MAE}{GHI_{\text{measured}}^{\text{mean}}} \quad (4.11)$$

$$nRMSE = \frac{RMSE}{GHI_{\text{measured}}^{\text{mean}}} \quad (4.12)$$

The $GHI_{\text{measured}}^{\text{mean}}$ is the mean of all measured GHI in the used datasets. For location Folsom all measured GHI is for the year 2016. However, for location Delft, there is no available consecutive year of data available, for that reason all measured GHI will be used. For the location Delft this is $263.5W/m^2$, whereas for the location Folsom it is $411.4W/m^2$.

Apart from the whole MAE over the 21-minute predictions, it is also interesting to look at the MAE for each time step over the 21-minute prediction. This is done to compare different models how they perform for the minutely prediction comparison. The formula for the MAE per timestep is shown in Equation 4.13.

$$MAE_{\tau} = \frac{1}{n_{\tau}} \sum_{t=1}^n |Y_i - \hat{Y}_i| \quad (4.13)$$

Where τ represent the specific timestep one wants to calculate, so n_{τ} is the amount of predictions for timestep τ .

Another method to control the performance relative to each other is the Forecast Skill (FS). The FS uses the baseline model to compare with the other models used in the research [Paletta et al. \[2021a\]](#). In this research the FS is done based on the

difference in MAE over the whole forecast horizon compared to other models and shown in Equation 4.14.

$$FS = \frac{MAE_{\text{base}} - MAE_{\text{new}}}{MAE_{\text{base}}} \quad (4.14)$$

With this, the other models can get quickly compared with the baseline model based on the positive (or negative) difference with the baseline model.

However, the FS based on averaged errors (in this case on the MAE) and other indicators offer little insight into the model's capacity to forecast infrequent but significant big errors [Paletta et al. \[2021b\]](#). For that reason the 95% quantile error based on the sorted list of MAE is also calculated to assess this component of the projections. For example, when you would have a 95% quantile of 100 W/m^2 based on the MAE, then 95% of the MAE's in the dataset would be 100 W/m^2 or lower.

5 | RESULTS

5.1 CLASSIFICATION RESULTS

The classification final results for both the location Delft and Folsom are depicted in Table 5.1. In this table one can see the total amount of data points that are used for making the classification for both situations. As one can see, there is a clear distribution between the classes itself and between the locations. On top of that, in Figure 5.1 is the classification distributed showed for both locations. Furthermore, it is important to note that an extra class named 'class 2 (partly)' is added. This class is the combined Class B, C & D and will be used later on in the prediction sensitivity analysis in Section 5.2.3.

	Delft	Folsom	Combined
Total	313729	260610	574339
Class A (clear)	82048	171520	253568
Class B (Partly clouded high irradiance)	49720	19253	68973
Class C (Partly clouded medium irradiance)	25096	20116	45212
Class D (Partly clouded low irradiance)	19738	9973	29711
Class E (overcast)	79655	23973	103628
Class 2 (partly)	94554	49342	143896
Classified	82%	94%	87%

Table 5.1: The final outcome of the classification for both location Delft & Folsom.

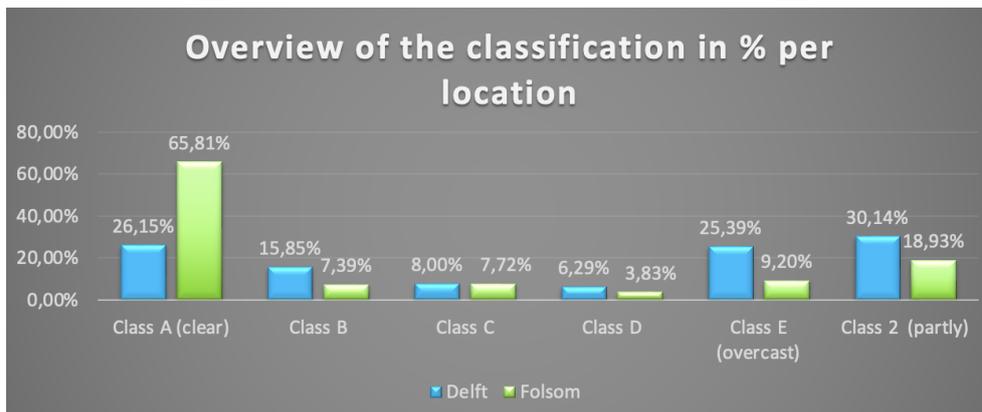


Figure 5.1: Overview of the distributions of the classification over the 5 classes for location Delft & Folsom.

One can see that for the location Folsom, the clear sky situation of Class A is almost $\frac{2}{3}^{th}$ of the time in a year. Whereas for the situation Delft is a little more than $\frac{1}{4}^{th}$ of the time. However, for the situation in Delft, the distribution between clear sky, partly clouded and overcast situations is fairly even. This difference in distribution for both locations were expected, since both distributions are in line with the regional yearly weather conditions KNMI [2021] & Spark [2016].

In order to calculate the Recall & Precision, 250 random ASI's were used and then manually checked with a researcher if the classification was in the correct class (TP). For calculating the Recall one wants to know for how many of the images could get classified are classified correctly. This is the case for 84.0% of the time as shown in the calculation below for location Delft.

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{200}{238} = 84.0\%$$

For precision one wants to look at the images that are already classified, if they are classified correctly. The calculation for location Delft is shown below.

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{200}{212} = 94.3\%$$

Then finally the F-score can be calculated by combining the outcome of the precision & recall. The final calculation for location Delft is shown in the next line and the final classification performance for both locations is shown in Table 5.2.

$$\text{F-score} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})} = \frac{2 * 0.943 * 0.840}{0.943 + 0.840} = 88.9\%$$

	Delft	Folsom
Recall	84.0%	93.3%
Precision	94.3%	89.4%
F-Score	88.9%	91.2%

Table 5.2: The final classification performance based on the recall, precision & F-score for both locations.

In Table 5.2, one can see that the Recall performance of location Folsom is higher whereas the Precision of location Delft is higher. Due to the high number of clear days in the location Folsom, the number of expected misclassifications is also lower. This is clearly shown in the Recall performance. Since the classification method is made for location Delft, it was also expected that the Precision for that location would be higher. However, the performance of the Precision for location Folsom is still convenient, which results in an overall higher F-score compared to location Delft.

In addition, on the calculation of the CC an extra validation is applied. In this analysis, for the 250 ASI's that are used for validation, the difference between CC from the algorithm and the manual observation is shown in Figure 5.2. First, the CC outcome from the algorithm will get converted to oktas as shown in Table 3.1 in Section 3.5. The CC difference is shown in Oktas, which means that when the difference in oktas in Figure 5.2 is zero, the algorithm outcome is in line with the manual observation.

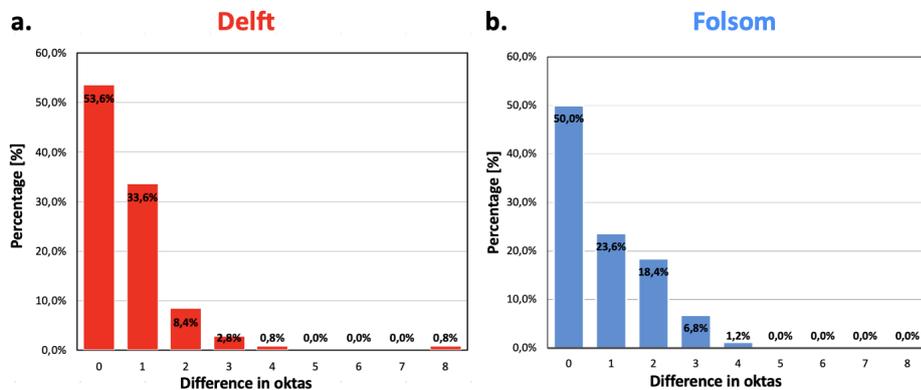


Figure 5.2: Difference distribution (in oktas) of CC algorithm and with the manual observation for (a) location Delft and (b) location Folsom.

In Figure 5.2, one can see that for 95.6% of the times the difference between manual observation and the CC algorithm is within 2 oktas for location Delft and 92% for location Folsom. This results in a higher precision for location Delft compared to Folsom. For location Delft, there are also 2 ASI's which have a difference of 8 oktas. In both these situations the CC algorithm calculated a CC of 0 okta while in the manual observation a sky situation of 8 oktas were obtained. Both ASI's were made early in the morning and are shown in Figure 5.3. These both ASI's should exactly get classified as 'no-class', but in this case got classified as Class A (clear sky). This is because the k_c is higher than 0.75 while the actual cloud cover is 8 okta.

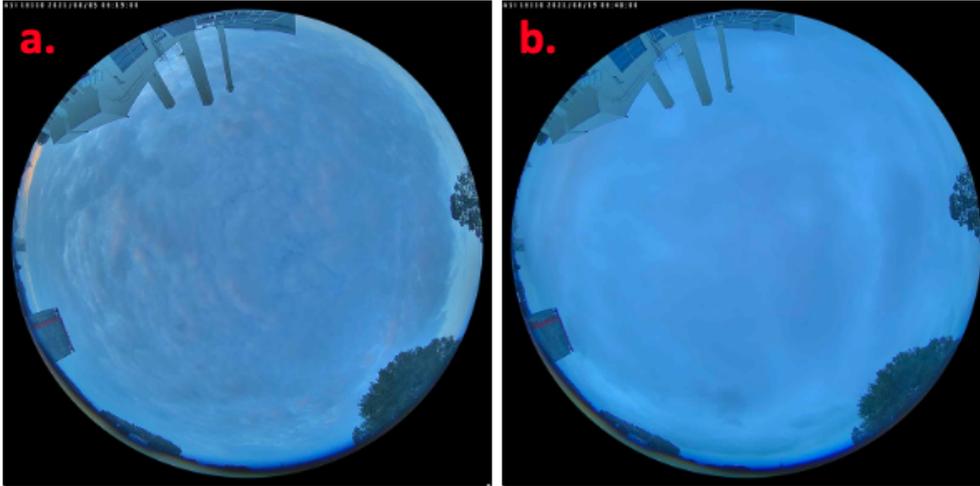


Figure 5.3: The two example where the CC algorithm misclassified the CC %. (a) ASI is taken 4:19 (UTC) in the morning of the fifth of August in 2021. (b) ASI is taken 4:40 (UTC) in the morning of the 19th of August in 2021.

Apart from the CC, one can also look at the VI for both locations and for the different classes. When the VI is 1, it indicates that the minutely difference in GHI between the clear sky situation and the measured situation is exactly the same. When the VI is smaller than one, it indicates that the measured GHI difference is smaller than for clear sky. This would indicate that the measured GHI over time is not changing much. When the VI is higher than one, it indicates that the measured GHI changes more than the Clear Sky conditions, which can indicate a difference in the sky situation. Also, a higher VI overall means that the model is more difficult to predict, since the GHI changes per minute are higher [Hartmann \[2020\]](#). From Figure 5.4 becomes clear that the overall VI is higher for location Delft compared to location Folsom, when the box-and-whiskers plots for both locations and classes will get compared.

5.1.1 Classification sensitivity analysis

The classification was done based on the Clear Sky Index (K_c) and the Cloud Cover (CC). For these two variables it is important to find the boundary values which separates the classes from each other. In Figure 5.5, one can see which combination of K_c with the CC corresponds to which class. There are however also combinations where there is no classification made. This is the case for relatively low CC (< 0.25) and the K_c lower than 0.75. This is done since it is expected that when the K_c and CC are low, the specific time moment will get incorrectly classified as clear sky. The same holds for the moments with CC higher than 0.75 and K_c higher than 0.4. The amount of sky covered by clouds and irradiance have an inverse connection, according to early researchers [Kimball \[1919\]](#). For that reason, it is expected that there are not much situations with low CC and low k_c or high CC and high k_c .

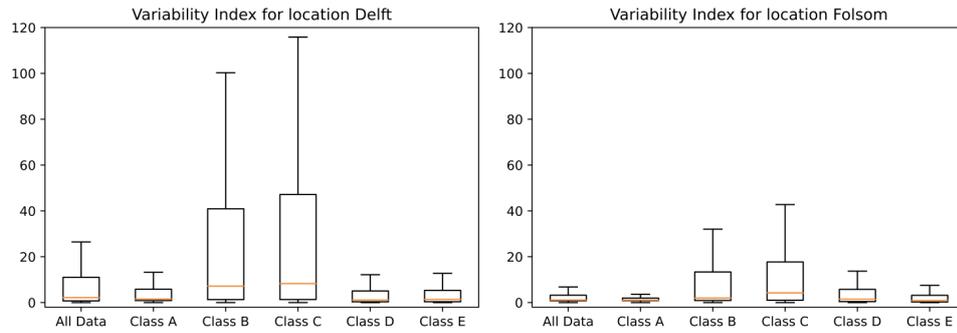


Figure 5.4: The Variability Index (VI) shown as box-and-whiskers plot for all the data and per class for location Delft on the left and location Folsom on the right.

When these situation do occur, it is expected that the CC is calculated incorrectly. For that reason, when these situations occur they get classified as 'No Class'.

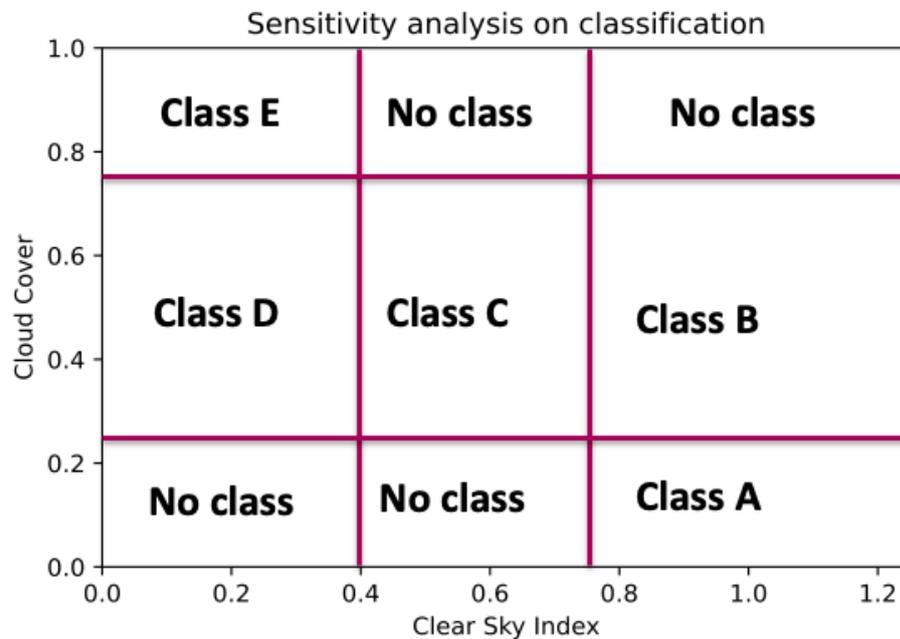


Figure 5.5: Overview how the classes are classified based on the K_c and CC.

The boundary values of the CC are based on the amount of Okta's. Where Class A (clear sky) is classified for a CC up until 2 Okta's, which corresponds to 25% CC. The other boundary value is for 6 okta's, which corresponds to 75% CC. This boundary values are chosen based on the National Weather Service (NWS). They state that between 0 and 2 Okta's, the sky can be classified as clear sky, between 2 and 6 Oktas partly clouded and between 6 and 8 Oktas overcast NWS [2022a]. This distribution is also applied in the paper of Kazantzidis et al. [2012a]. It is however also possible to make the boundary at 1 okta and 7 okta. For example, Li and Lam [1999] did research on the climatic parameters and their applications in the classification of sky condition and used the boundary values of 1 and 7 okta to make classes for clear sky, partly clouded and overcast. The boundary values for the k_c are different when it gets compared with multiple papers. Mathiesen and Kleissl [2011] classified $k_c < 0.4$ as cloudy conditions and when the k_c was around 0.75 or higher, the sky condition is classified as clear sky, since the future predictions correlated well with the clear sky irradiance. Between those two values,

the partly clouded conditions, is the situation with the highest RMSE when making irradiance predictions due to the irradiance variation (caused by the partly clouded conditions). Instead of the k_c , most papers use the Clearness Index (CI), which is the fraction of the actual total irradiance the theoretical maximum irradiance during the same period. The maximum CI is one, while the k_c can also be higher than one. Cucumo et al. [2007] used the boundary values of 0.2 and 0.65 based on the CI. Li and Lam [1999] used the boundary values of 0.15 and 0.7, and Muneer et al. [2007] used the clear sky boundary value of 0.6. However, Lauret et al. [2022] compared the solar forecast outcomes of the k_c with the CI, and showed that the use of the k_c gave a higher FS when comparing the prediction loss based on the RMSE. For this reason, the k_c will be used to make the classifications. Finally, to change the boundary values of the CC and k_c will make the 'no class' condition smaller and the Recall higher (more ASI's gets classified). The problem is that the increase of Recall can decrease the Precision of the model, since it is expected that more errors (wrong classified images) occur in the 'no-class' area.

5.2 PREDICTION RESULTS

5.2.1 Results without classification

For the prediction results, first one will look at the situation when no classification are used. When the situation without classification first gets analysed for both location, it can get compared with the situation when the classification is used. The results of the MAE, RMSE, nMAE, nRMSE and FS for the four prediction models are depicted in 5.3 for the location Delft.

Model	MAE [W/m^2]	RMSE [W/m^2]	nMAE	nRMSE	FS [%]
Smart Persistence (= baseline)	55.36	116.79	21%	44.3%	0
Persistence	57.76	118.65	21.9%	45%	-4.3%
ARIMA	67.86	89.34	25.8%	33.9%	-23.6%
CNN	44.28	84.03	16.8%	31.9%	20%

Table 5.3: The prediction results for the four different models in comparisons based on the MAE, RMSE, nMAE, nRMSE and FS for location Delft.

From the table becomes clear that the CNN models performs best, with an 20% FS improvement for the whole prediction for 21-minutes compared to the SP model. This is also the case when there will be looked at the minutely performs of the MAE in Figure 5.6 and the quantile 95% in Figure 5.7. On the other hand, the FS of the persistence and ARIMA are negative, which indicates that the SP model are performing better over the 21-minute prediction based on the MAE. Furthermore, from Table 5.3 one can see that the RMSE of the ARIMA model is lower than the baseline model, while the MAE is higher. The greater difference between the MAE and RMSE indicates a greater variance in the individual errors. This is also what can be see in Figure 5.7 when looking at the quantile 95%. After around 8 minutes, the quantile 95% of the SP model becomes higher than the ARIMA model, while the MAE per minute for the ARIMA (as depicted in Figure 5.6) is higher for all minutes. However, when this gets compared to the CNN model, the overall performance of the CNN still remains the best compared to the other three models.

Also, the prediction for the location Folsom was done and the results are depicted in Table 5.4.

Also for the situation of Folsom, the CNN outperforms all other prediction models. It is interesting to compare the outcome of the nMAE and nRMSE of the location Folsom in Table 5.4 with the outcome of location Delft in Table 5.3. Since the mean GHI for the location Folsom is $411.4W/m^2$ while the mean GHI for Delft is $263.5W/m^2$

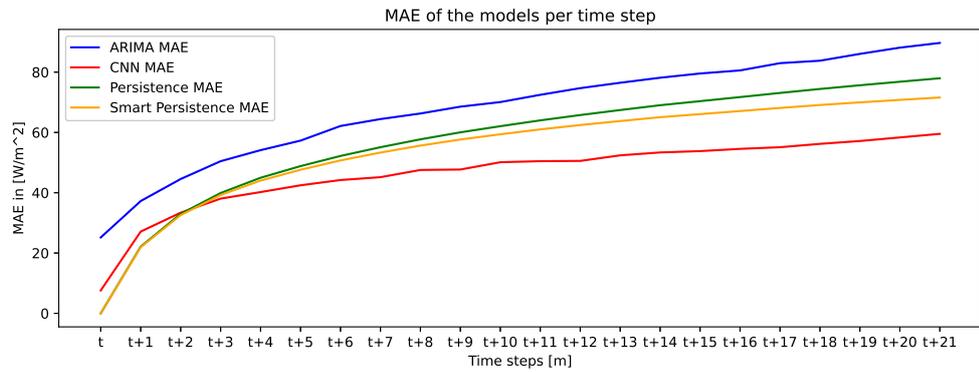


Figure 5.6: The 21-minute predictions of the 4 models namely: Persistence model, Smart Persistence model, ARIMA model & CNN model. The final outcome of the MAE per minute for the four prediction models for location Delft.

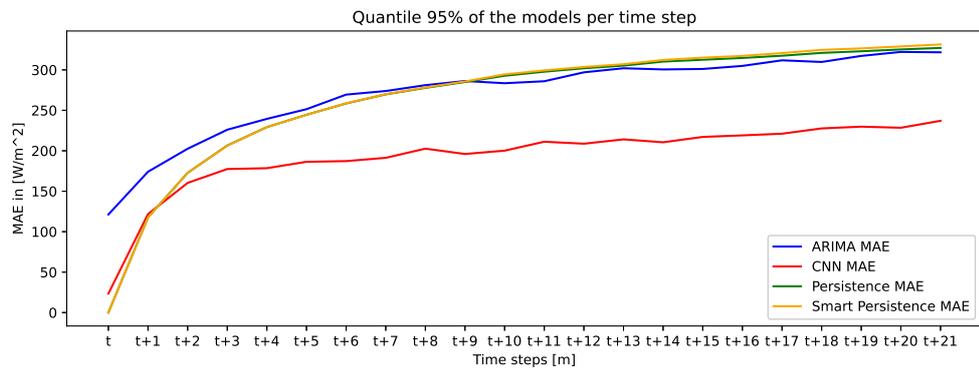


Figure 5.7: The 21-minute predictions of the 4 models namely: Persistence model, Smart Persistence model, ARIMA model & CNN model. The final outcome of the quantile 95% per minute for the four prediction models for location Delft.

and the overall MAE for location Delft is higher than the location Folsom. When the nMAE of the CNN model will get compared, one can see that for location Delft the nMAE is 2.9 times higher than for situation Folsom!

5.2.2 Results with classification

For the results with classification the four prediction models will get compared with the outcome of the MAE per classification. As previously explained, the classification is done based on 5 groups, where Class A represents clear sky, group E overcast and Class B, C, D partly cloudy. In Table 5.5, the results of the four prediction models per classification for location Delft is shown. For the CNN model two predictions were made. The first one 'CNN: class afterwards' is when the classification is done after the prediction. This indicates that the prediction is done based on the whole dataset, and after the prediction all the outcomes are labeled in the classes. The second one 'CNN: class forwards' is when the classification is done before the prediction. To compare these two with each other one can see the improvement of the prediction when the classification is applied before the prediction is done.

It is important to note that the results for the CNN forwards and afterwards that are showed in Table 5.5 are done with a dataset of 19000. This is because Class D (for location Delft) only has an dataset of that size. The outcome of the prediction with other sizes of datasets are shown in subsection 5.2.3. The comparison between the CNN when making prediction forwards and afterwards per class are shown in

Model	MAE [W/m^2]	RMSE [W/m^2]	nMAE	nRMSE	FS [%]
Smart Persistence (= baseline)	27.12	75.22	6.6%	18.3%	0
Persistence	39.68	79.73	9.6%	19.3%	-46.3%
ARIMA	46.89	58.54	11.4%	14.2%	-72.9%
CNN	23.87	54.12	5.8%	13.2%	12%

Table 5.4: The prediction results for the four different models in comparisons based on the MAE, RMSE, nMAE, nRMSE and FS for location Folsom.

Model	Class A	Class B	Class C	Class D	Class E
Smart Persistence (baseline)	32.4	108.71	92.21	56.40	30.99
Persistence	43.84	104.21	98.61	57.21	31.61
ARIMA	65.06	104.26	112.33	60.25	36.64
CNN: class afterwards	43.50	90.25	87.84	54.04	31.97
CNN: class forwards	30.94	86.26	86.60	48.10	29.29

Table 5.5: The results of the MAE per classification for the four prediction models for location Delft. For the CNN model 2 predictions were made. The first one 'CNN: class afterwards' is when the classification is done after the prediction. The second one 'CNN: class forwards' is when the classification is done before the prediction.

5.6. The total improvement for classifying the data before making predictions for this dataset is 8.6%.

Model	Class A	Class B	Class C	Class D	Class E	Total
CNN: afterwards	43.50	90.25	87.84	54.04	31.97	61.52
CNN: forwards	30.94	86.26	86.60	48.10	29.29	56.24
Improvement	28.7%	4.4%	1.4%	11%	8.4%	8.6%

Table 5.6: The total improvement for making classifications forwards and afterwards based on a dataset of 19000 at the location Delft .

This same analysis is done for the location **Folsom**. It is important to note that the results for the CNN forwards and afterwards that are showed in Table 5.5 are done with a dataset of 9900. This is because Class D (for location Folsom) only has an dataset of that size. The total improvement for classifying the data before making predictions for this dataset is 29.3%, as shown in Table 5.7.

For every class (A till E) the minutely MAE and quantile 95% are plotted for the location Delft and are shown in the Appendix A.1. One can see that also for minutely data over the whole 21-minute prediction, that when the data is classified first the outcome improved when it gets compared to the SP model and the CNN that gets classified afterwards.

5.2.3 Prediction sensitivity analysis

Sensitivity for five classes

Table 5.6 in subsection 5.2.2 already showed the improvement when for all classification a dataset of 19000 is used. The same is showed for Folsom where a dataset of 9900 was used, as shown in Table 5.7. In this section, for the sensitivity analysis based on the five classes that are being made, three different datasets will be used. The first dataset will consist of all the available data per classification for location Delft. In Table 5.8 this dataset is called 'Delft total (after/for)wards'. This indicated that the amount of data that gets compared per class is not equally divided. The same will be done for the location Folsom, this dataset is named 'Folsom total (after/for)wards'. Finally, a combined dataset named 'Combined (after/for)wards' is

Model	Class A	Class B	Class C	Class D	Class E	Total
CNN: afterwards	47.25	63.5	78.76	62.75	35.27	57.47
CNN: forwards	24.76	46.56	60.56	44.86	26.52	40.62
Improvement	47.6%	26.7%	23.1%	28.5%	24.8%	29.3%

Table 5.7: The total improvement for making classifications forwards and afterwards based on a dataset of 9900 at the location Folsom.

made for location Delft & Folsom to have a prediction dataset. All the results with the total improvement per dataset is shown in Table 5.8.

Datasets	Class A	Class B	Class C	Class D	Class E	Total improvement [%]
Delft total afterwards	29.81	74.36	80.23	44.68	30.23	
Delft total forwards	27.14	74.56	77.85	44.98	29.57	2.1%
Folsom total afterwards	20.35	46.39	60.88	47.42	28.16	
Folsom total forwards	17.05	39.3	59.08	44.83	24.55	9.1%
Combined afterwards	29.6	71.25	79.48	49.11	31.67	
Combined forwards	24.1	68.6	75.47	47.05	29.43	6.3%

Table 5.8: Results of the total improvement when all data per classification is used.

One can see that the improvement for all locations are positive when the data gets classified first before making predictions. It is important to note that this is even the case when the amount of data is not of the same size. For the combined dataset it can be expected that the total improvement lays between the two datasets.

Sensitivity for three classes

Besides the classification that is done based on the five classes, classifications have also been done on the basis of three classes. This is done by combining Class B,C,D into one partly clouded class namely Class 2. The three classes are separated as Class A (clear sky), Class 2 (partly clouded) and Class E (overcast). Five different datasets are used for making these predictions. The first prediction is done for location Delft where the dataset consist of 79000 time steps. This is done since this is the maximum amount of data that all classification can have the same amount of data. The class is called '79000 Delft (after/for)wards'. The second dataset consist of 23000 time steps for location Folsom for the same reason as the previous one. For the third dataset the location will get combined and a dataset of 100,000 time steps can be made. The last two datasets consist of all the available data per classification, and the afterwards prediction consist of all available data without classification. This is done for location Delft and Folsom. The results of these five datasets are shown in Table 5.9.

Dataset	Class A	Class 2	Class E	Total improvement [%]
79000 Delft afterwards	29.93	74.31	30.38	
79000 Delft forwards	27.92	50.82	29.63	19.5%
23000 Folsom afterwards	22.27	60.99	35.18	
23000 Folsom forwards	18.66	67.28	24.77	6.5%
100.000 Combined afterwards	27.28	64.28	29.41	
100.000 Combined forwards	26.64	48.96	27.59	14.7%
Total Delft afterwards	29.81	65.50	30.23	
Total Delft forwards	27.14	46.04	29.57	18.2%
Total Folsom afterwards	20.35	52.55	28.16	
Total Folsom forwards	17.05	43.77	24.55	15.5%

Table 5.9: Results of the total improvement when making classes of three instead of five for location Delft, Folsom and combined.

It is interesting to compare the improvement of Class 2 to the improvement of Classes B, C & D. When the 'Total Delft (after/for)wards' dataset will be taken as example, the total improvement of 5 classes is 2.1% while for 3 classes it is 18.2%. This while the only difference is made in Class 2, since Class A and E are the same for the five and three classes. This indicates that the overall improvement with only three classes is more than for five classes for location Delft. To go in more detail, also the MAE per time step and quantile 95% are taken into consideration. In Figure 5.8, one can see the MAE per time step for the whole 21-minute prediction. The overall improvement is significantly greater compared to the minutely improvement for classes B, C & D shown in Figures A.3, A.5 & A.7. The same is done with the quantile 95% for class 2 and is shown in Figure 5.9 which can be compared with the quantile 95% of classed B, C & D shown in Figure A.4, A.6 & A.8.

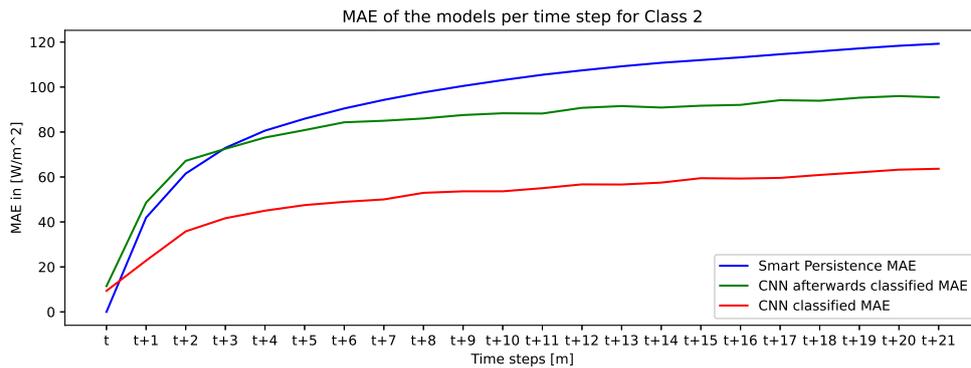


Figure 5.8: The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class E is not applied, so afterwards classified.

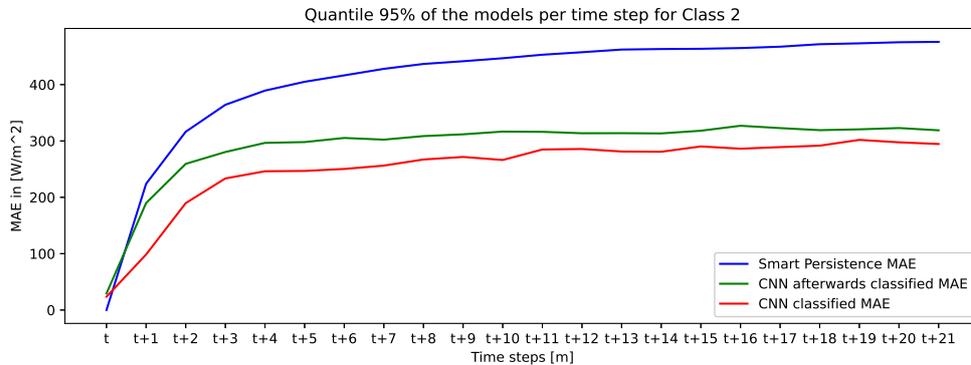


Figure 5.9: The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class E is not applied, so afterwards classified.

For location Folsom, the same comparing can be done by looking at the difference between Class 2 and the previous Classes B, C & D. First, for the dataset of 23000, the class 2 for the Folsom location does not even improve. However, when all the data is used, the total outcome class 2 when classify before predictions improves the outcome. It could be the case that a bigger dataset is required to get the desired improvement for partly clouded conditions, since more different weather conditions are classified in this group.

However, for both locations, the total improvement when making three classes is bigger than when making five classes. When three classes will be made, the datasets also increases. It could be possible that when the total classification dataset on a specific location increases the total improvement.

Ablation Sensitivity

For the ablation sensitivity, different input variables are used to make 21-minute predictions. These results will get compared with the original CNN model, named 'CNN - Main', when no classifications are applied. This CNN-main model is all the data that is available for location Delft. 'CNN - HSV' is made with the same dataset, the only difference is that the original input consists of RGB images whereas for this model the images are HSV (differences are explained in subsection 3.4.2. For the third model, which is called 'CNN - CC', the CC is added as an extra variable with the same dataset as 'CNN - Main'. For the final prediction all the available data is used to make predictions for only the location Delft. So, the training data of Delft & Folsom are combined to make predictions only for the location Delft.

Model	MAE [M/m^2]	nMAE	FS [%]
CNN - Main	44.28	16.8%	0
CNN - HSV	61.27	23.3	-38.4%
CNN - CC	45.51	17.3%	-2.8%
CNN - All	48.55	18.4	-9.6%

Table 5.10: The performance of the CNN-Main model compared to the CNN-Main models with the ablation of inputs is depicted.

As shown in Table 5.10, all the other results compared to the CNN-main results in a negative FS. This indicates that there is no improvement based on the final MAE outcome for the 21-minute prediction. Even when all data for both locations will be used as training set for testing the outcome of location Delft, results in worse outcomes.

6.1 CLASSIFICATION COMPARISON

In this section the outcome of the classification model will get compared with other papers. First of all, it is important to go into detail about the classification method that is used in this research. For this research the CC algorithm of [Zuo et al. \[2022\]](#) is used and slightly adjusted. The adjustment is that for partly clouded conditions, the SPS method of [Liu et al. \[2015\]](#) is used instead of the global Otsu threshold method. This was done because the paper of [Liu et al. \[2015\]](#) showed more promising results which were also in line with my own findings during this research. The paper of [Zuo et al. \[2022\]](#) used the algorithm as directly input for the prediction model. This means that the algorithm was not validated and no classification were made with the CC outcome.

For that reason, other papers needs to be used to compare the classification outcome. Two papers that made classification based on CC in ASI's are [Kazantzidis et al. \[2012a\]](#) and [Xie and Yiren \[2019\]](#). In the first paper the classification for the CC was done in the same way as this research. Which is classifying the ASI's into three groups based on the CC. This approach yields 83% within 1 okta and 94% within 2 oktas of the examined images that, respectively, agree to within 1 and 2 oktas with visual weather measurements at a nearby meteorological station. The paper of [Xie and Yiren \[2019\]](#) yields 80.6% within 1 okta and 93.9% within 2 oktas. In this paper the CC is found via a CNN model (i.e., the optimised U-Net model) that detect the clouds in the ASI's. The input of the CNN model were 400 labeled ASI's. The label corresponds to the amount of oktas that corresponds to the cloud situation in the ASI. The total cumulative difference between the two papers and this research are shown in Table 6.1.

Okta difference	This study: Delft	This study: Folsom	Xie and Yiren [2019]	Kazantzidis et al. [2012a]
0	53.6%	50%	44.9%	Unknown
1	87.2%	73.6%	80.6%	83%
2	95.6%	92%	93.9%	94%

Table 6.1: Comparing between the studies of [Xie and Yiren \[2019\]](#) and [Kazantzidis et al. \[2012a\]](#) and this research based on the difference in oktas with the CC algorithm compared to manual observation.

For calculating Clear Sky Index (k_c), the Simplified Solis method of [Ineichen \[2008\]](#) is used. The Simplified Solis method is one of the most broadly used clear sky calculation methods. In the paper of [Antonanzas-Torresa et al. \[2019\]](#), 70 different Clear sky models where compared and this method was one of the best performing methods. On top of that, in the pvlb-python algorithms this is one of the three clear sky models that can be used, while the other two performed worst in the comparison paper of [Antonanzas-Torresa et al. \[2019\]](#).

However, other methods to classify images could also have been applied. In the paper of [Hartmann \[2020\]](#), six different methods to categories the sky conditions are compared and analysed. In this methods, the cloud cover was not used as input for making the classifications. However, the cloud cover is used for validating afterwards if the classification of the sky conditions were done correctly. For the six

methods that were used, all the sky conditions are classified, which would result in an Recall of 100% in this research (number of classified divided by total number). The Recall in this research varies between 82% for location Delft and 94% for location Folsom. The validation was done by dividing the sky conditions into three groups based on the cloud cover. The groups were clear sky (0 to 1 okta), partly clouded (1 to 7 okta) & overcast (7 to 8 okta). The method with the highest precision (correctly classified/total classified) was 81.3%. The precision in this research varies between 95% for location Delft and 91% for location Folsom. The most accurate method from the paper of [Hartmann \[2020\]](#) makes the 5 classes based on the CSI and the VI, as shown in Table 6.2.

Class	VI	CSI
Clear	< 2	≥ 0.5
Overcast	< 2	< 0.5
Mild	$2 < VI < 5$	n.a
Moderate	$5 \leq VI < 10$	n.a
High	≥ 10	n.a

Table 6.2: How the best performing classification method of [Hartmann \[2020\]](#) is made

This classification can also be applied for the location of Delft and Folsom and are shown in Figure 6.1.

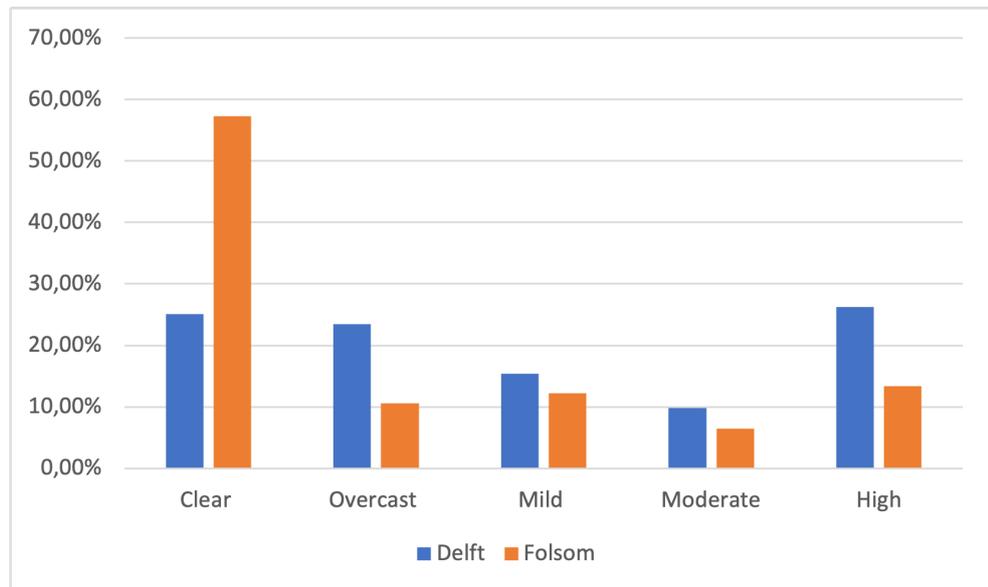


Figure 6.1: The outcome of the [Hartmann \[2020\]](#) classification applied for location Delft and Folsom.

6.1.1 All Sky Camera

The All Sky Camera is the Camera that is used for making the All Sky Images. In this research, the same method has been used to calculate the CC and k_c in order to make classifications of the sky conditions. For both locations, the All Sky Camera creates every minute a circular fish-eye image of the sky. The output for both locations is an RGB image with the pixel size of 1536 by 1536. For this reason, it is assumed that the classification method can be applied for both locations. However, in the paper of [Wacker et al. \[2015\]](#), the method of [Kazantzidis et al. \[2012a\]](#) was applied to calculate the CC at 4 different locations. The outcome of the results were very location dependent. When the algorithm is site-specifically trained, it has been reported that over 90% of the images were correctly categorised.

The mean success rate drops significantly to 50% to 80% when the algorithm is applied to the remaining sites. The reason could be due to the randomness of the image selection process, the type of used All Sky Camera and the researchers ability to classify the images. However, the results of two of the same camera's on different locations gave similar results.

On the other hand, the calibration of the Camera is also important in order to find the exact location of the sun in the ASI. [Urquhart et al. \[2015\]](#) found a method to calibrate the ASI in order to track the solar position in the images for an All Sky fish-eye camera. The results showed calibration performance ranged from 0.94 to 1.24 pixels RMSE. It is however first important that the camera is set up completely straight to the sky, which is not the case for the location Delft.

6.2 PREDICTION COMPARISON

In the previous research done by [Doodkorte \[2021\]](#), an extended study comparison is done for the used CNN model on the location Folsom. The study comparison was done with the papers of [Paletta et al. \[2021a\]](#) and [Kong et al. \[2020\]](#). The comparison was done based on the geographical location, Pre-processing, hardware limitations, forecast horizon, the input variables, dataset, deep learning architecture and the hyperparameters.

For the geographical location was stated that the CNN model for location Folsom performs better for clear sky conditions and worst for overcast situations because there are many more clear sky situations than overcast. This is indeed true when all the data is used for making this predictions. However, when the predictions are made with the same data size, the overcast prediction is better compared to the clear sky condition for the location Folsom. When this get compared with the location Delft, one can see that the total irradiance prediction for clear sky outperforms the overcast condition. This means that overall the model is able to better predict the clear sky conditions compared to the overcast conditions. When the clear sky outcome of Folsom gets compared with Delft, one can see that the overall predictions of Folsom outperforms location Delft. This is probably the case due to the less variance that occur during the day in the weather conditions for location Folsom compared to Delft. This becomes also clear when looking at the VI for both locations. With a low VI, the expected predictability increases and the model can make more accurate predictions. This becomes also clear from [Figure 6.1](#), since the total amount of situations which get classified as high are relatively more for location Delft compared to location Folsom.

For the pre-processing part, the biggest difference is the image input. For this research and [Kong et al. \[2020\]](#), the image input was 64x64x3 (RGB), whereas for [Paletta et al. \[2021a\]](#) the image input was 128x128x1 (grayscale). It is expected that higher sky resolutions ASI's performs better, but on the other hand performs colored images better than grayscale. Ideally, high resolutions colored ASI's would be implemented, but the problem that occurs are the hardware limitations. In this study, there was no availability of a powerful computer. Apart from that, the increase of resolutions would increase the computational time of the model. Since multiple predictions for both locations needed to be made and there was no powerful computer available, the input of the images were chosen to have the size of 64x64x3 (RGB).

The forecast horizon is for many papers different. However, most of the papers that make very short irradiance predictions are in the range of 0 to 30 minutes. The reason for 21-minutes in this research is explained in [Chapter 2](#). In addition, this 21-minute forecast range is chosen to compare the results with the MSc thesis of [Doodkorte \[2021\]](#). Within this short forecast horizon, the most common time

frequency are 1-minute. However, [Kong et al. \[2020\]](#) uses a time frequency of 30 seconds.

Furthermore, in this research multiple datasets and datasets sizes are used. It is generally known that overall more data for DL models results in better predictions. This is also shown in this research and of [Doodkorte \[2021\]](#). However, using the data of multiple locations in order to improve the prediction on one location decreases the final outcome, which is shown in this research. On the other hand, the input quality of the data can also improve the predictions. This can be done via three methods. The used input variables will change, the way of normalization of the variables change or a change in which data points are specifically used. In the MSc thesis of [Doodkorte \[2021\]](#) various input variables were tried in order to improve the final prediction. With the used CNN model, the clear sky irradiance and measured GHI came out to be the best performing prediction model. However, various papers use the CC as input for making irradiance predictions, as for example the paper of [Zuo et al. \[2022\]](#). In those papers, there is no comparison made between the outcome with and without CC. In this study, the CC was used as extra input, but the performance gave an -2.8% FS compared to the CNN- Main model. The last method to improve the quality of input is applied by making classification of the sky conditions before making irradiance predictions. Also, due to the classification method, the normalization per prediction is automatically changing. This is because the normalization happens inside the DL model, so for a different set of input a different normalization variation is applied.

At last, another important factor for the irradiance predictions is the DL architecture. In the MSc thesis of [Doodkorte \[2021\]](#) came out that the CNN model outperformed the RNN. However, other papers showed that the use of an RNN could improve the final irradiance predictions. This was also shown by [Paletta et al. \[2021a\]](#), were a LSTM-CNN structure outperforms the CNN-structure. The LSTM is used in order to keep track of past information when multiple images are implemented instead of one image. So, it is expected that DL models which uses a sequence of ASI's can improve the final irradiance predictions. However, in this research this is not done.

6.3 FUTURE WORK

In the previous two sections the classification method and prediction method were discussed. In this section one can see how the areas of concern can be addressed for future work. The most important points for the future work of this research are:

- A sequence of 5 or more high resolution color images and multiple past measured irradiances as input in order to improve the final irradiance predictions.
- An improved classification model which can classify all sky conditions situations.
- An improved classification model that can distinguish thick and thin clouds from each other.

During this research it was important to find out if it was possible to improve the final irradiance prediction when making sky classification first. This DL model use however only the input of one ASI at the time. It is not known if the classification method would still improve the outcome when a sequence of ASI are used to make predictions. It could also be possible that a hybrid of DL models could be used for different classes. This is also not tried before in other research up until now. On the other hand, the classification method could still be approved. First of all, it is important to use a classification method which can classify all sky conditions throughout the year. Up until this moment, it was not possible to receive sufficient precision results when classifying all sky conditions.

Furthermore, it should be possible to distinguish between thick and thin clouds in the ASI, which is already done by Ghonima et al. [2012]. In the paper of Masuda et al. [2019] found even a way to retrieve the optical cloud thickness from ASI by using an CNN model. The optical cloud thickness could then be used to predict the cloud movement. Chow et al. [2011] found a promising method to make this prediction of the cloud movement. When this both methods could be implemented in a DL model to make irradiance prediction, it is expected that the irradiance predictions will get improved. This is however not yet accomplished in any research.

At last, there are at this moment also other DL methods that can be used for interpreting the ASI's. Video prediction algorithms which forecast future visuals based on a series of past images Guen and Thome [2020], are promising techniques. Another interesting approach is employing DL to quickly and effectively solve NWP problems Li et al. [2020]. Consequently, it is anticipated that new techniques will be created in the upcoming years.

6.4 CONCLUSION

In order to maintain grid stability when more PV installations will get integrated into the electricity net, the importance of short term irradiance predictions are expected to increase. The irradiance prediction is location dependent and the biggest influence of variance in the outcome are the clouds. This indicates that the irradiance predictions are sky condition dependent. For that reason, in this study first a classification method is applied to classify the ASI into five groups based on sky conditions. During this study the classification method and the irradiance predictions were elaborated for two different locations. The total improvement when making classifications before making the irradiance prediction on all the data for location Delft was 2.1% and for location Folsom 9.1 %. However, when this were applied for three classes, the total improvement on all the available data for location Delft was 18.2% and Folsom 15.5%. Furthermore, when using the cloud cover as extra input variable for the DL model didn't improve the final outcome compared to the main model. The same applies for using the data for both locations to make the irradiance predictions on one location. Up until now, the use of classifying the all sky images before making irradiance predictions is not done before. For that reason, it is important to find out if this could also be applied on different DL models in order to improve the final prediction outcome. To conclude, the control of solar power plants will be made easier with the help of very short-term irradiance predictions by the use DL models. This will lessen the issues related to the intermittency of renewable energy sources like solar energy. As a result, PV energy will be even better received, making it easier for it to be integrated into power networks and assisting in the transformation of the energy industry to an electric system devoid of emissions.

BIBLIOGRAPHY

- 60904-3, I. (2008). Photovoltaic devices – part 3: Measurement principles for terrestrial photovoltaic (pv) solar devices with reference spectral irradiance data.”. *IEC*.
- Aguiar, L. M. and Díaz, F. (April 2018). Solar radiation forecasting with statistical models. *Green Energy and Technology*.
- Alessandro Niccolai, A. N. (November 2020). Sun position identification in sky images for nowcasting application. *Forecasting*.
- Alzahrana, A., Shamsia, P., Daglib, C., and Ferdowsia, M. (2017). Solar irradiance forecasting using deep neural networks. *Procedia Computer Science*, 114:304–313.
- Annsky (2 june 2022). *deposit photos*.
- Antonanzas-Torresa, F., Urracab, R., Poloc, J., Perpiñán-Lamigueirod, O., and Escobara, R. (2019). Clear sky solar irradiance models: A review of seventy models. *Elsevier*.
- Arthur, D. and Vassilvitskii, S. (January 2007). k-means++: The advantages of careful seeding. *Symposium on Discrete Algorithms*.
- Barbieri, F., Rajakaruna, S., and Ghosh, A. (2017). Very short-term photovoltaic power forecasting with cloud modeling: A review. *Elsevier*, page 243.
- Biswal, A. (September 2022). Convolutional neural network tutorial. *Simplilearn*.
- Blanc, P. (2017). Short-term solar power forecasting based on satellite images. *Renewable Energy Forecasting*.
- Blazek, M. and Pata, P. (June 2015). Colour transformations and k-means segmentation for automatic cloud detection. *Meteorologische Zeitschrift*.
- Brown, T. (May 20, 2022). Atmosphere. *Education National Geographic*.
- Brownlee, J. (April 2020). How do convolutional layers work in deep learning neural networks? *Deep learning for computer vision*.
- Brownlee, J. (December 2016). How to make baseline predictions for time series forecasting with python. *Time Series*.
- Brownlee, J. (February 2021). Difference between backpropagation and stochastic gradient descent. *Machine Learning mastery*.
- Brownlee, J. (January 2017). How to create an arima model for time series forecasting in python. *Time Series*.
- Buis, A. (February 2020). Milankovitch (orbital) cycles and their role in earth’s climate. *NASA’s Jet Propulsion Laboratory*.
- Carter, M. and Bruce, R. (2009). *Op Amps for Everyone*. Texas Instruments.
- Chauhan, N. S. (August 2019). Introduction to image segmentation with k-means clustering. *KD Nuggets*.
- Chauvin, R., Nou, J., Thil, S., and Grieu, S. (July 2015). Modelling the clear-sky intensity distribution using a sky imager. *Solar Energy*.
- Chhabria, M. (July 2021). K-means clustering algorithm & it’s use cases. *Medium*.

- Chin, J. (September 2021). Hue, saturation, value: How to use hsv color model in photography. *Master Class*.
- Chow, C. W., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Shields, J., and Washom, B. (November 2011). Intra-hour forecasting with a total sky imager at the uc san diego solar energy testbed. *Solar Energy*.
- Christopher, A. (February 2021). K-nearest neighbor. *Medium*.
- Coimbra, C. F. and Kleissl, J. (2013). Overview of solar-forecasting methods and a metric for accuracy evaluation. pages 171–192.
- Coleman, J. and Law, K. (2015). Reference module in earth systems and environmental sciences. *Science Direct*.
- Crisosto, C. (2019). Autoregressive neural network for cloud concentration forecast from hemispheric sky images. *Int. J. Photoenergy*.
- Cucumo, M., Ferraro, V. D. R., Kaliakatsos, D., and V. Marinelli (November 2007). Correlations of global and diffuse solar luminous efficacy for all sky conditions and comparisons with experimental data of five localities. *Building and Environment*.
- Dabakoglu, C. (December 2018). What is support vector machine (svm) -with python. *Medium*.
- Dertat, A. (August 2017). Applied deep learning - part 1: Artificial neural networks. *Towards Data Science*.
- Diagne, M., David, M., Lauret, P., Boland, J., and Schmutz, N. (2013). Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews*, page 71.
- Doodkorte, P. (June 2021). Short-term irradiance forecasting using all-sky images and deep learning. *Sustainable Energy Technology*.
- EERE (2021). Solar radiation basics. *Solar Energy Technologies Office*.
- EERE (2022). Solar futures study. *Solar Energy Technologies Office*.
- Gao, H. and Liu, M. (2022). Short-term solar irradiance prediction from sky images with a clear sky model. *Computer Vision Foundation*.
- Gholamy, A., Kreinovich, V., and Kosheleva, O. (February 2018). Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. *Scholar Works*.
- Ghonima, M. S., Urquhart, B., Chow, C. W., Shields, J. E., Cazorla, A., and Kleissl, J. (November 2012). A method for cloud detection and opacity classification based on ground based sky imagery. *Atmospheric Measurements Techniques*.
- Goswami, B. and Bhandari, G. (December 2011). Automatically adjusting cloud movement prediction model from satellite infrared images. *IEEE India Conference*.
- Guen, V. L. and Thome, N. (2020). A deep physical model for solar irradiance forecasting with fisheye images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*.
- Gupta, D. (January 2020). Fundamentals of deep learning – activation functions and when to use them? *Analytics Vidhya*.
- Hans, C. (July 2019). Very short term time-series forecasting of solar irradiance without exogenous inputs. *Research Gate*.

- Harikrishnan, N. B. (December 2019). Confusion matrix, accuracy, precision, recall, f1 score. *Medium*.
- Hartmann, B. (July 2020). Comparing various solar irradiance categorization methods – a critique on robustness. *Renewable Energy*.
- Heinle, A. and A. Macke, A. S. (6 May 2010). Automatic cloud classification of whole sky images. *Atmospheric Measurement Techniques*.
- Holmgren, W., Hansen, C., and Mikofski, M. (September 2018). pvlib python: a python package for modeling solar energy systems. *Journal of open source network*.
- Hukseflux (2006).
- Hyndman, R. J. and Athanasopoulos, G. (April 2018). *Forecasting: Principles and Practice*. Texas Instruments, Melbourne, Australia.
- IBM (April 2020). What is artificial intelligence? *Cloud Education*.
- IEA-PVPS (October 2017). Snapshot of global pv markets 2016.
- Ineichen, P. (2008). A broadband simplified version of the solis clear sky model. *Elsevier*.
- Jain, V. (December 2019). Everything you need to know about “activation functions” in deep learning models. *Towards Data Science*.
- Jarvela, M. and Kari Lappalainen, S. V. (September 2018). Cloud enhancement phenomenon and its effect on pv generators. *35th European Photovoltaic Solar Energy Conference and Exhibition*.
- Jawahar, P. and Anoop, P. (2022). Foundations of artificial intelligence & machine learning. *mylearningsinaiml*.
- Jayadevan, V. T., Rodriguez, J. J., and Cronin, A. D. (Feb 2015). A new contrast-enhancing feature for cloud detection in ground-based sky images. *Journal of Atmospheric and Oceanic Technology*.
- Jeong, J. (January 2019). The most intuitive and easiest guide for convolutional neural network. *Towards Data Science*.
- Kanagasundaram, A., Valluvan, R., and Kaneswaran, A. (March 2019). Solar irradiance forecasting using deep learning approaches. *Research Gate*.
- Kato, T. (2016). Prediction of photovoltaic power generation output and network operation. *Integration of Distributed Energy Resources in Power Systems*.
- Kazantzidis, A. (2017). Short-term forecasting based on all-sky cameras. *Renewable Energy Forecasting*.
- Kazantzidis, A., Bais, A. F., Tzoumanikas, P., and Fotopoulos, S. (September 2012a). Cloud detection and classification with the use of whole-sky ground-based images. *Atmospheric Research*.
- Kazantzidis, A., Tzoumanikas, P., Blanc, P., Massip, P., Wilbert, S., and Ramirez-Santigosa, L. (2017). Short-term forecasting based on all-sky cameras. *Renewable Energy Forecasting*.
- Kazantzidis, A., Tzoumanikas, P., F.Bais, Fotopoulos, S., and G.Economoub (September 2012b). Cloud detection and classification with the use of whole-sky ground-based images. *Atmospheric Research*.
- Kher, A. (2022). Solar analemma. *Time and Date*.
- Kimball, H. (December 1919). Variations in the total and luminous solar radiation with geographical position in the united states. *Monthly Weather Review*.

- Kingma, D. and Ba, J. (December 2014). Adam: A method for stochastic optimization. *International Conference for Learning Representations*.
- KNMI (2021). Grafieken van het lopende jaar. *Koninklijk Nederlands Meteorologisch Instituut, Ministerie van Infrastructuur en Waterstaat*.
- Kong, W., Jia, Y., Dong, Z. Y., Meng, K., and Chai, S. (November 2020). Hybrid approaches based on deep whole-sky-image learning to photovoltaic generation forecasting. *Applied Energy*.
- Kreuter, A., Zangerl, M., Schwarzmann, M., and Blumthaler, M. (17 January 2009). All-sky imaging: a simple, versatile system for atmospheric research. *Optical Society of America*.
- Kumari, P. and Toshniwal, D. (2021). Deep learning models for solar irradiance forecasting: A comprehensive review. *Journal of Cleaner Production*, page 2.
- Kumari, P. and Toshniwal, D. (May 2020). Extreme gradient boosting and deep neural network based ensemble learning approach to forecast hourly solar irradiance. *Journal of Cleaner Production*, page 2.
- Kunert, B. W. J., Lorenz, E., Kramer, O., and Heinemann, D. (2016). Comparing support vector regression for pv power forecasting to a physical modeling approach using measurement, numerical weather prediction, and cloud motion data. *Solar Energy*, page 135:197–208.
- Lauret, P., Alonso-Suárez, R., Salle, J. L. G. L., and David, M. (Oktober 2022). Solar forecasts based on the clear sky index or the clearness index: Which is better? *Solar*.
- LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *neural computation*.
- Li, D. and Lam, J. (1999). An analysis of climatic parameters and sky condition classification. *Building and Environment*.
- Li, X., Wang, B., Qiu, B., and Wu, C. (2021). An all-sky camera images classification method using cloud cover features. *Atmospheric Measurements Techniques*.
- Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., A. Stuart, and Anandkumar, A. (2020). Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations*.
- Liu, S., Zhang, L., Zhang, Z., Wang, C., and Xiao, B. (Februari 2015). Automatic cloud detection for all-sky images using superpixel segmentation. *IEEE Geoscience and remote sensing letters*.
- Long, C. N., Sabburg, J. M., and J. Calbó, D. P. (01 May 2006). Retrieving cloud characteristics from ground-based daytime color all-sky images. *Journal of Atmospheric and Oceanic Technology*.
- Malone, T., Rus, D., and Laubacher, R. (December 2020). Artificial intelligence and the future of work. *MIT*.
- Masuda, R., Iwabuchi, H., Schmidt, K. S., Damiani, A., and Kudo, R. (August 2019). Retrieval of cloud optical thickness from sky-view camera images using a deep convolutional neural network based on three-dimensional radiative transfer. *Remote sensing*.
- Mathiesen, P. and Kleissl, J. (March 2011). Evaluation of numerical weather prediction for intra-day solar forecasting in the continental united states. *Solar Energy*.
- May, S. (19 December 2017). What are clouds? *NASA*.

- May, S. (January 2021). Revolve. *NASA*.
- Muneer, T., Gul, M., and Kubie, J. (November 2007). Models for estimating solar radiation and illuminance from meteorological parameters. *Journal of Solar Energy Engineering*.
- Ng, A. (March 2018). Student notes: Convolutional neural networks (cnn) introduction. *indoml*.
- NIOH (September 2022). Brain basics: The life and death of a neuron. *National Institute of Neurological Disorders and Stroke - Home*.
- Novati, A. (31 May 2022). *NASA climate kids*.
- NREL (2016). *Greening the grid*.
- NWS (2022a). Forecast terms.
- NWS (25 January 2022b). The four core types of clouds. *National Weather Service*.
- Otsu, N. (January 1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 62–66.
- Paletta, Q., Arbod, G., and Lasenby, J. (February 2021a). Benchmarking of deep learning irradiance forecasting models from sky images – an in-depth analysis. *Solar Energy*.
- Paletta, Q., Hu, A., Arbod, G., and Lasenby, J. (29 November 2021b). Eclipse : Envisioning cloud induced perturbations in solar energy. *University of Cambridge, UK*, page 5.
- Palubicki, W. (October 2020). Stormscapes: Simulating cloud dynamics in the now. *Research Gate*, page 175.
- Pedro, H., Larson, D., and Coimbra, C. (June 2019). A comprehensive dataset for the accelerated development and benchmarking of solar forecasting methods.
- Perez, R., Kivalova, S., Schlemmer, J., Jr., K. H., Renné, D., and E.Hoff, T. (2010). Validation of short and medium term operational solar radiation forecasts in the us. *Solar Energy*.
- Pokhrel, S. (September 2019). Beginners guide to convolutional neural networks. *Towards data science*.
- Popov, V., Tenison, C., and Ostarek, M. (March 2018). Practices and pitfalls in inferring neural representations. *Neurolmage*.
- Prabhakaran, S. (November 2019). Augmented dickey fuller test (adf test). *Machine learning plus*.
- Pramoditha, R. (March 2021). The relationship between ai, ml, nns and dl. *Medium*.
- Raj, A. (October 2020). Unlocking the true power of support vector regression. *Towards Data Science*.
- Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. *IEEE International Conference on Computer Vision*.
- Reno, M. J. and Hansen, C. W. (2015). Identification of periods of clear sky irradiance in time series of ghi measurements. *Sandia National Laboratories*, page 2.
- Roffo, G. (June 2017). Ranking to learn and learning to rank: On the role of ranking in pattern recognition applications. *Research Gate*.
- Russel, R. (2015). Layers of earth's atmosphere. *University Corporation for Atmospheric Research*.

- Schneider, D., Schwalbe, E., and Maas, H. (2009). Validation of geometric models for fisheye lenses. *ISPRS J. Photogramm.*
- Sharma, S. (September 2017). Activation functions in neural networks. *Towards Data Science*.
- Sharp, T. (March 2020). An introduction to support vector regression (svr). *Towards Data Science*.
- Siddiqui, T. A., Bharadwaj, S., and Kalyanaraman, S. (2019). A deep learning approach to solar-irradiance forecasting in sky-videos. *2019 IEEE Winter Conference on Applications of Computer Vision*.
- Skull, R. (2017). *Practical Meteorology: An Algebra-based Survey of Atmospheric Science*. University of British Columbia.
- Smets, A., Jager, K., Isabella, O., van Swaaij, R., and Zeman, M. (June 2015). *Solar Energy - The physics and engineering of photovoltaic conversion, technologies and systems*. UIT Cambridge, Delft University of Technology.
- Souza-Echer, M. P., Pereira, E. B., Bins, L. S., and Andrade, M. A. R. (March 2006). A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera. *Journal of Atmospheric and Oceanic Technology*.
- Spark, W. (2016). Climate and average weather year round in california. *NOAA's National Center for Environmental Information*.
- Stein, J. S., Hansen, C., and Reno, M. J. (May 2012). The variability index: A new and novel metric for quantifying irradiance and pv output variability. *Research Gate*.
- Stoffel, T., Renné, D., Myers, D., Wilcox, S., Sengupta, M., George, R., and Turchi, C. (September 2010). Best practices handbook for the collection and use of solar resource data. *NREL*.
- Szelag, M. (May 2020). Evaluation of cracking patterns in cement composites—from basics to advances: A review. *Materials*.
- Taboga, M. (2021). Statistical model, lectures on probability theory and mathematical statistics. *Kindle Direct Publishing*.
- UCAR (2019). Cloud types. *Center for Science Education*.
- Urquhart, B., Kurtz, B., and Kleissl, J. (January 2015). Sky camera geometric calibration using solar observations. *Atmospheric Measurements Techniques*.
- Vignola, F., Michalsky, J., and Stoffel, T. (2017). *Solar and infrared radiation Measurements*. CRC Press, Boca Raton.
- Visser, L., Lorenz, E., Heinemann, D., and Sarka, W. G. (2022). *Comprehensive Renewable Energy*. Elsevier, Delft University of Technology.
- Wacker, S., Gröbner, J., Zysset, C., Diener, L., Tzoumanikas, P., Kazantzidis, A., Vuilleumier, L., Stöckli, R., Nyeki, S., and Kämpfer, N. (January 2015). Cloud observations in switzerland using hemispherical sky cameras. *JGR Atmospheres*.
- Wei, X. Z. H., Wang, H., Zhu, T., and Zhang, K. (2019). 3d-cnn-based feature extraction of ground-based cloud images for direct normal irradiance prediction. *Solar Energy*.
- Xie, W. and Yiren, W. (July 2019). Development of an all-sky imaging system for cloud cover assessment. *Applied optics*.
- Yabuki, M., Shiobara, M., Nishinaka, K., and Kuji, M. (August 2014). Development of a cloud detection method from whole-sky color images. *Polar Science*.

- Yamashita, M., Yoshimura, M., and Nakashizuka, T. (2014). Cloud cover estimation using multitemporal hemisphere imageries. *Research Institute for Humanity and Nature*.
- Yang, J., Min, Q., Lu, W., Ma, Y., Yao, W., Lu, T., Du, J., and Liu, G. (February 2016). A total sky cloud detection method using real clear sky background. *Atmospheric Measurements Techniques*.
- yang Jun, Weitao, Ying, Wen, and Qingyong, L. (2010). An automatic ground based cloud detection method based on the local threshold interpolation. *Acta Meteorologica Sinica*.
- Zhang, A. B. and Gourley, D. (2009). Digitising material. *Creating Digital Collections*.
- Zhena, Z., Wang, Z., Wanga, F., Mia, Z., and Lia, K. (August 2017). Research on a cloud image forecasting approach for solar power forecasting. *Energy Procedia*.
- Zuckerman, C. (24 April 2019). Clouds, explained. *National Geographic*.
- Zuo, H.-M., Qiu, J., Jia, Y.-H., Wang, Q., and Li, F.-F. (March 2022). Ten-minute prediction of solar irradiance based on cloud detection and a long short-term memory (lstm) model. *Energy Reports*.

A.1 MINUTELY MAE AND QUANTILE 95% PER CLASS

Class A

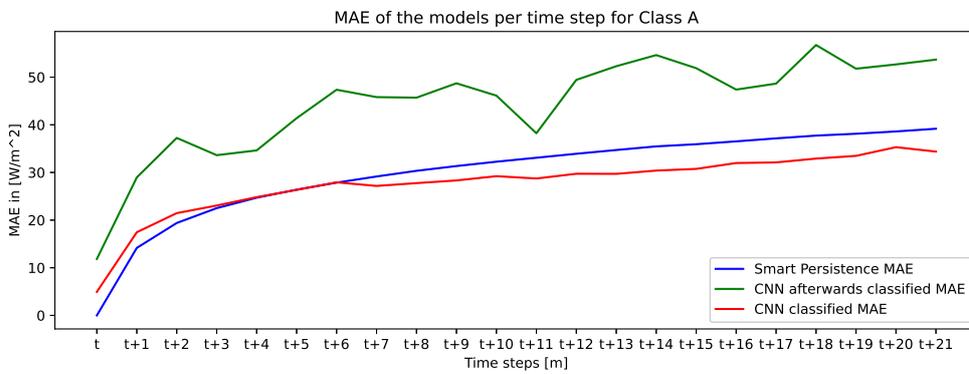


Figure A.1: The 21-minute MAE of the baseline model (SP) against the CNN model when the classification for Class A is applied beforehand against when the classification is not applied, so afterwards classified.

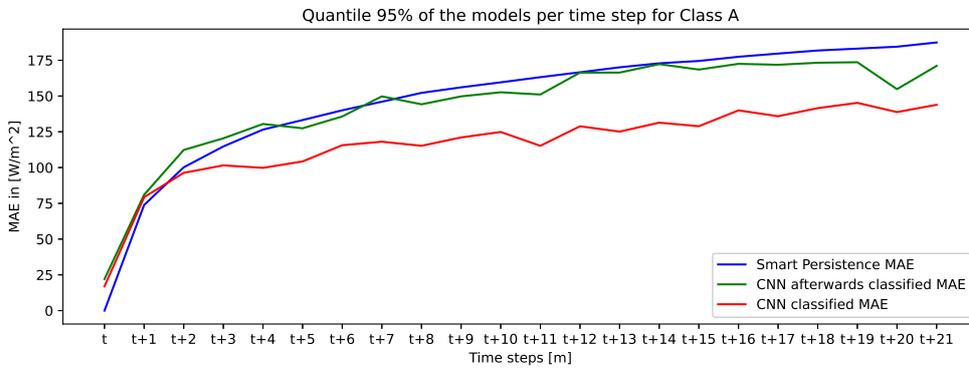


Figure A.2: The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification for Class A is applied beforehand against when the classification is not applied, so afterwards classified.

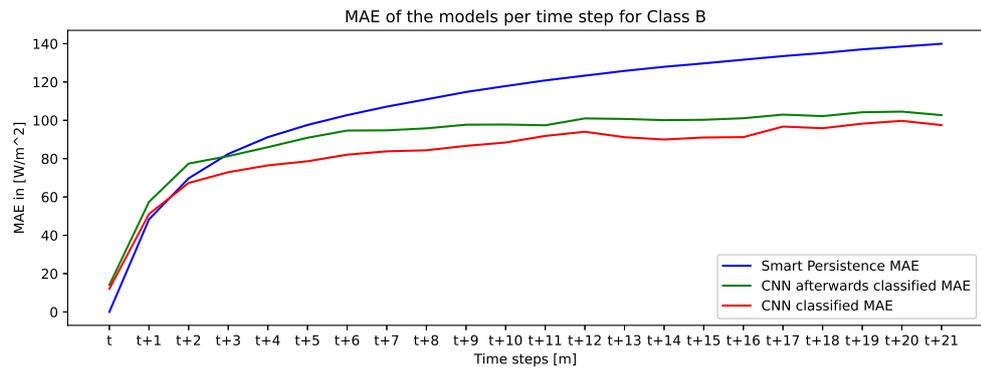
Class B

Figure A.3: The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class B is not applied, so afterwards classified.

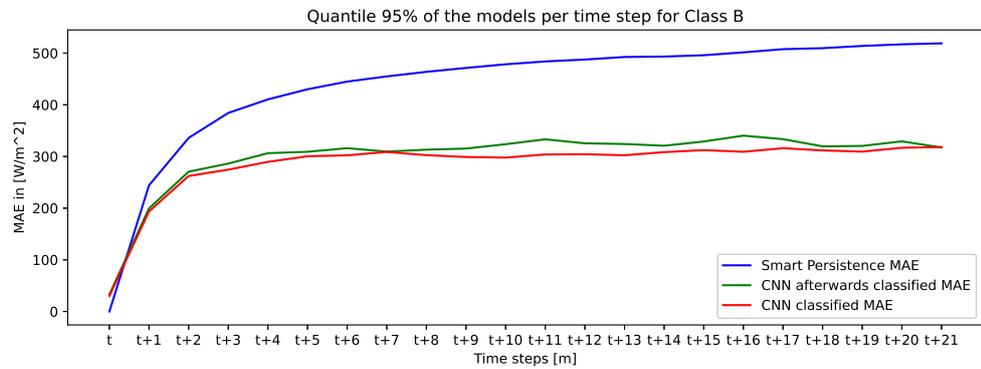


Figure A.4: The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class B is not applied, so afterwards classified.

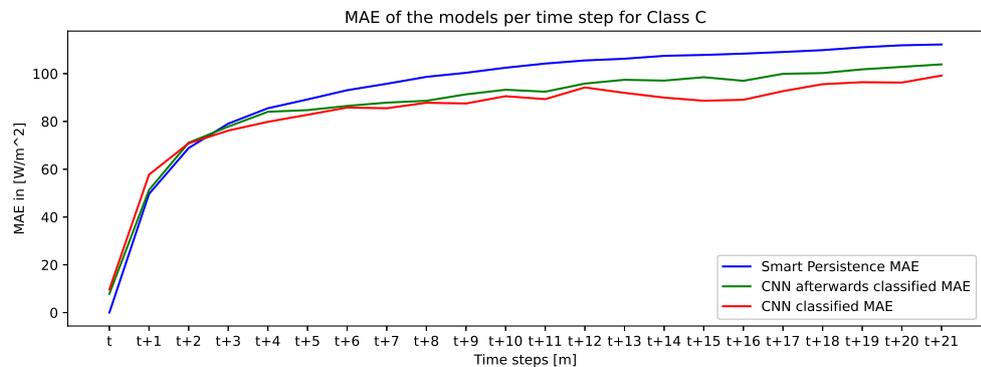
Class C

Figure A.5: The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class C is not applied, so afterwards classified.

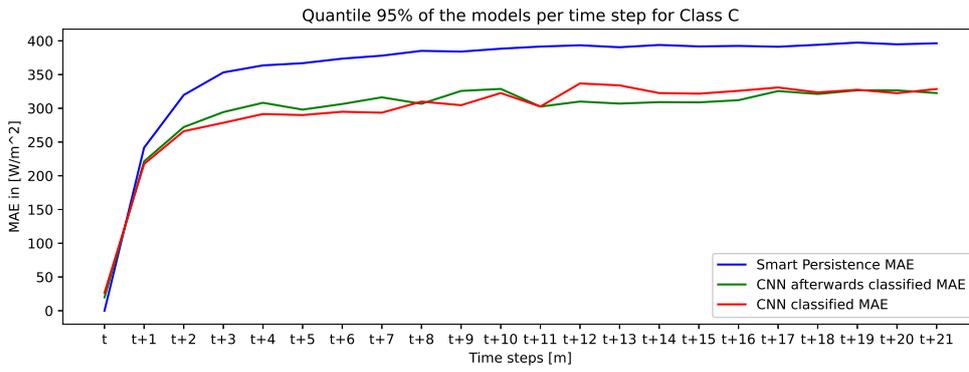


Figure A.6: The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class C is not applied, so afterwards classified.

Class D

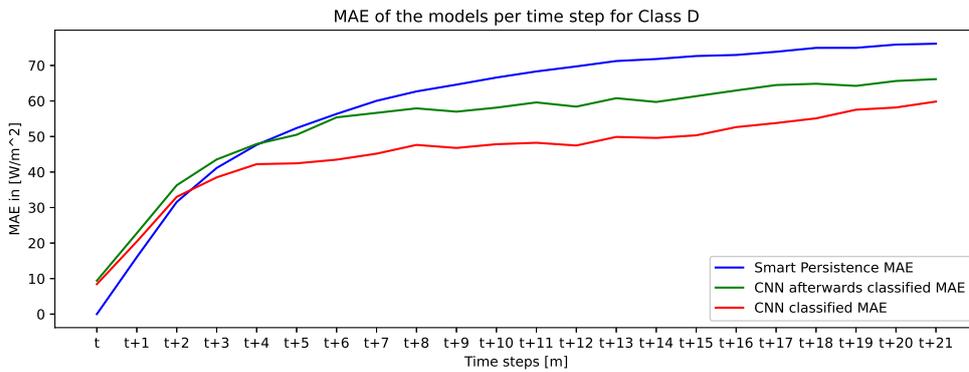


Figure A.7: The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class D is not applied, so afterwards classified.

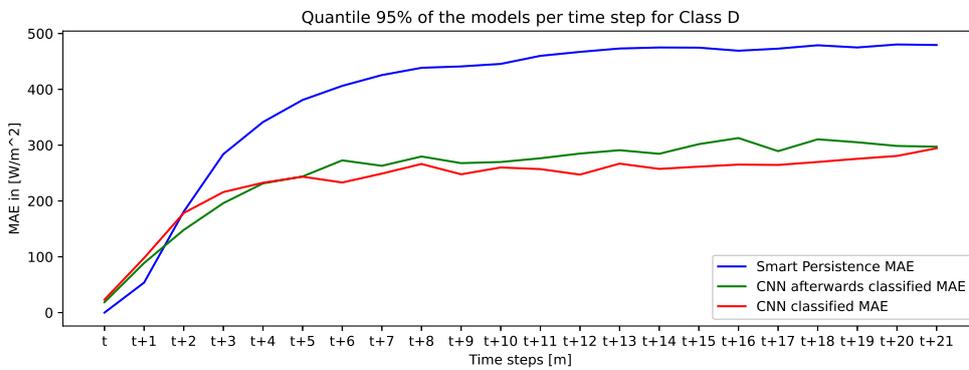


Figure A.8: The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class D is not applied, so afterwards classified.

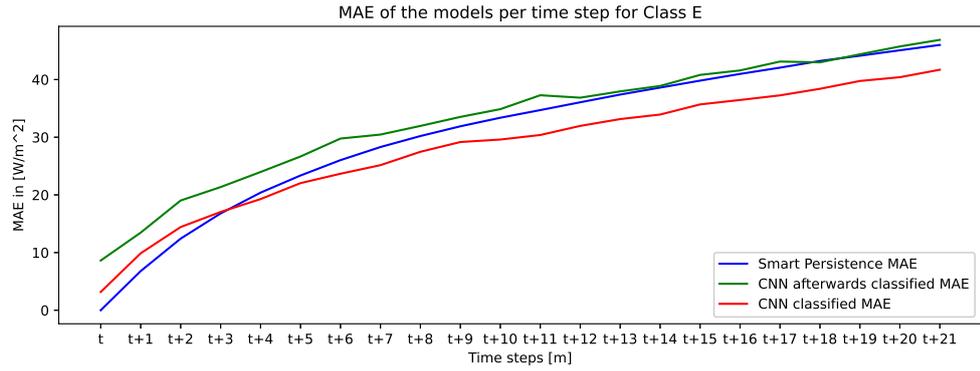
Class E

Figure A.9: The 21-minute MAE of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class E is not applied, so afterwards classified.

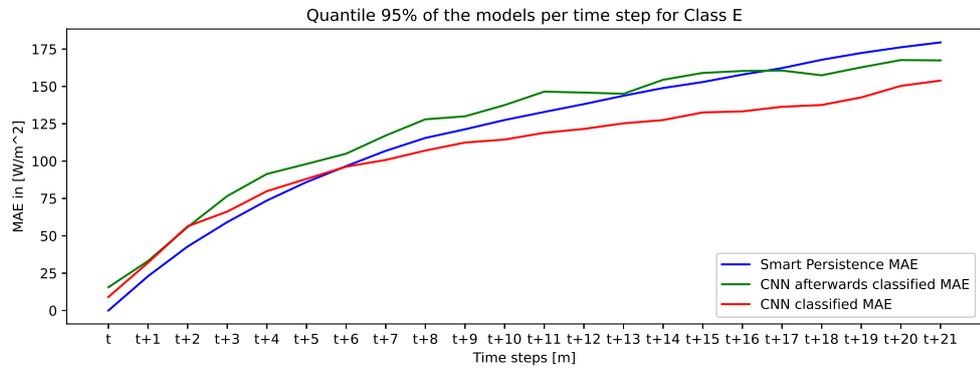


Figure A.10: The 21-minute quantile 95% of the baseline model (SP) against the CNN model when the classification is applied beforehand against when the classification for Class E is not applied, so afterwards classified.

COLOPHON

This document was typeset using \LaTeX . The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from André Miede.

