

POLITECNICO DI MILANO
School of Industrial and Information Engineering
Department of Energy
Master of Science in Electrical Engineering



Solar Irradiance Nowcasting

Supervisor: Prof. Sonia Leva
Co-Supervisor: Dr. Alessandro Niccolai

Candidate:
Seyedamir Orooji
Matr. 926382

Academic Year 2020-2021

To my beloved parents, the only real friends of mine, for always supporting and loving me. Without them this work was impossible.

Acknowledgement

With many thanks to Professor Sonia Leva and Dr. Alessandro Nicolai for their consistent support and guidance during this research work despite all the difficulties during Covid-19 pandemic.

Contents

Abstract	XIII
Sommario	XV
1 Introduction	1
1.1 Context and Motivations	1
1.2 Structure of the Thesis	2
2 Nowcasting	3
2.1 Forecasting and Different Techniques of Forecasting	3
2.2 Nowcasting Definition	4
2.3 All-Sky Imager	5
2.4 Cloud Tracking	6
2.5 Nowcasting Procedure	6
2.6 ASI Images	7
2.6.1 Cloud Tracking using a cluster of images	7
2.6.2 Determination of cloud transmittance for ASI	9
2.6.3 Cloud height and tracking accuracy in systems with multiple ASI	9
2.6.4 Validation of ASI-based nowcasting	10
3 Artificial Intelligence	12
3.1 Artificial Intelligence Introduction	12
3.2 Machine Learning	13
3.2.1 Supervised Learning	14
3.2.2 Unsupervised Learning	14
3.2.3 Semi-Supervised Learning	14
3.2.4 Reinforcement Learning	15

3.3	Deep Learning and ANN	15
3.3.1	CNN	17
3.3.2	RNN	18
3.3.3	GAN	19
3.4	Tensorflow	20
3.5	Keras	21
4	Irradiation Data Analysis	22
4.1	Data Analysis Starting Phase	22
4.2	Converting Time Scale	23
4.3	Ramp Identification	24
4.4	Ramps Number Calculation	31
5	Sky Images Analysis	36
5.1	Image Analysis Starting Phase	36
5.2	Preparation of Images	38
5.3	Editing the Images	38
5.4	Dataset Creation Introduction	42
5.5	Dataset Creation with three labels based on GHI	42
5.6	Dataset Creation with four labels based on GHI	49
5.7	Dataset Creation with another image processing method	56
5.8	Dataset Creation with hand and by the help of GHI	60
5.9	Last Classification	63
6	Deep Learning Model Application	65
6.1	Why Deep Learning?	65
6.2	How does ANN work?	66
6.3	Important Activation Functions	68
6.3.1	Sigmoid	71
6.3.2	Tanh	71
6.3.3	ReLU	71
6.3.4	SoftMax	71
6.4	Deep Learning Model:	72
6.4.1	Convolutional Layer	72
6.4.2	MaxPooling Layer	74
6.4.3	Flatten Layer	74
6.4.4	Fully Connected Layer	74
6.5	Training Deep Learning Model on the Dataset	75

6.6	Image Preprocessing	77
7	Forecasting	80
7.1	Dataset Preparation Procedure	80
7.2	Forecasting Model	83
7.3	Forecasting Results	89
8	Conclusion and Future Work	106
9	Appendix: Irradiation Data	108
10	Appendix: Deep Learning Model Application	125
11	Appendix: Deep Learning Model Application	128
	Bibliography	153

List of Abbreviations

ANN	<i>Artificial Neural Network</i>
ASI	<i>All-Sky Imager</i>
CBH	<i>Cloud Base Height</i>
CNN	<i>Convolutional Neural Network</i>
COT	<i>Cloud Optical Thickness</i>
CSI	<i>Clear Sky Index</i>
CSL	<i>Clear Sky Library</i>
CSV	<i>Comma Separated Value</i>
CTH	<i>Cloud Top Height</i>
DIF	<i>Diffuse Horizontal Irradiance</i>
DNI	<i>Direct Normal Irradiance</i>
GAN	<i>Generative Adversarial Networks</i>
GHI	<i>Global Horizontal Irradiation</i>
ML	<i>Machine Learning</i>
MLP	<i>Multi Layer Perceptron</i>
MMCR	<i>Millimeter Wave Cloud Radar</i>
NWP	<i>Numerical Weather Prediction</i>
PIL	<i>Python Imaging Library(Pillow)</i>
PV	<i>Photovoltaic</i>
PZA	<i>Pixel Zenith Angle</i>
RES	<i>Renewable Energy Sources</i>
RL	<i>Reinforcement Learning</i>
RMSE	<i>Root Mean Square Error</i>
RNN	<i>Recurrent Neural Network</i>
Skimage	<i>Scikit-Image</i>
SPA	<i>Sun Pixel Angle</i>

List of Figures

2.1	ASI Captures	5
2.2	Feature Points Detection	8
2.3	Cloud Detection Steps	11
3.1	ML Different Approaches	15
3.2	ANN Architecture	16
3.3	CNN Architecture	17
3.4	RNN Architecture	18
3.5	GAN Architecture	19
4.1	September 2019 CSV File	22
4.2	2019 Minutely GHI	23
4.3	2020 Minutely GHI	24
4.4	September 2019 Ramp-up and Ramp-down	24
4.5	September Ramps Minutely	25
4.6	October Ramps Minutely	26
4.7	September Ramps 5 Minutes Time Horizon	26
4.8	October Ramps 5 Minutes Time Horizon	27
4.9	September Ramps 10 Minutes Time Horizon	27
4.10	October Ramps 10 Minutes Time Horizon	28
4.11	September Ramps 15 Minutes Time Horizon	28
4.12	October Ramps 15 Minutes Time Horizon	29
4.13	September Ramps 20 Minutes Time Horizon	29
4.14	October Ramps 20 Minutes Time Horizon	30
4.15	September Ramps Number	31
4.16	October Ramps Number	32
4.17	November Ramps Number	32
4.18	December Ramps Number	33

4.19	January Ramps Number	33
4.20	February Ramps Number	34
4.21	March Ramps Number	34
4.22	April Ramps Number	35
5.1	Captured Image on September 18, 2019, at 11:25 A.M	37
5.2	Captured Image on October 17, 2019, at 3:36 P.M	37
5.3	Captured Image on September 18, 2019, at 11:25 A.M	38
5.4	CSV file for September 18, 2019	39
5.5	Code for cropping the images concerning the Sun location	40
5.6	The Main Image	41
5.7	The Cropped Image	41
5.8	Python Code for Classification based on GHI in the first method	43
5.9	First Classification Method labels	44
5.10	Image Labelled as Sunny in the First Classification Method	45
5.11	Image Labelled as Semi-Covered in the First Classification Method	45
5.12	Image Labelled as Cloudy in the First Classification Method	46
5.13	Image Incorrectly Labelled as Sunny in the first method	47
5.14	Image Incorrectly Labelled as Semi-Covered in the first method	48
5.15	Image Incorrectly Labelled as Cloudy in the first method	48
5.16	Python Code for Classification based on GHI in the second method	50
5.17	Image Labelled as Clear Sky in the Second Method	50
5.18	Image Labelled as Low-Covered in the Second Method	51
5.19	Image Labelled as High-Covered in the Second Method	51
5.20	Image Labelled as Rainy in the Second Method	52
5.21	Confusion Between Low-Covered or High-Covered Label	53
5.22	Confusion Between Low-Covered or High-Covered Label	54
5.23	Confusion Between Low-Covered or High-Covered Label	54
5.24	Captured Image on September 20, 2019, at 7:15 A.M	56
5.25	Edited Image	56
5.26	Python Code for removing Red Part	57
5.27	Python Code for removing Red Part	57
5.28	Sun Trajectory Obtained by ANN on September 29, 2019, at 8:29 AM	58

5.29	Sun Trajectory Obtained by ANN on September 29, 2019, at 10:59 AM	59
5.30	Sun Trajectory Obtained by ANN on September 29, 2019, at 13:23 AM	59
5.31	Image Labeled as Clear sky on September 19, 2019, at 5:41 PM	61
5.32	Image Labeled as Covered on March 31, 2020, at 6:46 PM	61
5.33	Image Labeled as Covered on April 1, 2020, at 4:07 PM	62
5.34	Image Labelled as Rainy on September 22, 2019, at 5:27 PM	62
5.35	Classification Road Map	63
5.36	".Mat" file of classification based on CSI value	63
6.1	Difference between ML and Deep Learning	65
6.2	ANN Architecture	66
6.3	ANN Functional Architecture	67
6.4	Sigmoid Activation Function	69
6.5	Tanh Activation Function	70
6.6	ReLU Activation Function	70
6.7	Deep Learning Model Layers	73
6.8	MaxPooling Layer Process	74
6.9	Code for Splitting the Dataset	75
6.10	The Main Image before Removing the White Square	77
6.11	The Main Image after Removing the White Square	77
6.12	Image PreProcessing	78
6.13	Confusion Matrix	78
7.1	5 Minutes Forecasting Dataset Preparation	81
7.2	10 Minutes Forecasting Dataset Preparation	82
7.3	First Confusion Matrix for 5' Forecasting.	83
7.4	Second Confusion Matrix for 5' Forecasting.	84
7.5	First Confusion Matrix for 10' Forecasting.	85
7.6	Second Confusion Matrix for 10' Forecasting.	86
7.7	First Confusion Matrix for 15' Forecasting.	87
7.8	Second Confusion Matrix for 15' Forecasting.	88
7.9	Model Accuracy in 10 different and independent trials for 5 Minutes Forecasting. The average forecasting accuracy for this time horizon is 62.7%.	89

7.10	Model Accuracy in 10 different and independent trials for 10 Minutes Forecasting. The average forecasting accuracy for this time horizon is 65.48%.	90
7.11	Model Accuracy in 10 different and independent trials for 15 Minutes Forecasting. The average forecasting accuracy for this time horizon is 58.9%.	90
7.12	Classified and Misclassified samples in 5' Forecasting for the best day	91
7.13	Classified and Misclassified samples in 5' Forecasting for the worst day	92
7.14	Classified and Misclassified samples in 10' Forecasting for the best day	93
7.15	Classified and Misclassified samples in 10' Forecasting for the worst day	94
7.16	Classified and Misclassified samples in 15' Forecasting for the best day	95
7.17	Classified and Misclassified samples in 15' Forecasting for the worst day	96
7.18	The Model Accuracy curve for 5' Forecasting	97
7.19	The Model Loss curve for 5' Forecasting	98
7.20	The Model Accuracy Curve for 10' Forecasting	98
7.21	The Model Loss Curve for 10' Forecasting	99
7.22	The Model Accuracy Curve for 15' Forecasting	99
7.23	The Model Loss Curve for 15' Forecasting	100
7.24	Sparsity Curve for 5 Minutes Forecasting: Red dots show the misclassified samples and Correctly classified samples are represented by blue linear line	101
7.25	Sparsity Curve for 10 Minutes Forecasting: Red dots show the misclassified samples and Correctly classified samples are represented by blue linear line	102
7.26	Sparsity Curve for 15 Minutes Forecasting: Red dots show the misclassified samples and Correctly classified samples are represented by blue linear line	102
7.27	Sorted Prediction Performance in Descending order for 5 Minutes Forecasting	103
7.28	Sorted Prediction Performance in Descending order for 10 Minutes Forecasting	104

7.29	Sorted Prediction Performance in Descending order for 15 Minutes Forecasting	104
9.1	November Ramps Minutely	108
9.2	December Ramps Minutely	109
9.3	January Ramps Minutely	109
9.4	February Ramps Minutely	110
9.5	March Ramps Minutely	110
9.6	April Ramps Minutely	111
9.7	November Ramps 5 Minutes Time Horizons	112
9.8	December Ramps 5 Minutes Time Horizons	113
9.9	January Ramps 5 Minutes Time Horizons	113
9.10	February Ramps 5 Minutes Time Horizons	114
9.11	March Ramps 5 Minutes Time Horizons	114
9.12	April Ramps 5 Minutes Time Horizons	115
9.13	November Ramps 10 Minutes Time Horizons	115
9.14	December Ramps 10 Minutes Time Horizons	116
9.15	January Ramps 10 Minutes Time Horizons	116
9.16	February Ramps 10 Minutes Time Horizons	117
9.17	March Ramps 10 Minutes Time Horizons	117
9.18	April Ramps 10 Minutes Time Horizons	118
9.19	November Ramps 15 Minutes Time Horizons	118
9.20	December Ramps 15 Minutes Time Horizons	119
9.21	January Ramps 15 Minutes Time Horizons	119
9.22	February Ramps 15 Minutes Time Horizons	120
9.23	March Ramps 15 Minutes Time Horizons	120
9.24	April Ramps 15 Minutes Time Horizons	121
9.25	November Ramps 20 Minutes Time Horizons	121
9.26	December Ramps 20 Minutes Time Horizons	122
9.27	January Ramps 20 Minutes Time Horizons	122
9.28	February Ramps 20 Minutes Time Horizons	123
9.29	March Ramps 20 Minutes Time Horizons	123
9.30	April Ramps 20 Minutes Time Horizons	124
10.1	Binary Step Activation Function	125
10.2	Linear Activation Function	126
10.3	Leaky ReLU	126
10.4	Exponential Linear Unit Activation Function	127

10.5 Swish Activation Function	127
11.1 Third Confusion Matrix for 5' Forecasting.	129
11.2 Fourth Confusion Matrix for 5' Forecasting.	130
11.3 Fifth Confusion Matrix for 5' Forecasting.	131
11.4 Sixth Confusion Matrix for 5' Forecasting.	132
11.5 Seventh Confusion Matrix for 5' Forecasting.	133
11.6 Eighth Confusion Matrix for 5' Forecasting.	134
11.7 Ninth Confusion Matrix for 5' Forecasting.	135
11.8 Tenth Confusion Matrix for 5' Forecasting.	136
11.9 Third Confusion Matrix for 10' Forecasting.	137
11.10Fourth Confusion Matrix for 10' Forecasting.	138
11.11Fifth Confusion Matrix for 10' Forecasting.	139
11.12Sixth Confusion Matrix for 10' Forecasting.	140
11.13Seventh Confusion Matrix for 10' Forecasting.	141
11.14Eighth Confusion Matrix for 10' Forecasting.	142
11.15Ninth Confusion Matrix for 10' Forecasting.	143
11.16Tenth Confusion Matrix for 10' Forecasting.	144
11.17Third Confusion Matrix for 15' Forecasting.	145
11.18Fourth Confusion Matrix for 15' Forecasting.	146
11.19Fifth Confusion Matrix for 15' Forecasting.	147
11.20Sixth Confusion Matrix for 15' Forecasting.	148
11.21Seventh Confusion Matrix for 15' Forecasting.	149
11.22Eighth Confusion Matrix for 15' Forecasting.	150
11.23Ninth Confusion Matrix for 15' Forecasting.	151
11.24Tenth Confusion Matrix for 15' Forecasting.	152

Abstract

Limited Energy Resources or Nonrenewable Energy Resources like Oil, Natural Gas, Coal, etc., are going to finish. Statistics show that world energy consumption in 2017 was near 18.1 Terra Watts and predicted 28 Terra Watts by 2050. Nevertheless, the solar resource energy is 23000 Terra Watts by only considering the area over lands over a year. These results show that the Sun's energy is around a thousand times bigger than our energy demand, and this is the reason for focusing on Solar Energy.

There are two main research topics regarding the improvement of Solar cell efficiency: Improving the Materials used to Fabricate solar cells and Maximize the benefit of the received energy on Earth. As we know, only near 30% of the Solar Energy reaches Earth after passing the atmosphere. So, it is crucial not to waste this received energy anymore.

In this research work, the focus is on the second topic to benefit maximally from the Solar Irradiance. Solar Irradiance reaching the earth consists of three main components:

- 1- Direct Irradiation (Beam Irradiance)
- 2- Diffuse Irradiation
- 3- Ground Reflected Irradiation

However, since the last two parameters are hard to model, the Global Horizontal Irradiation(GHI) is used for modeling. GHI depends on Cosine Corrected Direct Normal Irradiance and Diffuse Horizontal Irradiance. By tilting the panel always to face the irradiance directly, the produced energy by the panels is increased. Moreover, if the data of future irradiance and cloud position will be available, the process of tilting the panel can be done automatically. This research work focuses on predicting the GHI value and Clear Sky Index (CSI) value, which with combination of these values with the cloud position, the produced energy by solar cells can be maximized.

There are three main steps done in this study work. First, the GHI Ramp Identification and Computing GHI Ramps number are done. Second, multiple image processing steps are done to make the images well-prepared for Deep Learning method. Third, two different classifications are implemented to group the images first based on the feature of the images and the cloud type and depth and secondly based on the CSI value. Finally, the classified images are processed by the Deep Learning model for the forecasting part. In this study work, the CSI value is forecasted for 5', 10', and 15' time horizon.

Sommario

Risorse energetiche limitate o risorse energetiche non rinnovabili come petrolio, gas naturale, bobina, ecc., Stanno per finire. Le statistiche mostrano che il consumo mondiale di energia nel 2017 era vicino a 18,1 Terra Watt e prevedeva 28 Terra Watt entro il 2050.

Tuttavia, l'energia della risorsa solare è di 23000 Terra Watt considerando solo l'area sulle terre nell'arco di un anno. Questi risultati mostrano che l'energia del Sole è circa mille volte più grande della nostra domanda di energia, e questo è il motivo per cui ci siamo concentrati sull'energia solare.

Ci sono due principali argomenti di ricerca riguardanti il miglioramento dell'efficienza delle celle solari: migliorare i materiali utilizzati per fabbricare celle solari e massimizzare il beneficio dell'energia ricevuta sulla Terra.

Come sappiamo, solo quasi il 30% dell'energia solare raggiunge la Terra dopo aver superato l'atmosfera. Quindi, è fondamentale non sprecare più l'energia ricevuta.

In questo lavoro di ricerca, l'attenzione si concentra sul secondo argomento per trarre il massimo vantaggio dall'irraggiamento solare.

L'irraggiamento solare che raggiunge la terra è costituito da tre componenti principali:

- 1- Irradiazione diretta (Beam Irradiance)
- 2- Irradiazione diffusa
- 3- Irradiazione riflessa dal suolo

Tuttavia, poiché gli ultimi due parametri sono difficili da modellare, l'irradiazione orizzontale globale (GHI) è utilizzato per la modellazione. Il GHI dipende dall'irraggiamento diretto normale corretto al coseno e dall'irraggiamento orizzontale diffuso. Inclinando il pannello sempre per affrontare direttamente l'irraggiamento, si aumenta l'energia prodotta dai pannelli.

In oltre, se saranno disponibili i dati dell'irraggiamento futuro, il processo di inclinazione del pannello può essere eseguito automaticamente. Questo lavoro di ricerca si concentra sulla previsione del valore GHI e del valore Clear Sky Index (CSI), che aiuta a massimizzare l'energia solare ricevuta.

Ci sono tre fasi principali svolte in questo lavoro di studio. In primo luogo, vengono eseguite l'identificazione della rampa GHI e il numero di rampe di calcolo GHI. In secondo luogo, vengono eseguite più fasi di elaborazione delle immagini per rendere le immagini ben preparate per il metodo Deep Learning. In terzo luogo, vengono implementate due diverse classificazioni per raggruppare le immagini prima in base alla caratteristica delle immagini e al tipo e alla profondità della nuvola e in secondo luogo in base al valore CSI. Infine, le immagini classificate vengono elaborate dal modello Deep Learning per la parte previsionale. In questo lavoro di studio, il valore CSI è previsto per un orizzonte temporale di 5', 10' e 15'.

Chapter 1

Introduction

This chapter provides an overview of the thesis. In section 1.1, we explain the context and motivations which led us to conduct this research. Then, section 1.2 presents information about the structure of the thesis.

1.1 Context and Motivations

Forecasting Energy, especially Renewable Energy Sources(RES), is now one of the most popular topics in science community. [1].

Since Fossil Fuels' demand is increased, solar cells (PV) and wind turbines have become more popular concerning growth in their installation capacity [2].

The expected power output, which is the primary goal of forecasting, is unpredictable behavior in integrated electric systems. Concerning this problem which is still an unsolved problem, forecasting accuracy can constantly be improved [3]. By improving the forecasting accuracy of RES, the ordinary operating reduces. Also, it reduces the balance between energy production and demand in terms of cost [4]. PV Power Forecast can be implemented in different periods. Nowcasting, which forecasts PV power in the short term, becomes interesting to solve grid stability issues and optimize the microgrid's energy management systems. Identifying the sun position is the first step of Nowcasting implementation [5]. The next important step is the measurement of Global Horizontal Irradiation (GHI). GHI is the received horizontal short-

wave irradiation from the sky on a surface placed in the ground. GHI depends on DNI, Direct Normal Irradiance, and DIF, Diffuse Horizontal Irradiance. The importance of GHI identification is in identifying irradiance ramps and CSI value. Irradiance ramps are the sharp spikes or a sudden change in the irradiance. Nowcasting can also benefit from sky images captured by regular cameras installed close to the solar panel. However, the more challenging part of nowcasting is estimating cloud appearance and movement, weather conditions, etc. These challenges are solved by computer vision [6].

Several methods can perform nowcasting; the popular categories to do nowcasting are using cloud imagery and the machine learning models [7]. The focus of this context is on implementing Machine Learning, and Deep Learning models.

1.2 Structure of the Thesis

Chapter 2 provides information about Nowcasting definition and how nowcasting can be done. Also it introduces the devices needed and the main concepts and theories related to this topic.

In Chapter 3, an introduction to Artificial Intelligence and its different kinds are introduced. Also, the differences of these different methods and their advantages are represented.

Chapter 4 provides the basic concepts of Irradiation. The main focus of this chapter is to do Data Analysis on solar irradiation in different time horizons and ramp identification.

In chapter 5, the analysis on our images to make the image dataset is provided. In this chapter, we prepare the image dataset by doing Image Processing Method for Deep Learning Model.

Chapter 6 includes an introduction of Convolutional Neural Networks(CNN) and the model used for Deep Learning with the obtained results.

In chapter 7, the forecasting part for CSI value prediction in different time horizons are provided. This chapter includes multiple results of the forecasting analysis.

Chapter 2

Nowcasting

This chapter provides the basic theoretical frameworks and concepts about Nowcasting and the tools and procedures needed to achieve the final result.

2.1 Forecasting and Different Techniques of Forecasting

Forecasting means making predictions using past and present data and, most commonly, analyzing trends to predict the future. There are three different forecasting techniques in Solar cells depending on the forecasting horizon listed below:

- Nowcasting (forecasting 1' to 30'): Nowcasting in terms of solar cells is similar to meteorological nowcasting in both spatial and temporal scales. Nowcasting is used to forecast in short periods from one minute to thirty minutes ahead.
- Short-term Forecasting (30' to 4 Hours): This kind of forecasting is also valuable for forecasting grid operations for both grid operators and electric market operators [8].
- Long-term Forecasting (lengthy periods): Energy Producers can benefit from this approach for long-term contracts with financial utilities. This kind of forecasting is not so much popular among the previous approaches.

The main goal of solar power nowcasting, except forecasting in a short time horizon, is to gain 90% predictability [9].

2.2 Nowcasting Definition

In Nowcasting, the images taken by upward-facing cameras are used to predict future irradiances. The number of solar plants needed can be decreased by nowcasting because the goal is to find an optimized way to generate the wanted amount of power. Also, the negative ramp-rate can be fulfilled traditionally. In nowcasting, the optimized resolution of receiving data is around 5-10 minutes. Nowcasting can be obtained by techniques such as:

- Statistical Techniques: Implementing statistical methods on the present and past data.
- Satellite-based Techniques: Implementing forecasting based on satellite images.
- Ground-based Techniques: Implementing forecasting based on images taken by All-Sky Imager (ASI).

Most Numerical Weather Prediction (NWP) and satellite data are unsuitable because of their low spatial and temporal resolution. Also, in our case, to analyze the irradiance more accurately, a weather station near our plants should be installed which might be impossible, since the weather stations are sparse and few. Thus, finding another way is preferred. There are some ways to provide a service to obtain more accurate data from stations to be used for our PV plants, but they are expensive. The best way for nowcasting is to capture the local weather at the solar panel location but with high spatial and temporal resolution using a regular video camera that points towards the sky and is installed near the solar panel [10]. NWP models are used to forecast in time horizons up to several days ahead with temporal resolutions in hours which is needed to improve balance area and control performance and scheduling. Satellite-based systems are used to get more accurate forecasting for time horizons up to 8-9 hours. However, Concerning the spatial and temporal resolution, these systems are not suitable for intra-hour forecasting. Ground-Based ASI is used for the immediate future forecast with a high temporal and spatial resolution [11].

2.3 All-Sky Imager

ASI is the device used to take pictures from the sky and store them for further analysis. ASI is an automatic full-sky camera with 180 fields of view. It provides high-resolution images such as JPEGs and live-video stream. It can also store the temperature and humidity of the environment. An image of ASI captures can be seen in figure 2.1. The 3D positions and shapes of all visible clouds are determined via voxel carving. The voxel is an abbreviation for volume element, and voxel carving derives 3D objects from multiple camera perspectives. The cameras are synchronized using an NTP server, and each camera takes pictures from the sky every minute.

There are multiple factors determine the ASI-based nowcasting systems' accuracy. Cloud Height and Cloud tracking are the most critical factors. In this research work, the main focus is on Cloud Tracking, which will be discussed in section 2.4.



Figure 2.1: ASI Captures

2.4 Cloud Tracking

The most popular cloud tracking approaches are formed on Block Matching or optical flow algorithms. Also, radiative transfer models are helpful in order to analyze the radiative effects on clouds. Analyzing clouds leads to some attributes, which the most important attributes are: Height, Motion, Position, and Transmittance [12]. The radiative transfer model calculates electromagnetic radiation's transfer through a planetary atmosphere such as the Earth's.

The physical energy transfer phenomenon in the form of electromagnetic radiation is called Radiative Transfer. Absorption, scattering, and emission are the processes that affect the propagation of radiation.

Detecting clouds, identify their motion, and analyzing the radiative effect are obtained by these type of nowcasting systems.

2.5 Nowcasting Procedure

ASI cameras are indeed cheap, but the challenge here is that they only provide photos of the sky, and we need to find a relation between these images and the output power of the PV panels. This analysis is challenging because we need to analyze the clouds' model, speed, shapes, etc. Then predict these parameters in the future. So here there are two jobs to be done:

- 1- Analyze the movements of the cloud by image-processing techniques.
- 2- Using deep learning methods to predict the movements of clouds and obtain an algorithm or algorithms to find a relation between the sky appearance (like clouds, sun, clear sky, and ...) and the output power of the solar panels.

In nowcasting, Machine Learning (ML) methods or, in a specific way, deep learning methods are used. The architectures of deep-learning or estimation that can be used are:

- Multilayer-Perceptron (MLP): MLP is a class of feed-forward Artificial Neural Network (ANN). There are mainly three layers in MLP such as Input Layer, Hidden Layer, Output Layer [13].

- 2- Convolutional Neural Network (CNN): CNN is a class of Deep Neural Networks. CNN is most widely used to analyze images [14].
- 3- Long Short-Term Memory (LSTM): Long short-term memory (LSTM) is a field of Deep Learning which is also called Recurrent Neural Network (RNN). LSTM has feedback connections that are not similar to other feed-forward models. The application area of LSTM is not limited to Images. It can be applied to Videos and Speech [15].

MLP has the lowest Root Mean Squared Error (RMSE) with 7%, and LSTM has the largest RMSE with 21%. CNN has RMSE with an amount of 12%. Some steps should be taken for power computation to compute solar panels' power and predict future power values. The critical parameters in this project are taken from the sky-images. So, by performing image processing techniques on the taken images, the analysis's necessary data can be obtained. The predictions of future results are achievable by deep learning methods. The images' crucial parameters are cloud tracking, cloud movement, position, the shadow of the sun, the sun's position, the type of clouds, etc. The tasks which could be predicted by deep learning methods are:

- Future frames from historical image sequences.
- Future rainfall intensity.
- Forecasting solar irradiance and power.

2.6 ASI Images

2.6.1 Cloud Tracking using a cluster of images

Clustering is an unsupervised classification technique used to group similar data into a cluster. The data which can be used in clustering are Images, Videos, etc [16]. In nowcasting, clustering can be used to predict the coverage of sunlight by clouds [17]. There are two different types of events related to the coverage of the sun with clouds: (in fact, they are the same physical phenomenon analyzed from a different point of view):

- 1- Obscuration
- 2- Ramp Down

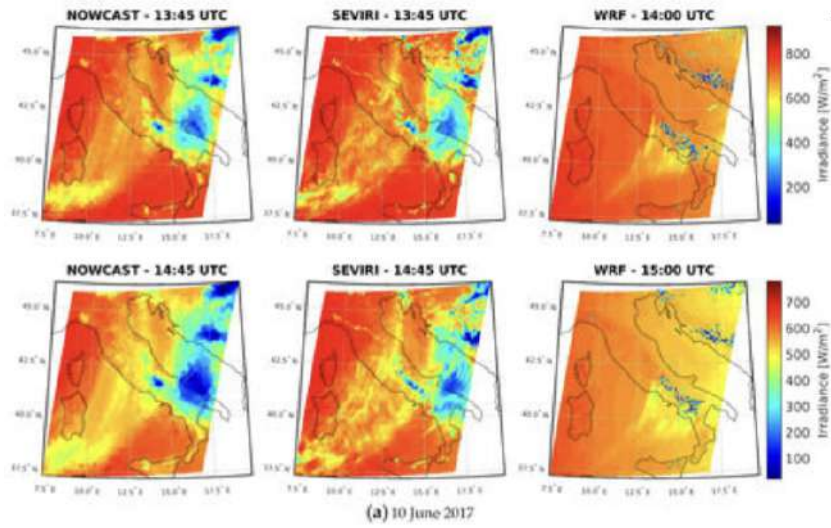


Figure 2.2: Feature Points Detection

Obscuration is called the condition in which the sun is covered partly or entirely by clouds.

Ramp-down is a situation when the irradiance drops dramatically within a short-time-interval.

Feature points should be used to adopt tracking in clustering. The Feature-Point Detection method aims to obtain image information at every image point and make decisions whether the given image includes the given type or not. The feature points are clustered in different groups using the proposed enhanced clustering algorithm. Figure 2.2 [18] makes the definition of Feature Points detection more familiar.

For Obscuration events prediction, the position of the sun must be located in all sky-images. A ramp-down event happens when the difference between current irradiance and the irradiance at the previous time is more significant than the threshold. The ramp-down event analysis is necessary to initialize alternative energy sources if the predicted solar power is not enough to be dispatched to the utilities.

2.6.2 Determination of cloud transmittance for ASI

The cloud transmittance is the ratio of ground-level horizontal irradiance and top-level irradiance [19].

There are two types of transmittance:

- Known Transmittance
- Unknown Transmittance

The angular distance of the relevant clouds from the Sun, which depends on cloud height and speed, is the distinguishing factor of being Known-transmittance or Unknown transmittance [20].

The approach selection for Cloud Transmittance determination is based on two facts:

- Vertical cloud profiles hold essential information about the types of clouds. It means that the cloud types can be identified by cloud height.
- The types of clouds can be associated with different optical properties. In other words, solar irradiance is attenuated in the atmosphere by absorption and scattering, and the attenuation itself caused by cloud is described by the cloud optical thickness (COT).

COT depends on particle size distribution, shape, water path, thermodynamic phase, and vertical extent [21]. A larger adequate particle size increases the absorbance. On the other hand, smaller particle size increases scattering [22]. Although there is no simple relation between cloud, height, type, and optical thickness, a probabilistic approach includes historical and current cloud height and transmittance measurements can be suitable.

2.6.3 Cloud height and tracking accuracy in systems with multiple ASI

This task is complex since the clouds have different layers, and they move in different directions simultaneously. Cloud heights are an essential factor in detecting the correct positions of the shadows on the ground. Moreover, an error in the detection of cloud height leads to an error in cloud size detection [23].

Measuring clouds height is usually done with Lidars and Ceilometers, which sample the sky only above the sensor. The distance between Cloud Base Height (CBH) and Cloud Top Height (CTH), equals to Cloud Boundaries, is measured by Lidars. However, concerning the laser beam’s attenuation, especially for clouds with high optical depth, these capabilities are limited. There is another option called Millimeter-Wave Cloud Radar (MMCR) which is capable of scanning the entire horizon by measuring the different properties of clouds like layer height and thickness [24].

Both Lidars and MMCR are so costly for nowcasting systems. Recently, coverage of cloud height and motion is achievable with satellites. Achieving a large field of view is one of these systems’ benefits. ASIs capture images from the sky simultaneously, and then by matching the segmented clouds, the cloud heights are achievable.

2.6.4 Validation of ASI-based nowcasting

Segmenting clouds is one of the most challenging parts of nowcasting. In nowcasting, CSL (Clear Sky Library) is used, a library that provides estimates of the surface solar irradiance under a clear sky [23] [25]. There is some vital information stored in CSL like the RGB values for every pixel of the Clear Sky, dependent on Sun Pixel Angle(SPA), Pixel Zenith Angle(PZA), air mass, and Linke Turbidity.

Modelling atmospheric absorption and scattering of solar irradiation is called Linke Turbidity [26]. It represents the optical thickness of the atmosphere due to the absorption of water vapor and the absorption and scattering of aerosol particles in comparison to a dry and clear atmosphere [27].

Each color channel has 80 layers in the CSL. A special jointed air mass and Linke turbidity bin belong to each layer. The RGB values of photographs taken under clear sky conditions are contained within a layer depends on two angles, SPA, and PZA. If a pixel of the image is clouded, the RGB values deviate from the CSL, and the cloud is detected with CSL color channel ratios. Based on the current air mass and Linke turbidity, 1 CSL layer per color channel is selected. From these CSL layers, ratios (depicted: red-to-blue ratio) and differences are calculated compared to the current image. Cloud detection is then performed by using multiple thresholds. The thresholds used for the segmentation depend on PSA, PZA, and the current weather situation

(e.g., overcast, clear sky). The current weather situation is determined considering irradiance and illuminance fluctuations, detected movements, and cloud coverages found in the previous segmentation [25].

In figure 2.3, the different steps in cloud detection by using CSL can be seen.

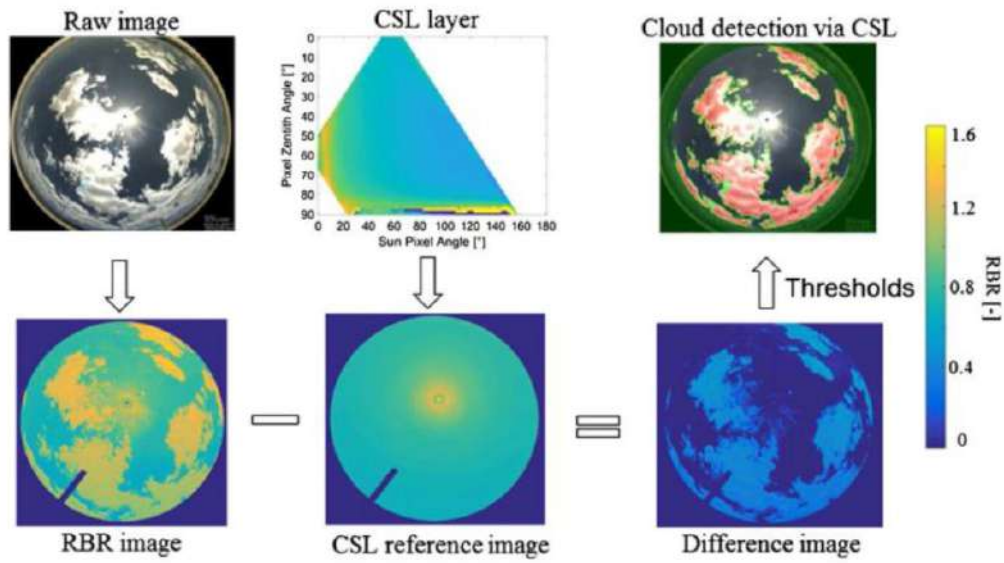


Figure 2.3: Cloud Detection Steps

Chapter 3

Artificial Intelligence

This chapter provides an introduction to Artificial Intelligence and its branches Machine Learning and Deep Learning.

3.1 Artificial Intelligence Introduction

The goal of Artificial Intelligence is to teach the computers the way that human thinks. It is related to simple computers we use; it can also teach robots how human thinks. AI teaches computers how to decide, how to work, how to think. The most important part is to teach them how to conclude the obtained results in the training process. The conclusion helps to solve the problems, predictions, etc. Machine Learning and Deep Learning are the main branches of AI. Deep Learning is the focus of this research work.

AI has been dominant in various fields such as [28]:

- Gaming: AI plays a vital role in running strategic games where the computer is taught to think about many possible ways to win the game. AI is used in games such as Chess, Poker, Backgammon, etc.
- Natural Language Processing: It is possible to use computers to understand human language to interact with humans.
- Expert Systems: Some applications that can interact with a human in the way of advising and reasoning. These applications integrate machine software for interaction.

- **Speech Recognition:** AI can be used in intelligent systems that can hear and comprehend sentences and catch the meanings while a human is talking to them. These systems have a database of all the accents and slang words to comprehend any single word. They are also capable of realizing a change in human voice caused by illness.
- **Handwriting Recognition:** The machines are taught to distinguish the letters and numbers to determine any handwriting.
- **Intelligent Robots:** Robots can perform the assignments given by a human. They have sensors to distinguish physical information from the real world, such as light, warmth, temperature, development, sound, bump, and weight. They have effective processors, numerous sensors, and colossal memory to display insights. Also, they are competent to learn from their mistakes and adjust to the modern environment.

3.2 Machine Learning

ML is the study of trainable algorithms. It means that these algorithms are capable of improving by training more and more [29].

ML algorithms build a model by training on the training set and then implement these trained algorithms on the test set to make predictions or decisions [30].

ML can work on both small datasets and large datasets by examining the data to find a typical pattern for the next steps.

Why Machine Learning?

The main reason that data scientists need Machine Learning is the importance of predicting the future. Future prediction can help make better decisions. Since this field is critical in science, it is continuously evolving to make prediction accuracy better.

Besides the importance of future prediction, ML helps to improve the way of analysis. It increases the speed of analysis with the automatic process. It is now a replacement to previous traditional statistical techniques [31].

There are four different approaches in ML:

- Supervised Learning
- Unsupervised Learning
- Semi-Supervised Learning
- Reinforcement Learning

3.2.1 Supervised Learning

The main characteristic of Supervised Learning, a branch of ML, is labeled data in the training set. Two sets of data are available for the process, Training Set and Test Set. The machine learns the concept of labels by being trained on the training set and applying these concepts to the test set where the data is unlabelled. So, the main goal of Supervised Learning is to identify the label in the test set with high accuracy. Identifying the fruits and sorting them into the appropriate categories is an excellent example of Supervised Learning. The model will choose from a variety of fruit photographs. The model begins practicing on some of these labeled pictures and tries to predict the correct category for unlabeled data in the test set [32].

3.2.2 Unsupervised Learning

The main difference between Unsupervised Learning and Supervised Learning is the presence of labeled data. In Unsupervised Learning, the model mainly works on unlabelled data. How to represent the input data to return the overall collection's statistical structure is Unsupervised Learning's goal. Unsupervised Learning is more similar to how the brain does the process, and concerning that, it is more critical than supervised learning [33].

3.2.3 Semi-Supervised Learning

Supervised Learning learns from the labels and predicts based on the labels. However, Unsupervised Learning should be able to predict the labels too. Semi-Supervised Learning aims to understand how to learn if both labeled data and unlabelled data exist in the dataset. Understanding the combination of labeled and unlabelled data and design algorithms to take advantage of this combination is Semi-Supervised Learning's goal. It can use unlabeled

data to improve supervised learning tasks when the labeled data or labeling them are expensive [34].

3.2.4 Reinforcement Learning

Reinforcement learning (RL) is a Machine Learning branch that explores how intelligent agents can behave in a given context to optimize the principle of cumulative reward [35]. An excellent example of RL is to train a robot to reach the destination only by walking through the white squares and ignoring the black squares. RL's labeled input or output is not a concern like Supervised or Unsupervised Learning [36].

Figure 3.1 [37] indicates the differences between the mentioned ML approaches.

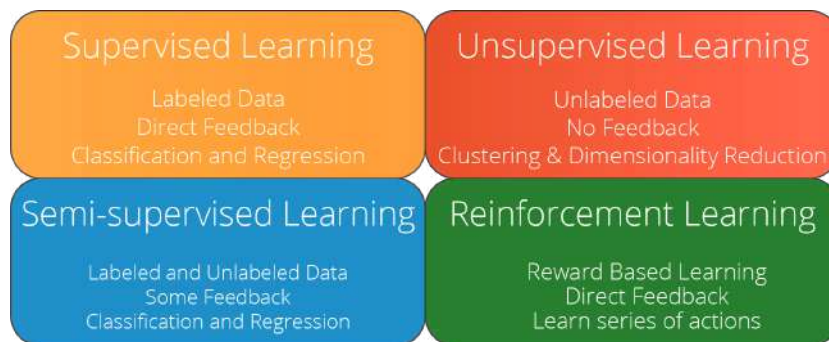


Figure 3.1: ML Different Approaches

3.3 Deep Learning and ANN

Deep Learning and ANN's strength is Speech Recognition, Handwriting Recognition, Transfer Learning, Gaming, etc. In these fields, Deep Learning and ANN have shown higher performance [38]. In this research work, the main focus is on the classification area of ANN.

In this research work, the Deep Learning performance of analyzing the visual data is used. Deep Learning can automatically learn all the representative parts of the images like the edges, the colors, the shapes, etc. By repeating these processes multiple times, the model can better understand the training process and learn these parts. By applying this learned model on the

test set, the model does the classification and predicts based on the learned parameters [39].

An Artificial Neural Network (ANN) is a model that attempts to replicate biological neural networks' structure and functionality. An artificial neuron, or a simple mathematical model, is the fundamental building block of any ANN (function). Multiplication, summation, and activation are three basic principles in this model. At the artificial neuron entrance, the inputs are weighted, meaning that every input value multiplies with individual weight. Artificial Neuron's middle section is a sum function that sums all weighted inputs and bias. At the artificial neuron output, the sum of previously weighted inputs and bias passes through the activation function called the transfer function [40].

In figure 3.2, the architecture of ANN can be seen [41].

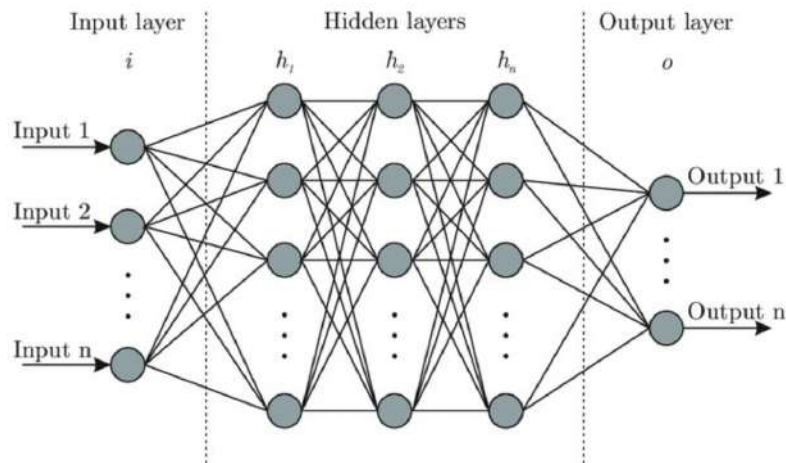


Figure 3.2: ANN Architecture

The main architectures of deep learning are:

- Convolutional Neural Networks (CNN)
- Recurrent Neural Network (RNN)
- Generative Adversarial Networks (GAN)

3.3.1 CNN

Convolutional Neural Networks (CNNs) are similar to standard artificial neural networks (ANNs) in that they are made up of neurons that learn to optimize themselves. Each neuron will still receive input and operate (such as a scalar product followed by a non-linear function) - the basis of countless ANNs. From the input raw image vectors to the final output of the class score, the entire network will still express a single perceptive score function (the weight). The last layer will contain loss functions associated with the classes, and all of the regular tips and tricks developed for traditional ANNs still apply. The only significant distinction between CNNs and standard ANNs is that CNNs are mainly used for image pattern recognition. It allows us to encode image-specific features into the architecture, making the network more suited for image-focused tasks - while further reducing the parameters required to set up the model. Traditional types of ANN struggle with the computing complexities needed to compute image images, which is one of their most significant shortcomings [42].

Figure 3.3 shows the architecture of a CNN model [43]:

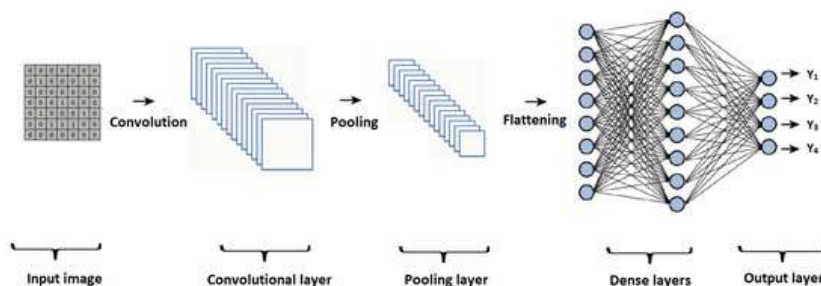


Figure 3.3: CNN Architecture

CNN is the focus of this research work. Details of this architecture will be discussed in chapter 6.

3.3.2 RNN

A recurrent neural network (RNN) is a type of artificial neural network in which nodes are connected in a directed graph that follows a temporal series. It can show temporal hierarchical behavior as a result of this. RNNs derived from feedforward neural networks can process variable-length sequences of inputs using their internal state (memory) [44] [45]. It makes them applicable to tasks such as unsegmented, connected handwriting recognition [46] or speech recognition [47] [48].

”Recurrent Neural Network” is roughly applied to two types of networks that have a standard general structure: finite impulse and infinite impulse networks. The behavior in both types of networks is temporally complex [49]. A directed acyclic graph replaced with a strict feed-forward neural network is known as a finite impulse recurrent network. An infinite impulse recurrent network, on the other hand, is a directed cyclic graph that cannot be unrolled.

Additional stored states are possible in both finite and infinite impulse recurrent networks, and neural networks can control the storage. If the network or graph contains time delays or has feedback loops, it may also overwrite the storage. Gated state or gated memory refers to these regulated states used in Long short-term memory networks (LSTMs) and gated recurrent units. It is also called Feedback Neural Network (FNN).

Figure 3.4 shows the architecture of the RNN model.

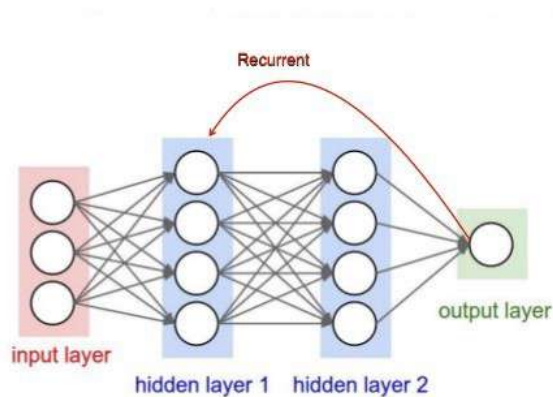


Figure 3.4: RNN Architecture

3.3.3 GAN

GANs, or Generative Adversarial Networks, is a form of generative modeling that employs deep learning techniques such as convolutional neural networks.

Generative modeling is an unsupervised learning task that involves automatically detecting and learning regularities or patterns in input data in machine learning. The model can generate or output new examples that plausibly could have been drawn from the original dataset.

GANs are an intelligent way of teaching an innovative algorithm by posing the problem as a supervised learning problem with two sub-models: the generator model, which we train to produce new examples, and the discoverer model, which attempts to distinguish examples as true (from the domain) or false (not from the domain). Both models are trained in an adversarial zero-sum game before the discriminator model is tricked around half of the time, indicating that the generator model produces plausible instances.

GANs are an exciting and rapidly changing field, delivering on the promise of innovative models in their ability to generate realistic examples across a range of problem domains, most notably in image-to-image translation tasks such as translating photos of summer to winter or a day. Also, generating natural photos of objects, scenes, and people that even humans cannot tell is fake [50].

In figure 3.5 [51], the architecture of GAN is indicated:

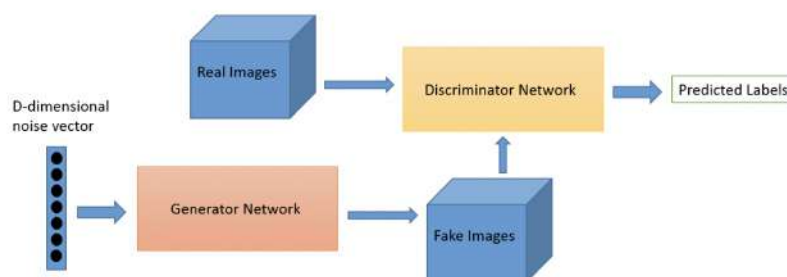


Figure 3.5: GAN Architecture

This research work's primary focus is on CNN, and the models are implemented by using Tensorflow and Keras libraries in python. In the following sections, simple introductions to these libraries can be found.

3.4 Tensorflow

TensorFlow is a Google-developed software library or platform for implementing machine learning and deep learning principles in the most simple way possible. It combines the computational algebra of optimization techniques for the easy calculation of many mathematical expressions.

TensorFlow has many machine learning libraries and is well-documented. It provides a few main functionalities and methods for doing so.

TensorFlow is often referred to as a "Google" product. A wide range of machine learning and deep learning algorithms are used. TensorFlow may train and run deep neural networks for handwritten digit classification, pattern recognition, word embedding, and the construction of various sequence models.

By taking multi-dimensional array inputs called Tensor, TensorFlow allows creating dataflow graphs and structures to describe how data flows through a graph. It enables the development of a flowchart of operations that can be done on these inputs and outputs. [52].

The crucial features of Tensorflow are:

- It includes a feature that defines, optimizes, and calculates mathematical expressions quickly with multi-dimensional arrays called tensors.
- It includes programming support of deep neural networks and machine learning techniques.
- It has a highly scalable computing capability that works for a variety of data sets.
- TensorFlow uses GPU computing, automating management. It also includes a unique feature of optimization of the same memory and the data used.

3.5 Keras

Keras is a high-level neural network API. It can be used with Tensorflow, Theano, or CNTK. It enables fast experimentation through a high-level, user-friendly, modular, and extensible API. Keras can operate on both CPU and GPU. Keras was developed and maintained by Francois Chollet and is part of the Tensorflow core, making it Tensorflow's preferred high-level API [53].

Keras leverages various optimization techniques to make high-level neural network API more accessible and more effective. It consists of the following features:

- Consistent, simple, and extensible API
- Minimal structure - easy to achieve the result without any frills
- It supports multiple platforms and backends
- It is able to perform on both CPU and GPU
- Highly scalability of computation

Keras is a compelling and dynamic framework and comes up with the following advantages:

- Larger community support
- Easy to test
- Keras is compatible with Python which makes things simpler
- Keras supports both convolution and recurrent networks
- Since deep learning models are discrete, they can be combined in a variety of ways.

Chapter 4

Irradiation Data Analysis

In this chapter, the focus is on obtained data from implementing image processing on ASI images including Irradiance. As known, the main part of calculating the power of the solar cells is Irradiation. The Irradiation data is available for eight consecutive months from September 2019 to April 2020.

4.1 Data Analysis Starting Phase

The first phase of Data Analysis is to convert the datasets into a readable format for the software. Python is used for data analysis in this research work. Some CSV files contain all the data are available for this work. The first rows of one of these CSV files are shown below.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Time	UR	T_AIR	ILL	V_AIR_DIR	G_DIFF	G_ORIZ_OIG_ORIZ	G_INCL	V_AIR	RAIN	V ALIM	
2	9/1/2019 0:00	63.8	22.46	-999999	17.4	0	0	0	0	0	0	13.7
3	9/1/2019 0:00	63.8	22.46	-999999	65.4	0	0	0	0	0	0	13.7
4	9/1/2019 0:00	63.8	22.46	-999999	64.2	0	0	0	0	0	0	13.7
5	9/1/2019 0:00	63.8	22.46	-999999	66.5	0	0	0	0	0	0	13.7
6	9/1/2019 0:00	63.8	22.46	-999999	62	0	0	0	0	0	0	13.7
7	9/1/2019 0:00	63.8	22.46	-999999	56.7	0	0	0	0	0	0	13.7
8	9/1/2019 0:01	64.2	22.42	-999999	54	0	0	0	0	0	0	13.7
9	9/1/2019 0:01	64.2	22.42	-999999	42.8	0	0	0	0	0	0	13.7
10	9/1/2019 0:01	64.2	22.42	-999999	42.8	0	0	0	0	0	0	13.7

Figure 4.1: September 2019 CSV File

In the beginning, all the CSV files should be converted to Pandas Data Frame. By Pandas Data Frame, full access to all the rows and columns for data analysis is granted [54]. In Figure 4-1, the column "GORIZ," which contains the data related to GHI, is critical in this research work.

4.2 Converting Time Scale

This dataset's time scale is 10 seconds. However, the images taken by the camera are in Minutely Scale. So, the Time Scale of the dataset should be changed from 10-Seconds to Minutely scale. This process can be done by selecting only the first row of each minute. In Figures 4-2 and 4-3, all the Minutely GHI for each month can be seen. As can be seen in Figures 4-2 and 4-3, the maximum values of GHI are in September 2019 and April 2020. Also, the GHI in December 2019 and January 2020 are lower among the others.

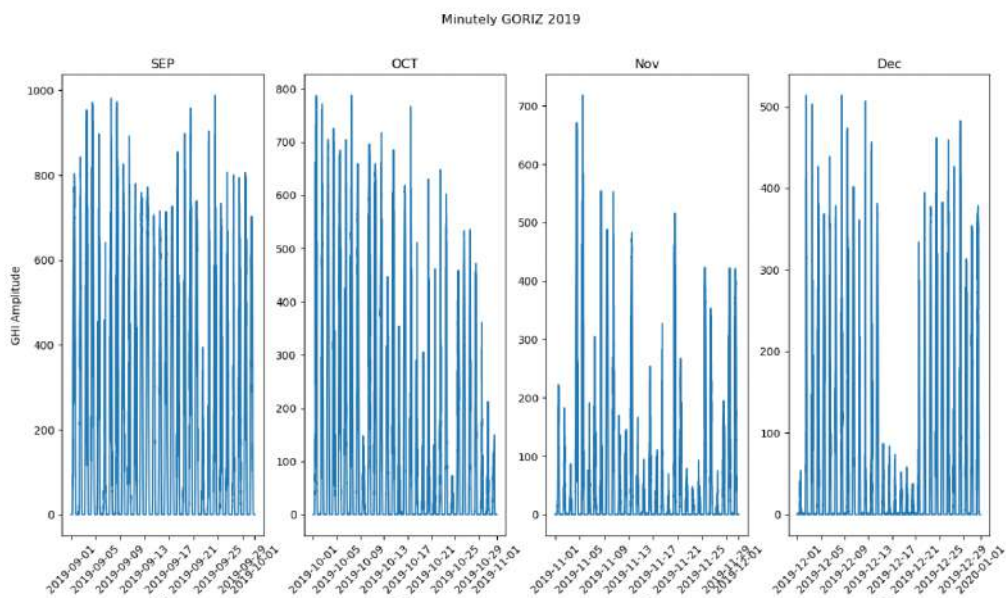


Figure 4.2: 2019 Minutely GHI

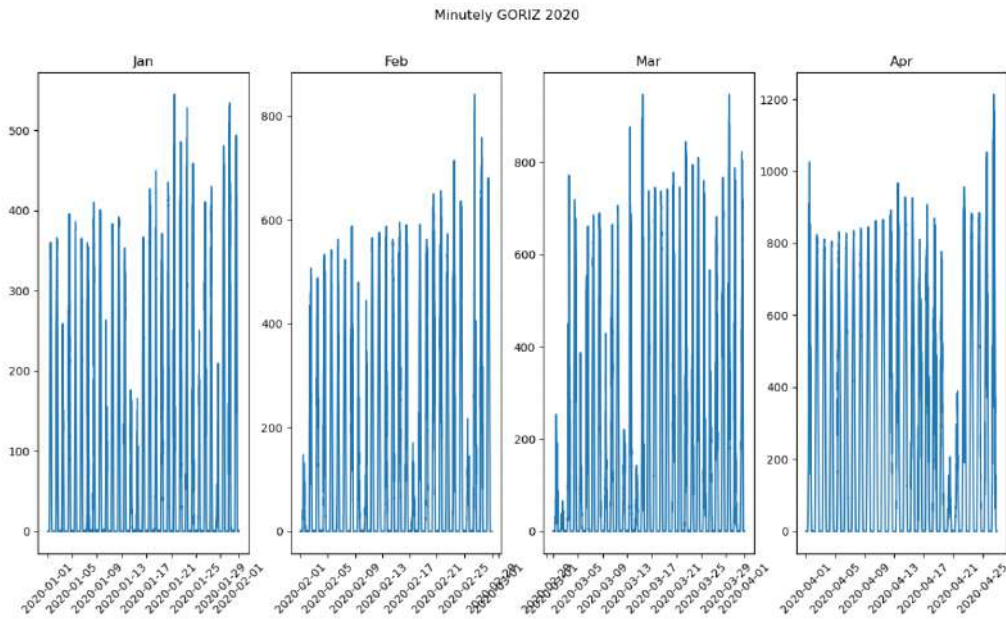


Figure 4.3: 2020 Minutely GHI

4.3 Ramp Identification

One of the essential points in Nowcasting is identifying the ramps, As mentioned in the first chapter. Both ramp-down and ramp-up are crucial. To better understand the ramps' meaning, and Figure 4-4 shows the ramp from the dataset of September 2019.

3507	9/1/2019 9:44	49.9	27.17	44.67	10.8	526	535	581	↓	696	0.3	0	13.7
3508	9/1/2019 9:44	49.9	27.17	44.67	18.1	526	537	283	↓	697	0.74	0	13.7
3509	9/1/2019 9:44	49.9	27.17	44.67	40	527	540	278	↓	698	0.74	0	13.7
3510	9/1/2019 9:44	49.9	27.17	44.67	41.6	528	541	271	↓	700	0.68	0	13.7
3511	9/1/2019 9:44	49.9	27.17	44.67	21.4	529	542	245	↓	701	0.63	0	13.7
3512	9/1/2019 9:45	50.2	27.12	45.09	85.1	529	540	261	↓	702	0.36	0	13.7
3513	9/1/2019 9:45	50.2	27.12	45.09	45.1	530	543	229	↓	702	0.36	0	13.7
3514	9/1/2019 9:45	50.2	27.12	45.09	334.9	530	542	449	↓	703	0.52	0	13.7

Figure 4.4: September 2019 Ramp-up and Ramp-down

As shown in Figure 4-4, the GHI value decreased suddenly from 581 to 283, called "Ramp-down." Also, there is a sudden increase in GHI value from 229 to 449 which is called "Ramp-up." The goal is to analyze the data to identify all the ramps that exist in the datasets. First, some values should be defined as a threshold. In this research work, if sudden changes are more

than 300 W/m², it will be defined as ramp-up and lower than -300 W/m² as ramp-down. The result of identifying the Minutely Scale's ramps can be seen in Figures 4-5 to 4-6 for the months September and October 2019. The Minutely Scales ramps for other months from November 2019 to April 2020 can be seen in Appendix A. To investigate the ramps, this process is repeated for different time scales such as 5 Minutes, 10 Minutes, 15 Minutes, 20 Minutes. In figures 4.7 to 4.14 the ramps identification results for months September and October can be seen for time horizons 5 Minutes to 20 Minutes respectively. The figures for the other months for all the mentioned time horizons are represented in Appendix A.

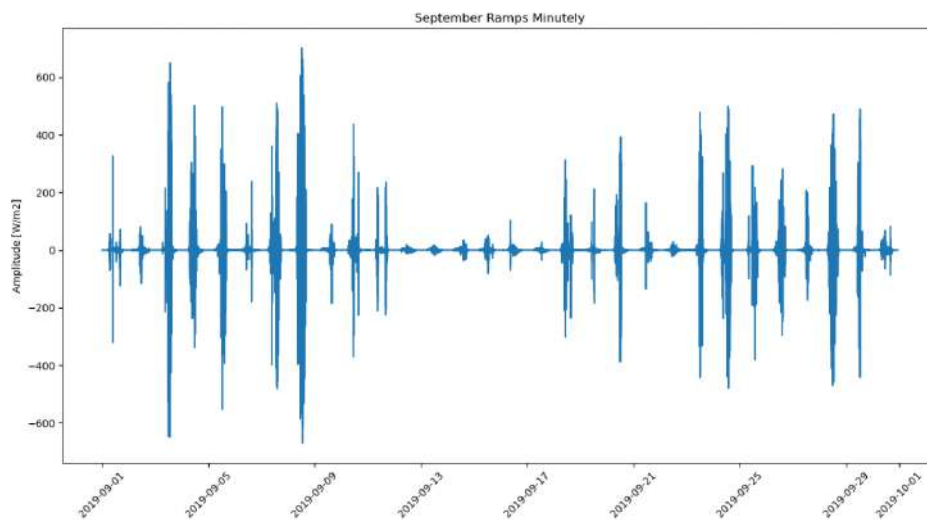


Figure 4.5: September Ramps Minutely

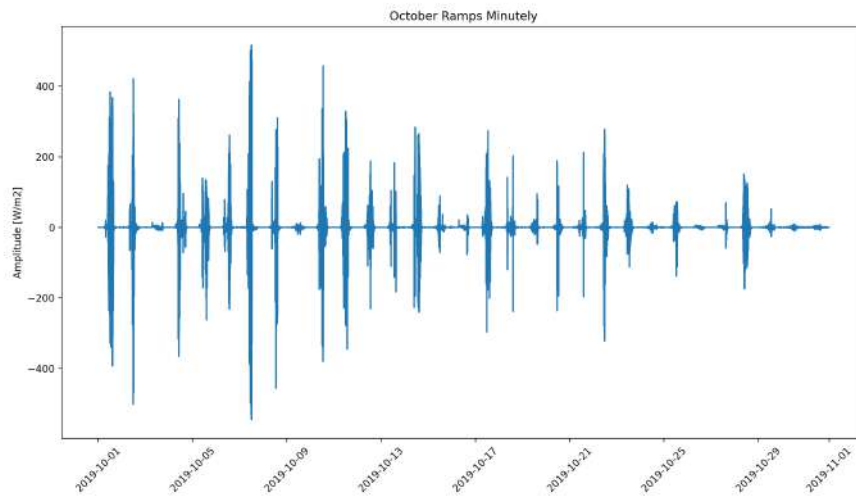


Figure 4.6: October Ramps Minutely

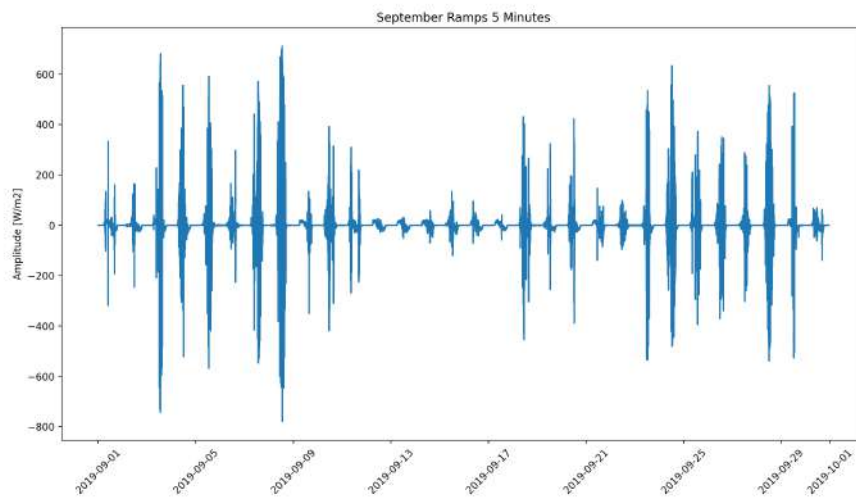


Figure 4.7: September Ramps 5 Minutes Time Horizon

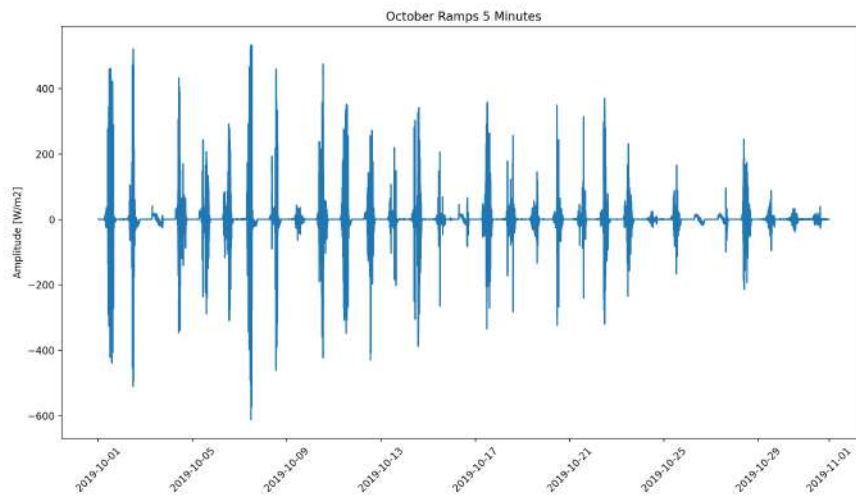


Figure 4.8: October Ramps 5 Minutes Time Horizon

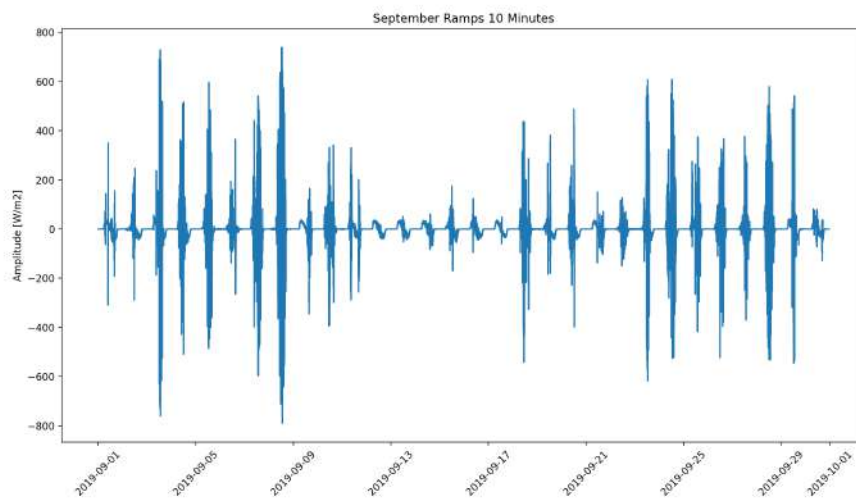


Figure 4.9: September Ramps 10 Minutes Time Horizon

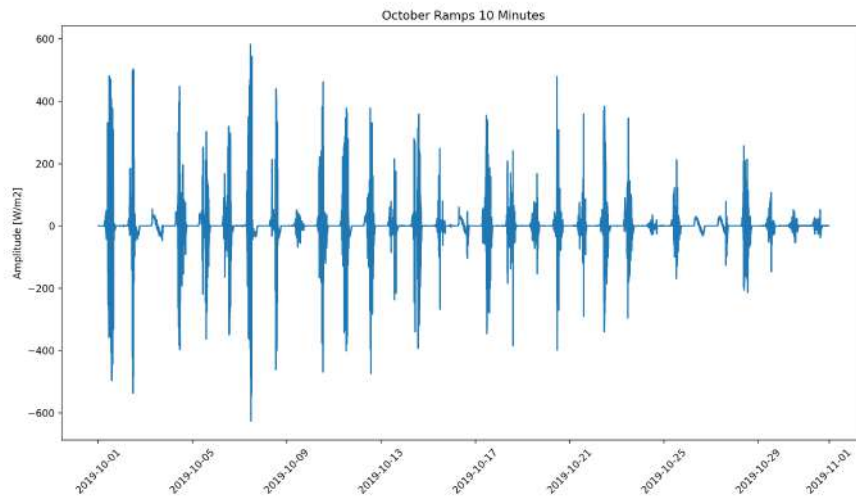


Figure 4.10: October Ramps 10 Minutes Time Horizon

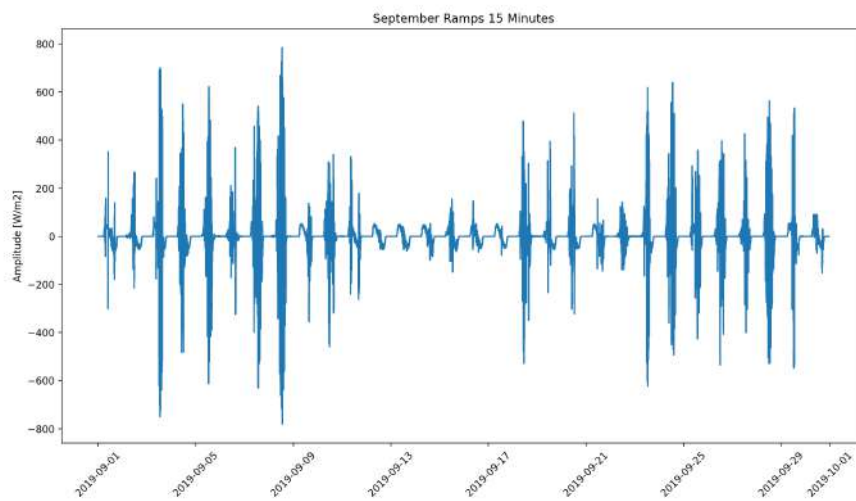


Figure 4.11: September Ramps 15 Minutes Time Horizon

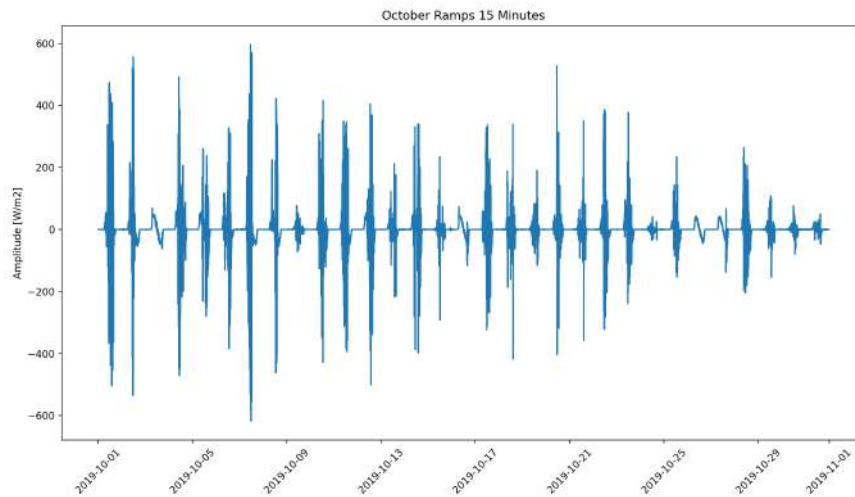


Figure 4.12: October Ramps 15 Minutes Time Horizon

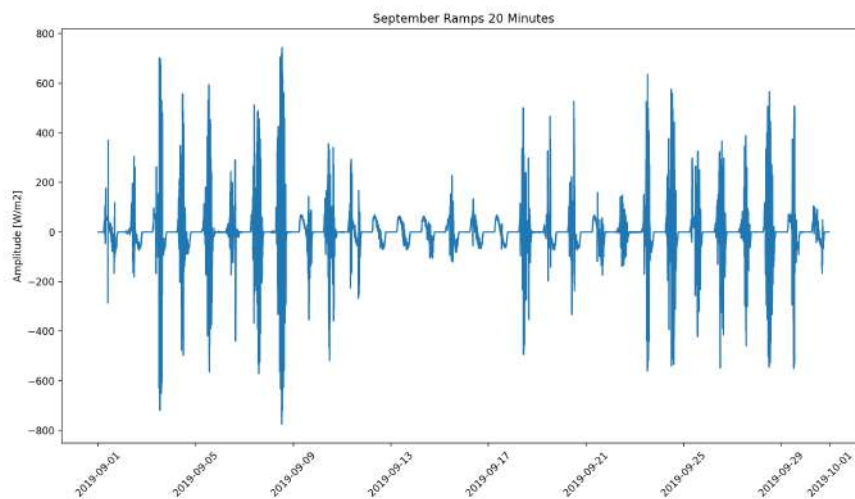


Figure 4.13: September Ramps 20 Minutes Time Horizon

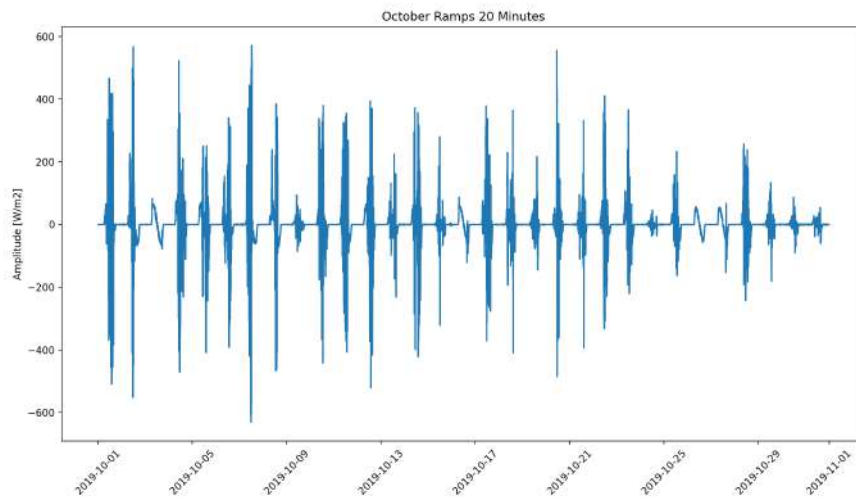


Figure 4.14: October Ramps 20 Minutes Time Horizon

4.4 Ramps Number Calculation

In this section, the number of ramps at different time scales is calculated. Since the goal is to compare the number of ramps at different time scales, the best method to show them is Histogram. In figures 4.15 to 4.22, the results of this part for both ramp-down and ramp-up can be seen. The number of ramps is shown for all the time scales from Minutely to 20 Minutes Scale. As shown in these graphs, the number of ramps usually is decreased by decreasing the time scale. In other words, by moving from the 20 Minutes Scale to Minutely Scale, there is a decrease in the number of ramps. Also, as mentioned before, the number of ramps in months September and April are the largest among the other months.

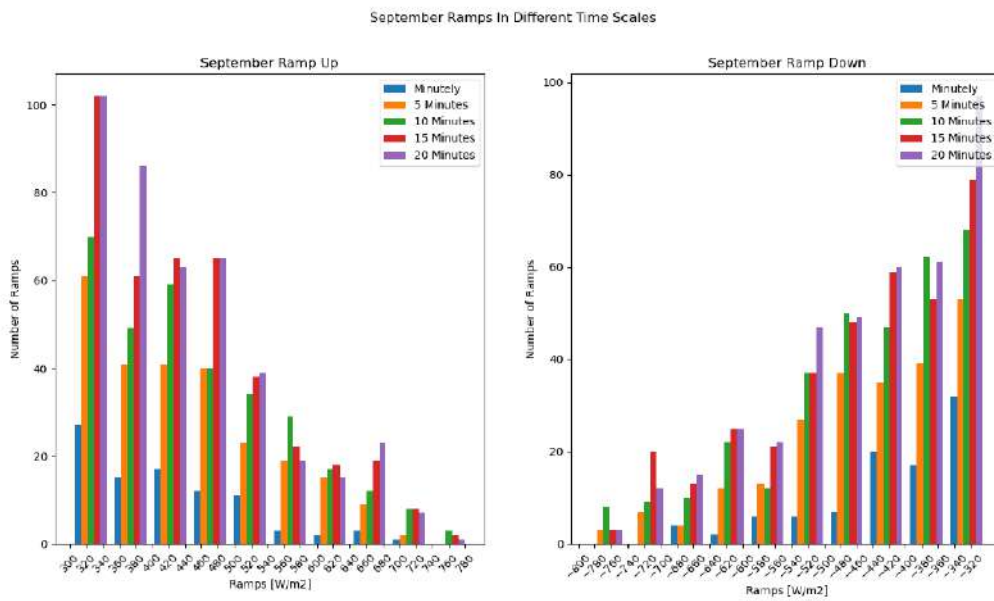


Figure 4.15: September Ramps Number

October Ramps In Different Time Scales

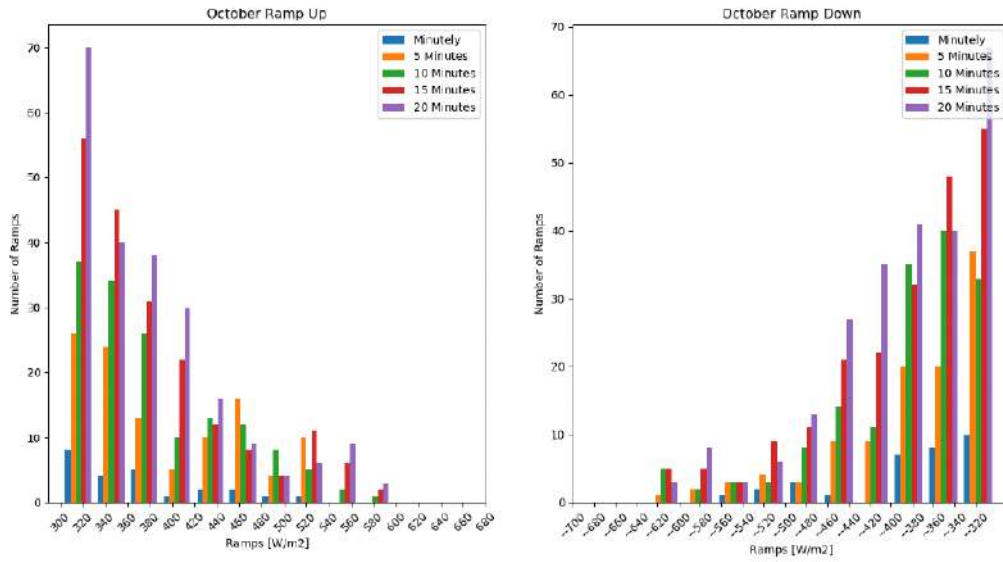


Figure 4.16: October Ramps Number

November Ramps In Different Time Scales

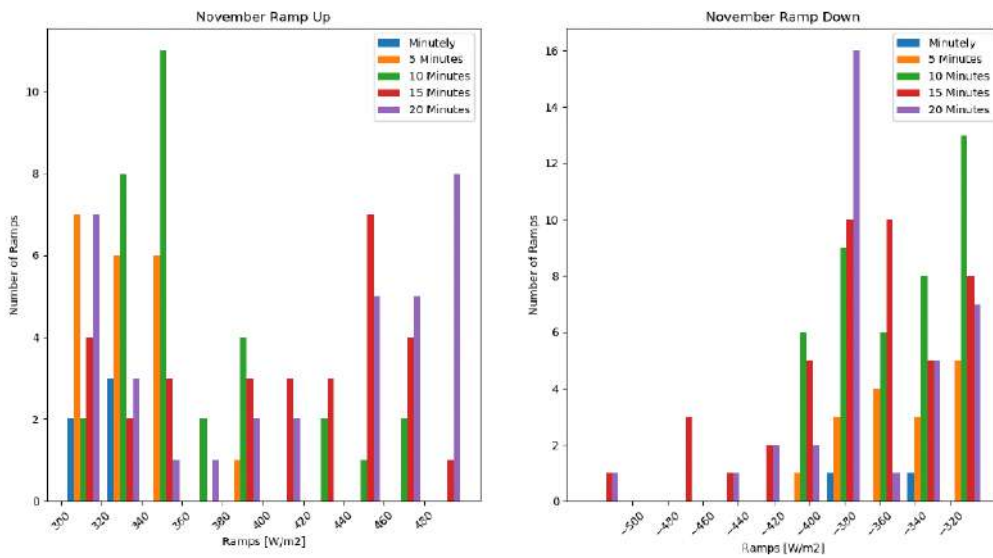


Figure 4.17: November Ramps Number

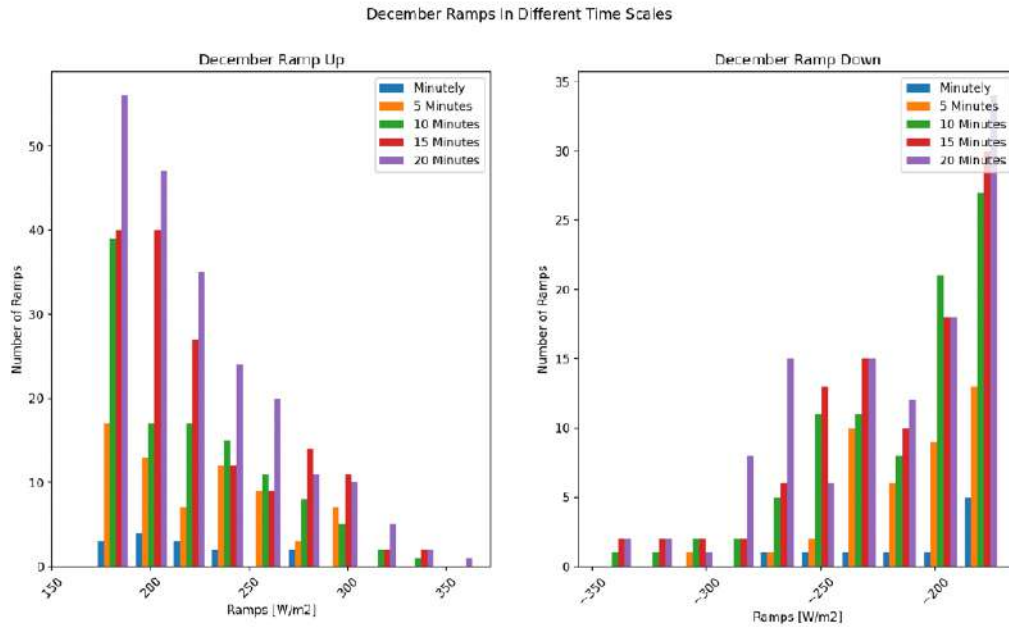


Figure 4.18: December Ramps Number

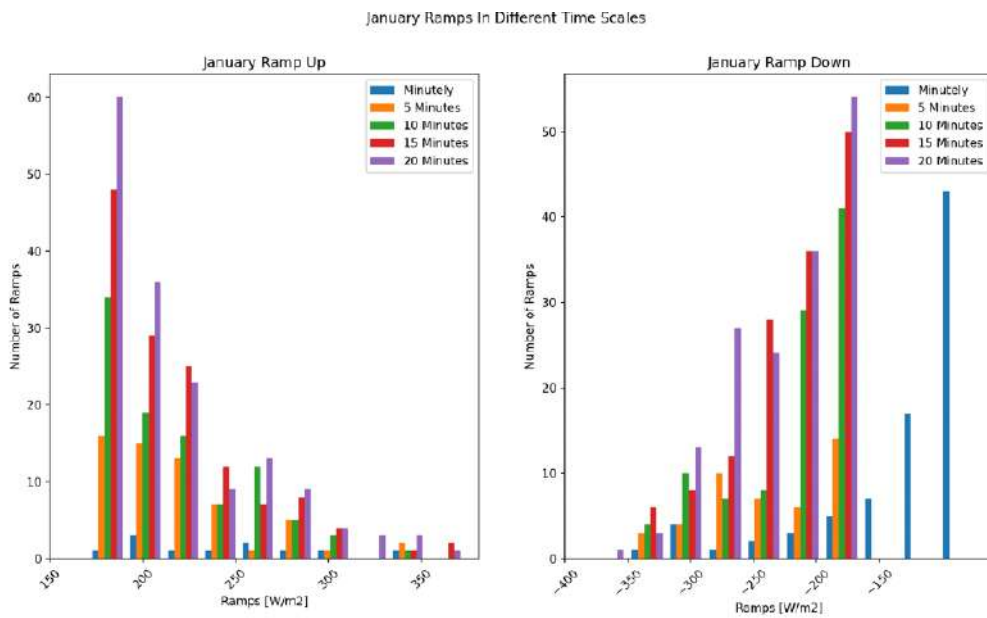


Figure 4.19: January Ramps Number

February Ramp Up In Different Time Scales

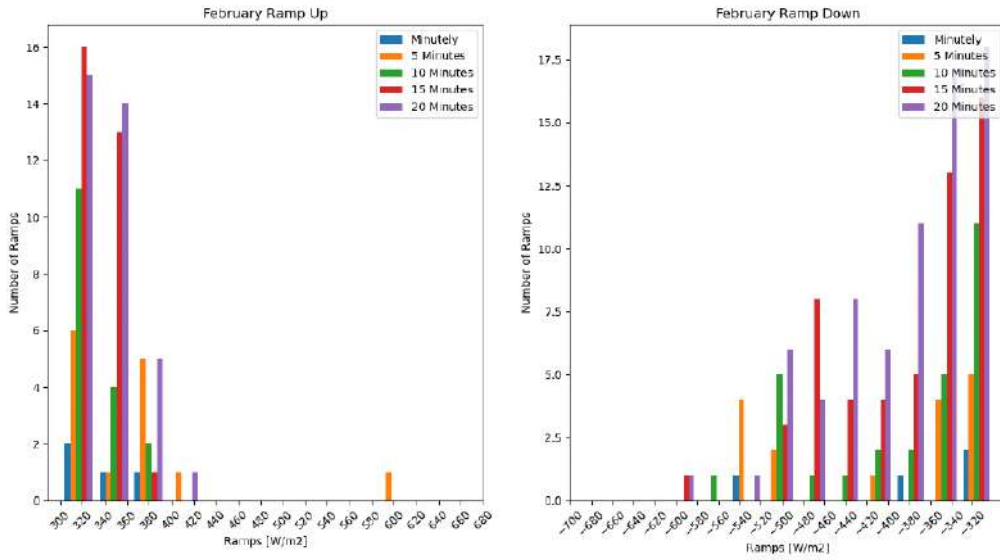


Figure 4.20: February Ramps Number

March Ramps In Different Time Scales

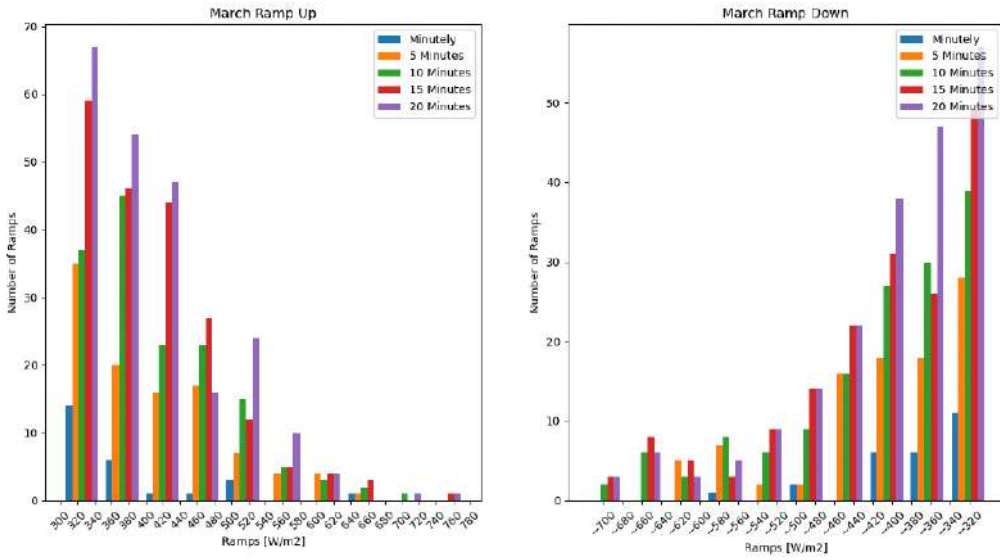
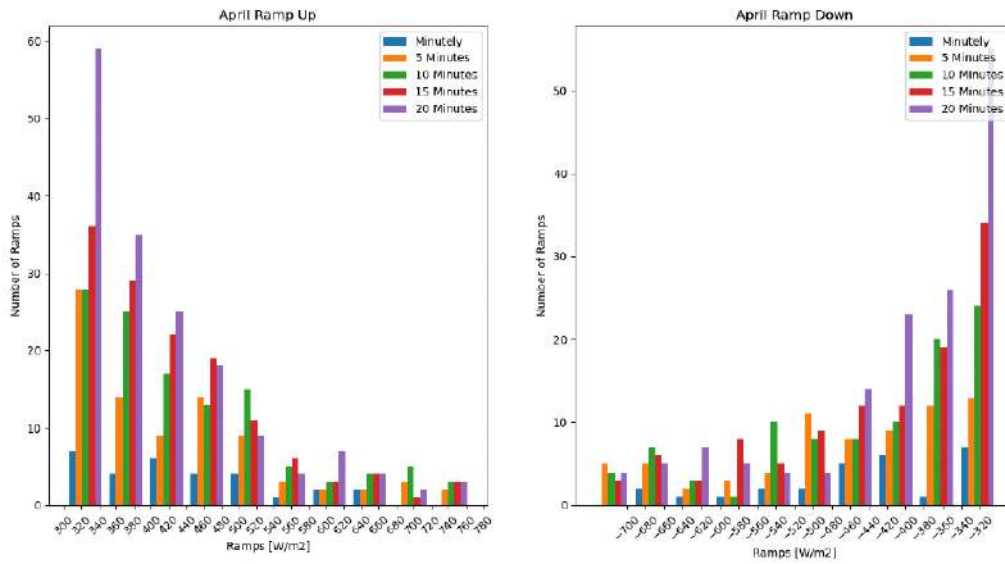


Figure 4.21: March Ramps Number

April Ramps In Different Time Scales



Chapter 5

Sky Images Analysis

This chapter analyzes the images captured from Solar Tech Lab [55] in Politecnico di Milano by ASI. These analyses' primary purpose is to make an image dataset for the Deep Learning and Machine Learning methods.

5.1 Image Analysis Starting Phase

Let us consider images 5.1 and 5.2, and then the specification of these images will be discussed. These are a subset of all the images in the dataset. As shown in these images, the dot point inside the white circle is Sun. The green shapes inside the images indicate the clouds. The deeper clouds are shown with dark blue color. In the next section, the way of producing the images is shortly discussed.

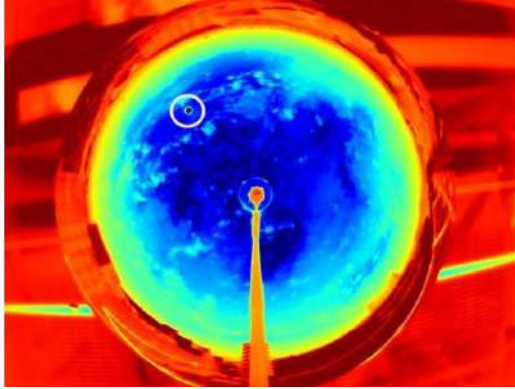


Figure 5.1: Captured Image on September 18, 2019, at 11:25 A.M

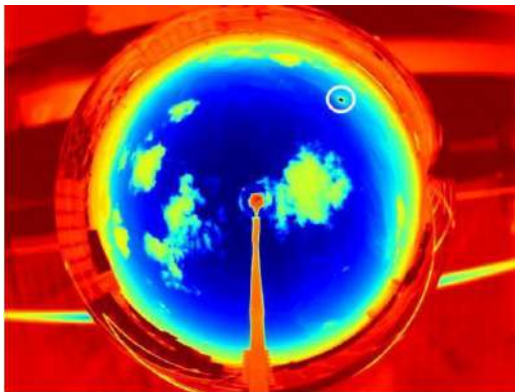


Figure 5.2: Captured Image on October 17, 2019, at 3:36 P.M

5.2 Preparation of Images

This section is a part of [5]. In this scenario, infrared sky images can effectively be implemented to provide information on local phenomena with very high accuracy. The use of these images can overcome some problems of the actual color. The most important one is related to the color saturation: infrared images are less impacted by the Sun's presence, as shown in Figure 5.3. Here, a comparison between the two types of images simultaneously is shown.

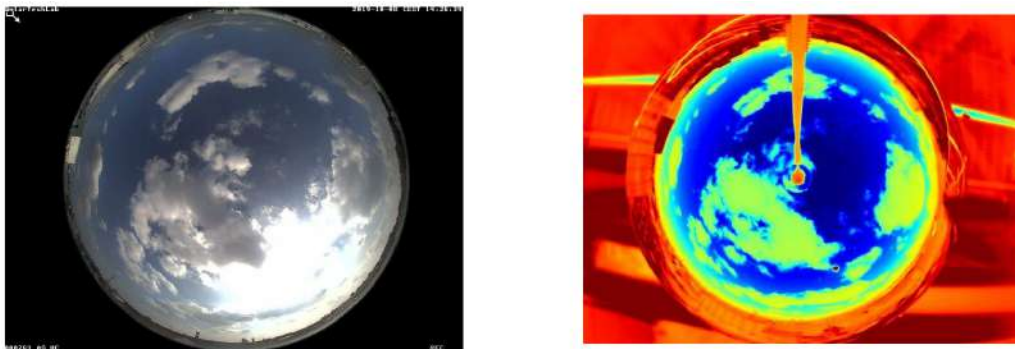


Figure 5.3: Captured Image on September 18, 2019, at 11:25 A.M

The infrared image (The figure at right) uses false colors to increase the contrast between the clear-sky (in blue) and the cloud coverage that can range from green to red according to the different types of clouds. In these images, the Sun is shown as a well-defined black dot.

5.3 Editing the Images

In this section, the goal is to make images ready for the Deep Learning process. Since the sun's location is the crucial point, the images are cropped by a square shape, so the sun is located in the middle of the images. The location of the sun is available for each minute in different CSV files for each month. In the figure below, some CSV file rows for September 18, 2019, can be seen.

time	GHI_clearSky	GHI_meas	sunPosition_x	sunPosition_y	imageName
9/18/2019 10:25	642.6533426	527	228.705332	137.6267469	img_20190918112500.jpg
9/18/2019 10:26	644.2195553	506	229.3902661	137.3967439	img_20190918112600.jpg
9/18/2019 10:27	645.7729161	541	230.0755107	137.1684604	img_20190918112700.jpg
9/18/2019 10:28	647.3133943	771	230.7610859	136.941913	img_20190918112800.jpg
9/18/2019 10:29	648.8409595	612	231.447013	136.7171186	img_20190918112900.jpg
9/18/2019 10:30	650.3555817	507	232.1333151	136.4940945	img_20190918113000.jpg
9/18/2019 10:31	651.8572309	424	232.820016	136.2728581	img_20190918113100.jpg
9/18/2019 10:32	653.3458777	529	233.5071411	136.0534269	img_20190918113200.jpg
9/18/2019 10:33	654.8214927	632	234.1947162	135.8358184	img_20190918113300.jpg
9/18/2019 10:34	656.2840467	442	234.8827685	135.6200502	img_20190918113400.jpg
9/18/2019 10:35	657.7335112	505	235.5713255	135.4061393	img_20190918113500.jpg

Figure 5.4: CSV file for September 18, 2019

As can be seen in figure 5.4, for each minute, the information below is available:

- GHI-clear sky: Received Irradiance by using CSL.
- GHI-meas: This Shows the measured GHI.
- sunPosition-x: This Shows the position of the sun on the x-axis in the images.
- sunPosition-y: This Shows the position of the sun on the y-axis in the images.
- imageName: This Shows the file name with which the file is stored.

For cropping the images based on the sun location, the Skimage library is used [56]. This library converts the images to a Numpy array [57].

A Numpy array is a grid of values, all of the same type, and is indexed by a tuple of non-negative integers. The number of dimensions is the array's rank; the shape of an array is a tuple of integers giving the array's size along each dimension.

Since the images are converted to different arrays, by considering the sun location in each moment, the images are cropped by taking a subset of these Numpy arrays.

In Figure 5.5, the code of the cropping procedure can be seen.

```
from skimage import data
import pandas as pd
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
import os
from natsort import natsorted, ns
from PIL import Image
import os.path, sys

image_list = []
Sun_position = pd.read_csv("20190918.csv", index_col = 0)

Sun_x_position = Sun_position["sunPosition_y"].tolist()
Sun_y_position = Sun_position["sunPosition_x"].tolist()

Sun_x_position = list(map(int, Sun_x_position))
Sun_y_position = list(map(int, Sun_y_position))

left = [l-50 for l in Sun_x_position]
right = [r+50 for r in Sun_x_position]
top = [t+50 for t in Sun_y_position]
bottom = [b-50 for b in Sun_y_position]

list_files = os.listdir('.')
list_files.remove("20190918.csv")
list_files.remove("Test1.py")

for filename in list_files:
    image_list.append(io.imread(filename))

for i in range(0, len(left)):
    cropped = image_list[i][left[i]:right[i], bottom[i]:top[i]]
    plt.imshow(cropped)
    plt.savefig(list_files[i])
    plt.close()
```

Figure 5.5: Code for cropping the images concerning the Sun location

There are some alternatives to do this process, like using OpenCV [58] or PIL [59] libraries, but the way used is more straightforward and faster.

As shown in Figure 5.5, the cropped image size is 100×100 pixels of the main image. In figures 5.6 and 5.7, the main image and the cropped image at the same time can be seen:

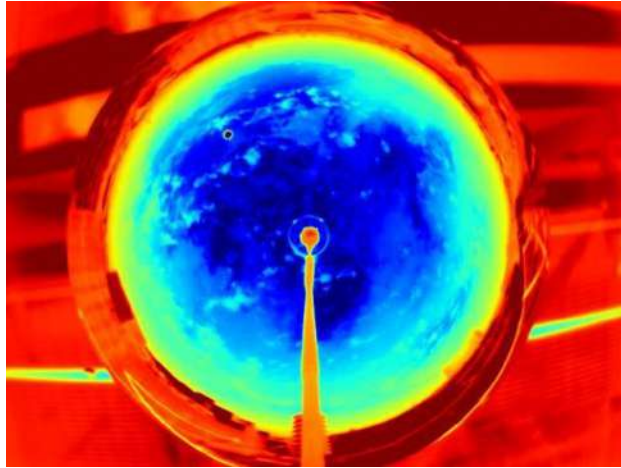


Figure 5.6: The Main Image

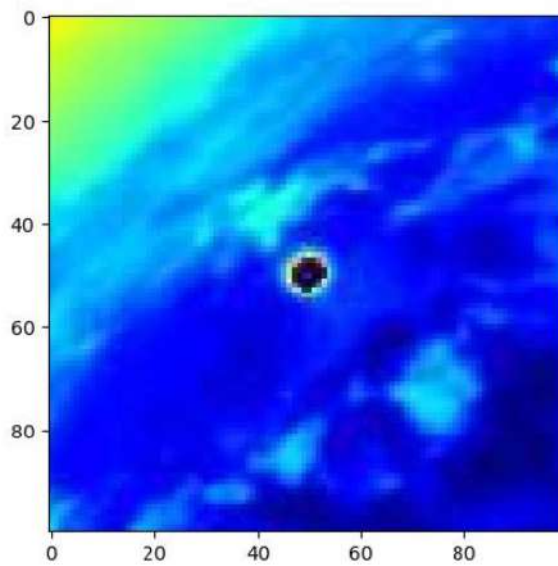


Figure 5.7: The Cropped Image

For all the images, this procedure is repeated. The total number of images is 85,000 images for 145 days, starting from September 18, 2019, and finishing April 9, 2020.

5.4 Dataset Creation Introduction

As mentioned in chapter 3, the main difference between Supervised Learning and Unsupervised Learning is the data label.

In order to label the data, four different methods are done. The last method is the best one. The main idea of classifying the images is based on GHI value.

Here, there is the list of the methods to label the data:

- Dataset Creation with three labels based on GHI.
- Dataset Creation with four labels based on GHI.
- Dataset Creation with another image processing method.
- Dataset Creation with hand and by the help of GHI.

5.5 Dataset Creation with three labels based on GHI

In the first dataset, the images are split into three different labels based on the GHI value:

- Sunny: This shows the sunny weather, which means that there are no clouds in the sky.
- Semi-Covered: This shows a sky with clouds, but it is not entirely-covered.
- Cloudy: Which shows a fully-covered sky.

This classification is done by Python and based on the GHI Value by considering the difference between Measured GHI and Clearsky GHI, specified as GHI-meas and GHI-clearSky for each image. The classification is done as listed below:

- Sunny: If the measured GHI is equal to or more than 70% of the Clear Sky GHI.
- Semi-Covered: If the measured GHI is between 40% and 70% of the Clear Sky GHI.
- Cloudy: If the measured GHI is lower than 40% of the Clear Sky GHI.

In figure 5.8, the code of this classification can be seen:

```
GHI_Meas = df["GHI_meas"]
GHI_Clear = df["GHI_clearSky"]
test = []
for i in range(0, len(GHI_Clear)):
    if GHI_Meas[i] >= 0.7 * GHI_Clear[i]:
        test.append(0)
    elif GHI_Meas[i] >= 0.4 * GHI_Clear[i] and GHI_Meas[i] < 0.7 * GHI_Clear[i]:
        test.append(1)
    else:
        test.append(2)
df1["label"] = test
df1.to_csv("labels.csv")
```

Figure 5.8: Python Code for Classification based on GHI in the first method

As shown in figure 5.8, the result of this classification is stored in a CSV file. In Figure 5.9, this CSV file can be seen:

27	img_20190918115200.jpg	0
28	img_20190918115300.jpg	0
29	img_20190918115400.jpg	0
30	img_20190918115500.jpg	1
31	img_20190918115600.jpg	1
32	img_20190918115700.jpg	1
33	img_20190918115800.jpg	2
34	img_20190918115900.jpg	2
35	img_20190918120000.jpg	2
36	img_20190918120100.jpg	2

Figure 5.9: First Classification Method labels

As shown in Figure 5.9, three different labels are indicated by 0, 1, and 2. The label Sunny is identified by 0, the label Semi-Covered is identified by 1, and the label Cloudy is identified by 2.

In figures 5.10 to 5.12, an image from each of these three labels can be seen:

As indicated in these images, in the Sunny image, the sky is clear with no clouds. In the Semi-Covered image, the sky is covered with clouds but not entirely. In the Cloudy image, the sky is fully covered so that Sun can not be found.

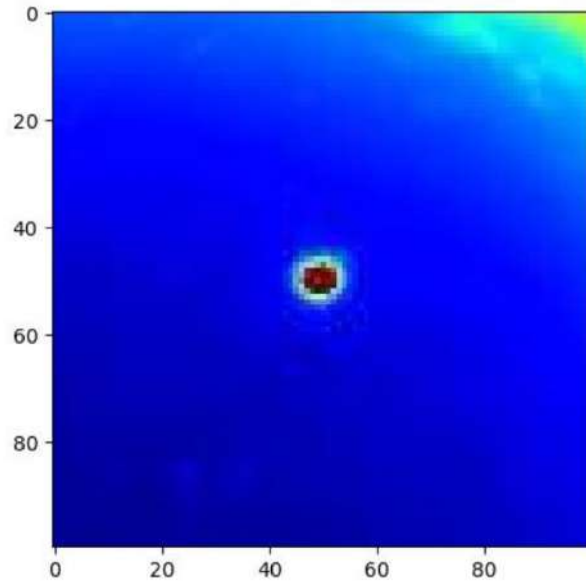


Figure 5.10: Image Labelled as Sunny in the First Classification Method

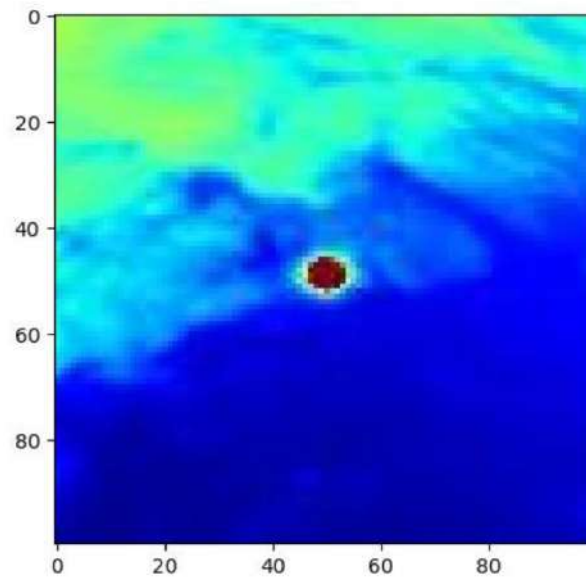


Figure 5.11: Image Labelled as Semi-Covered in the First Classification Method

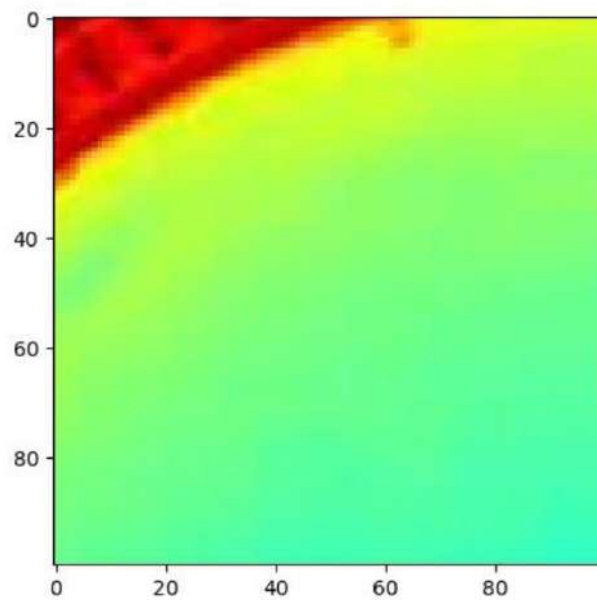


Figure 5.12: Image Labelled as Cloudy in the First Classification Method

However, in this method of classification, the number of images that are incorrectly labeled is large. The most significant number of incorrectly labeled images are for Semi-Covered in this classification method. In figures 5.13 to 5.15, an example of these incorrectly labeled images can be seen for each label:

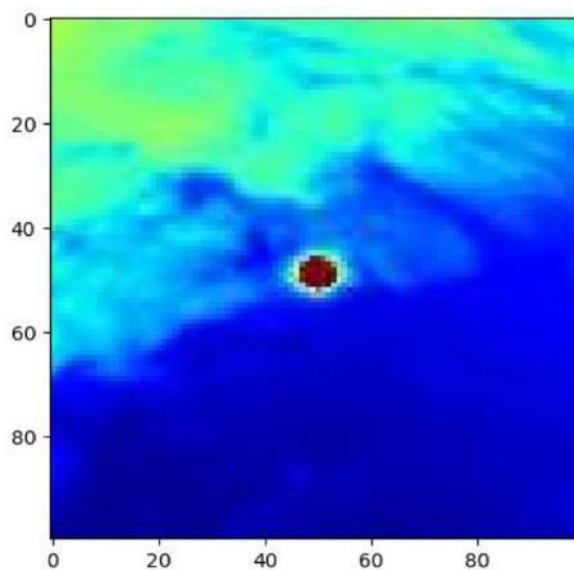


Figure 5.13: Image Incorrectly Labelled as Sunny in the first method

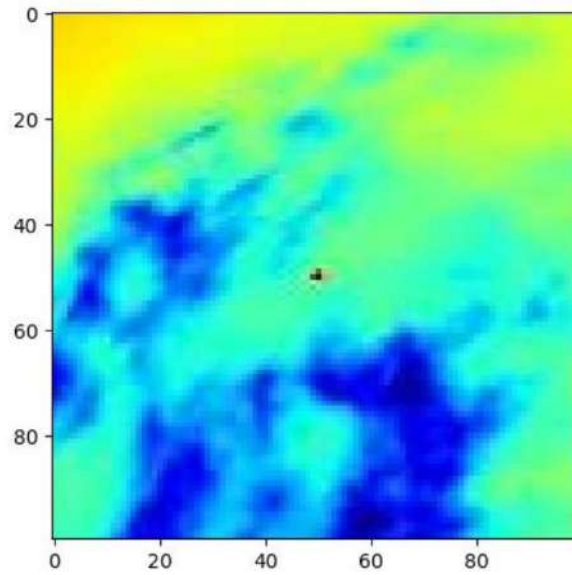


Figure 5.14: Image Incorrectly Labelled as Semi-Covered in the first method

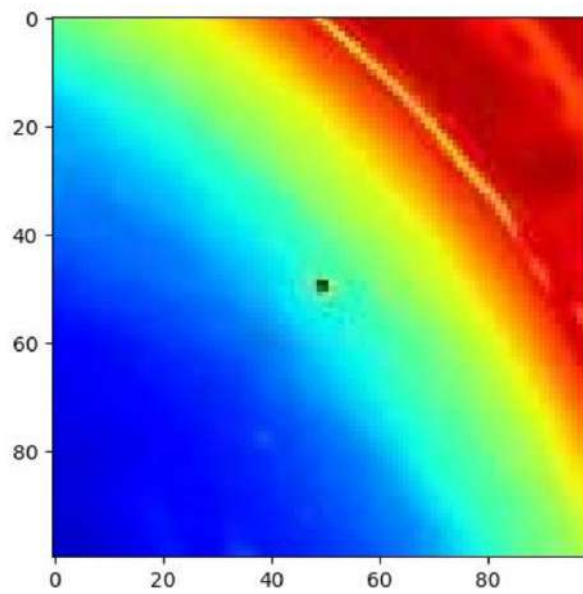


Figure 5.15: Image Incorrectly Labelled as Cloudy in the first method

As shown in Figure 5.13, the sky is covered by clouds, but it is not entirely-covered, so this image should have been labeled as Semi-Covered. Also, in figure 5.14, the sky is approximately fully-covered, so this image should have been labeled as Cloudy. Concerning many incorrectly- labeled images in this method, we tried a new method with 4 Labels.

5.6 Dataset Creation with four labels based on GHI

Since the most significant number of incorrectly labeled occurred in the Semi-Covered label, this label is split into two different parts Low-Covered and High-Covered. Also, since some Sunny label errors show some images with cloud in this label and some problems in Cloudy label, the classification percentage is changed. So, now there are four labels as listed below:

- Clear-Sky: If the Measured GHI is equal and higher than 80% of the Clear Sky GHI.
- Low-Covered: If the Measured GHI is between 50% and 80% of the Clear Sky GHI.
- High-Covered: If the Measured GHI is between 20% and 50% of the Clear Sky GHI.
- Fully Covered (Rainy): If the Measured GHI is lower than 20% of the Clear Sky GHI.

In figure 5.16, the code of this classification method can be seen:

```
for i in range(0, len(GHI_Clear)):
    if GHI_Meas[i] >= 0.8 * GHI_Clear[i]:
        test.append(0)
    elif GHI_Meas[i] >= 0.5 * GHI_Clear[i] and GHI_Meas[i] < 0.8 * GHI_Clear[i]:
        test.append(1)
    elif GHI_Meas[i] >= 0.2 * GHI_Clear[i] and GHI_Meas[i] < 0.5 * GHI_Clear[i]:
        test.append(2)
    else:
        test.append(3)

df1["label"] = test
df1.to_csv("labels.csv")
```

Figure 5.16: Python Code for Classification based on GHI in the second method

In figures 5.17 to 5.20, an image from each label can be seen. For the image labeled as Clear Sky, a blue sky without any clouds is shown in these figures. In the image labeled as Low Covered, a sky lowly covered with clouds is indicated. In the Highly Covered image, we can see that the sky is approximately full of clouds. Moreover, in the image labeled as Fully Covered (Rainy), rain in the image is visible.

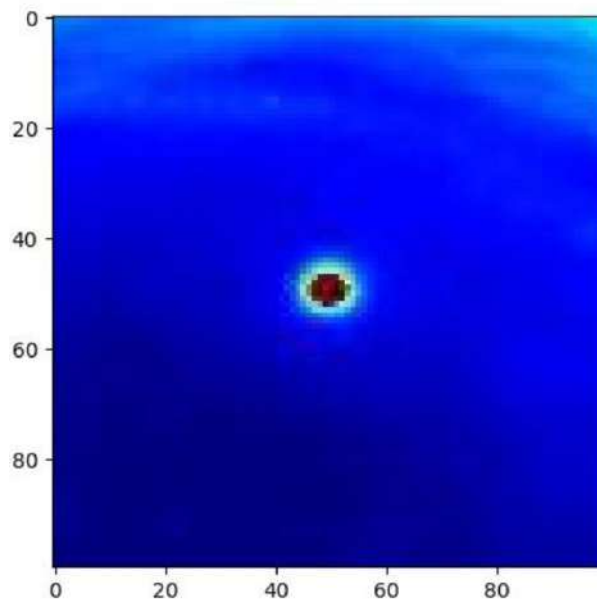


Figure 5.17: Image Labelled as Clear Sky in the Second Method

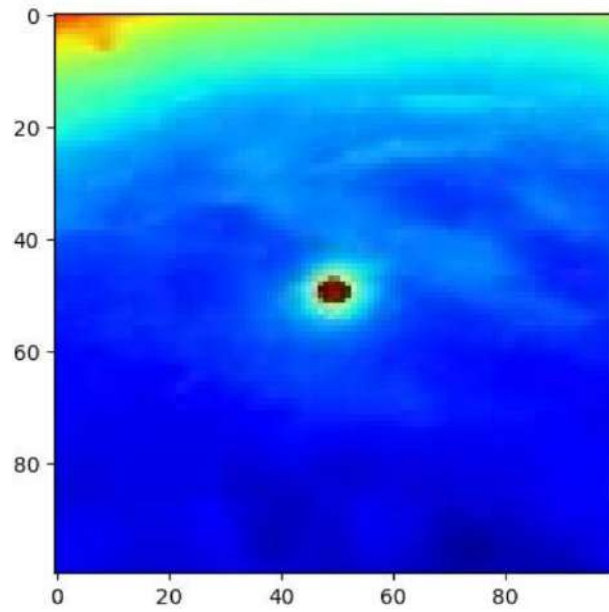


Figure 5.18: Image Labelled as Low-Covered in the Second Method

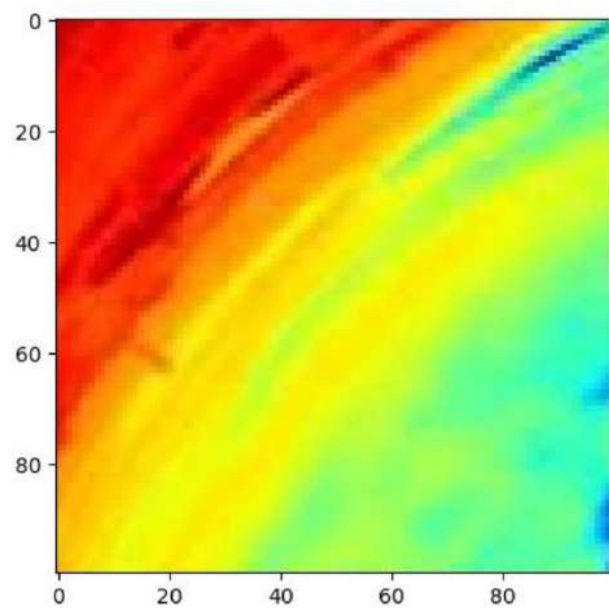


Figure 5.19: Image Labelled as High-Covered in the Second Method

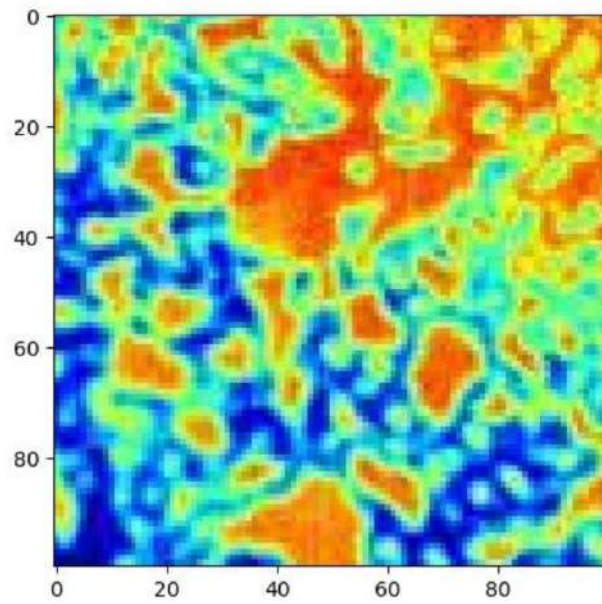


Figure 5.20: Image Labelled as Rainy in the Second Method

However, again in this method, there are some errors. In the list below, some of these errors can be seen:

- In the Clear Sky label, some images are covered with clouds that should be labeled as Low Covered images.
- Some images related to Clear Sky and some images related to the Highly Covered label exist in the Low Covered label.
- There are some images related to the Low Covered label and Fully Covered Label in the Highly Covered label.
- In the Fully Covered Label, some images should be labeled Highly Covered because there was no rain in the images.

The minor error is in the Clear Sky label and Fully Covered label. Some modifications were done by hand to fix the issues. Finally, the errors in Clear Sky Label and Fully Covered (Rainy) label are fixed. However, realizing the difference between Low Covered and Highly Covered images was a big problem.

In figures 5.21 to 5.23, some of these problems can be seen:

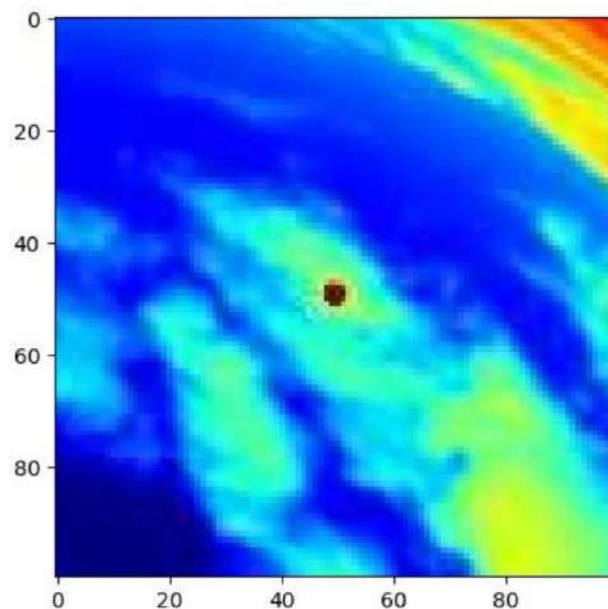


Figure 5.21: Confusion Between Low-Covered or High-Covered Label

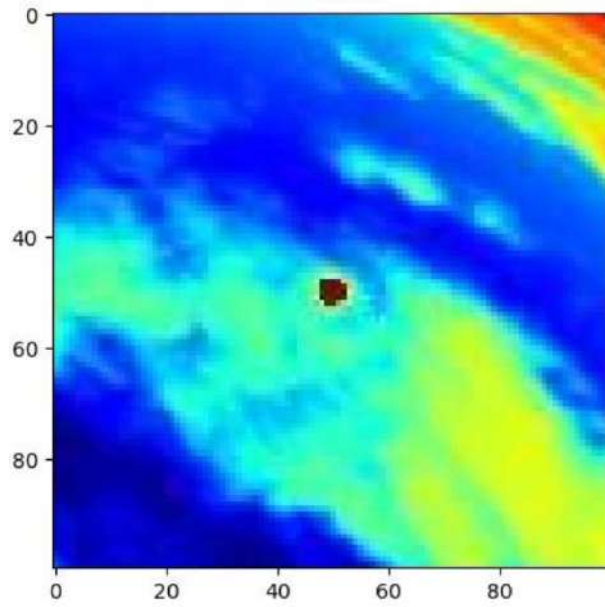


Figure 5.22: Confusion Between Low-Covered or High-Covered Label

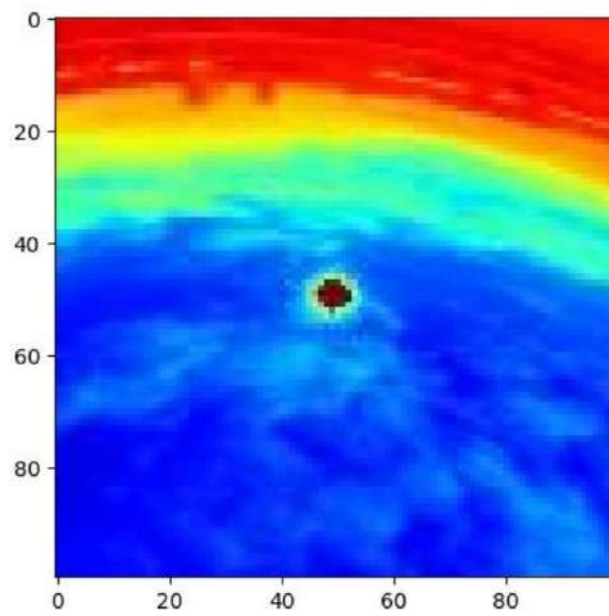


Figure 5.23: Confusion Between Low-Covered or High-Covered Label

As shown in these images, since the clouds' coverage is between Low-Covered and High-Covered, it poses a problem of labeling these images. Since there are many images with this problem in the dataset for this method, it would be good to choose another method.

5.7 Dataset Creation with another image processing method

In this method, another way to create the images and make them ready for the dataset is used. Because in the images, colors have the leading role in classification, so by removing the square red area from images, the classification would be more straightforward. In figures 5.24 and 5.25, the main image and the edited image by using this method can be seen:

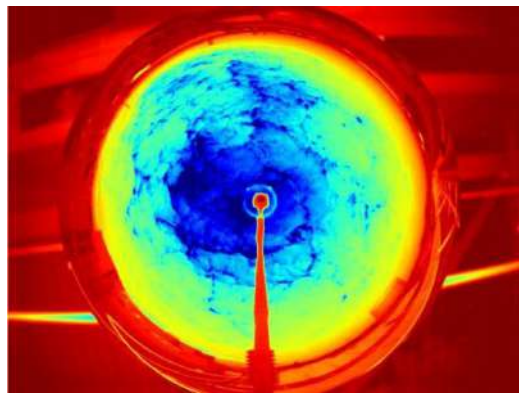


Figure 5.24: Captured Image on September 20, 2019, at 7:15 A.M

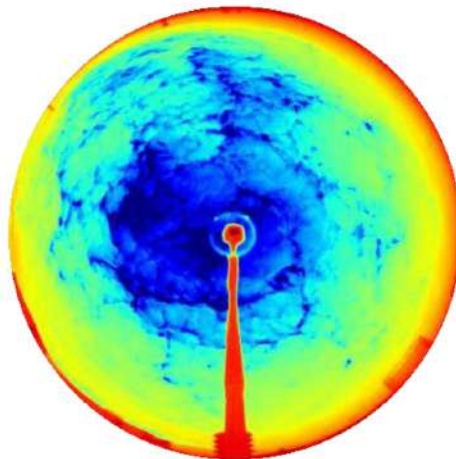


Figure 5.25: Edited Image

In figure 5.26 and 5.27, the code for this method can be seen:

```
import os
import glob
from PIL import Image, ImageDraw, ImageFilter
import numpy as np

def mask_circle_solid(pil_img, background_color, blur_radius, offset=0):
    background = Image.new(pil_img.mode, pil_img.size, background_color)

    offset = blur_radius * 2 + offset
    mask = Image.new("L", pil_img.size, 0)
    draw = ImageDraw.Draw(mask)
    draw.ellipse((offset, offset, pil_img.size[0] - offset, pil_img.size[1] - offset))
    #mask = mask.filter(ImageFilter.GaussianBlur(blur_radius))

    return Image.composite(pil_img, background, mask)

def mask_circle_transparent(pil_img, blur_radius, offset=0):
    offset = blur_radius * 2 + offset
    mask = Image.new("L", pil_img.size, 0)
    draw = ImageDraw.Draw(mask)
    draw.ellipse((offset, offset, pil_img.size[0] - offset, pil_img.size[1] - offset))
    #mask = mask.filter(ImageFilter.GaussianBlur(blur_radius))

    result = pil_img.copy()
    result.putalpha(mask)

    return result

def crop_max_square(pil_img):
    return crop_center(pil_img, min(pil_img.size), min(pil_img.size))

def crop_center(pil_img, crop_width, crop_height):
    img_width, img_height = pil_img.size
    return pil_img.crop(((img_width - crop_width) // 2,
                          (img_height - crop_height) // 2,
                          (img_width + crop_width - 26) // 2,
                          (img_height + crop_height) // 2))
```

Figure 5.26: Python Code for removing Red Part

```
for i in range(0, len(list_files)):
    im[i] = Image.open(list_files[i])
    im_square = crop_max_square(im[i]).resize((thumb_width, thumb_length), Image.LANCZOS)
    im_thumb = mask_circle_transparent(im_square, 15)
    im_thumb.save(list_files2[i])
```

Figure 5.27: Python Code for removing Red Part

As can be seen in the figures above, the PIL library is used to do this. Furthermore, a circle mask is created and applied to the image, and then the Red part from the main image is cropped.

Nevertheless, this method has a problem in finding the location of the sun. As shown in the images taken from [5], the predicted trajectory for the sun can be seen in figures 5.28 to 5.30:

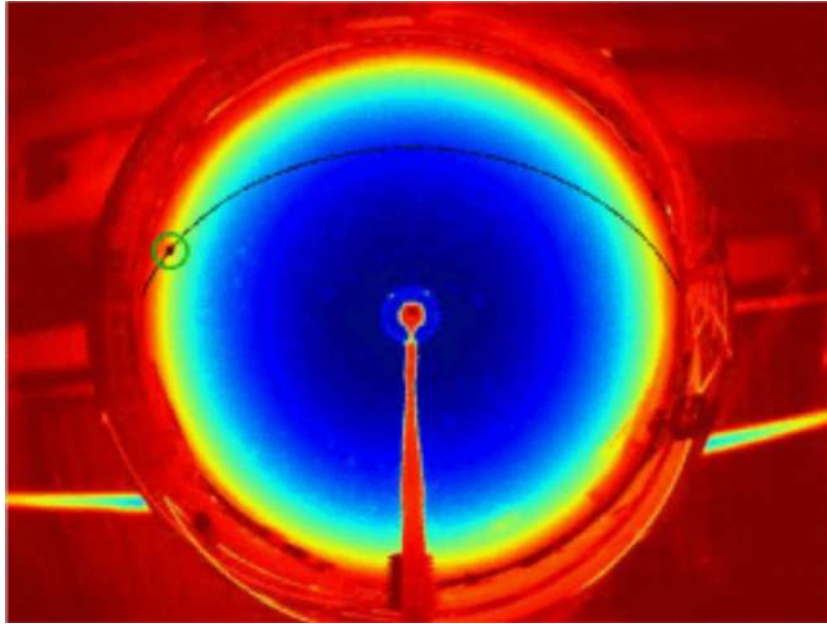


Figure 5.28: Sun Trajectory Obtained by ANN on September 29, 2019, at 8:29 AM

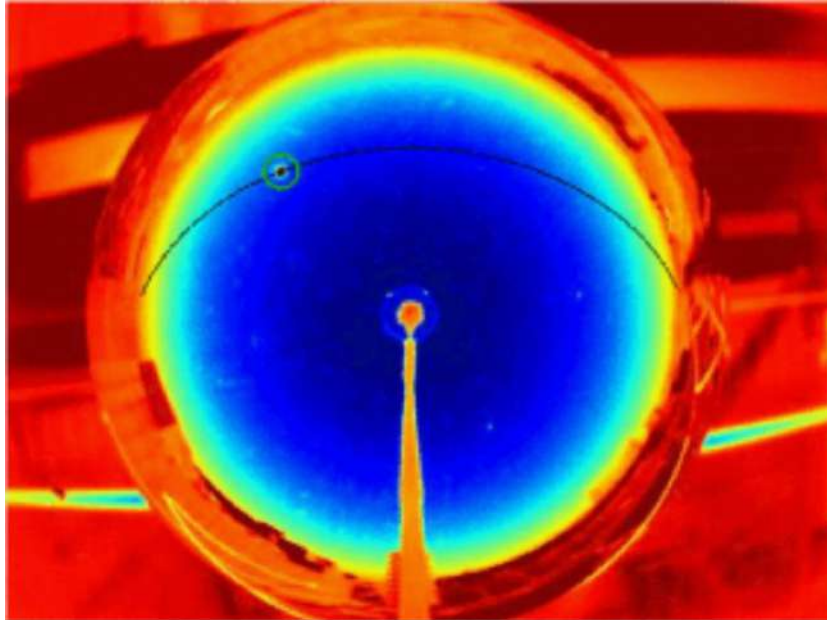


Figure 5.29: Sun Trajectory Obtained by ANN on September 29, 2019, at 10:59 AM

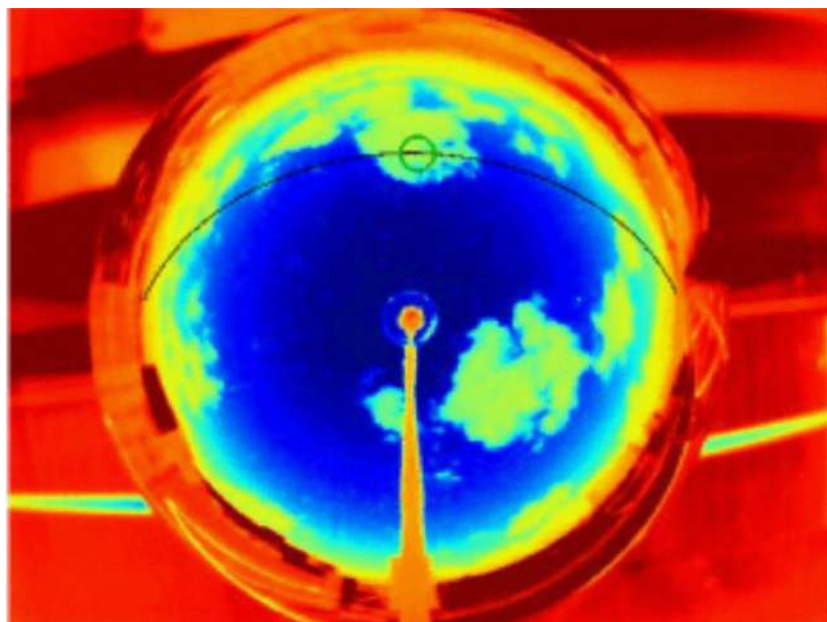


Figure 5.30: Sun Trajectory Obtained by ANN on September 29, 2019, at 13:23 AM

As can be seen in these images, the sun's trajectory is like an arc. The problem in this method occurs at deleting the Red area from the images. As mentioned previously, this job was done by creating a circular mask and applied that to the main images. So the Sun X and Y position change non-linearly in the images, because the cropping size should be calculated and then reduce it from the Sun X and Y position. However, since the trajectory is an arc and is not a straight line, some problems are posed. Because It could not crop the image with different sizes, also it is impossible to apply a specific mask to each image in this large dataset.

5.8 Dataset Creation with hand and by the help of GHI

This dataset is the dataset used for the next step, but another classification should be done before applying Deep Learning models on this dataset. This classification will be discussed in the next section.

This dataset has no errors, and it is created by using the percentages in section 5.5 and then reclassify them by hand. In this dataset, there are three labels: Clear Sky, Covered, Rainy.

In figures 5.31 to 5.34, some images of this dataset can be seen. As can be seen in these images, there are no clouds in the sky for the Clear Sky label. Also, in the covered label, as shown in the images, the images are in the same Covered label for both Low-Covered and High-Covered labels. Finally, for the images in which the weather is rainy, a separate label is assigned. In this new dataset, there is no error to enter the Deep Learning model. The number of images in each label is:

- Clear Sky:28704 Images
- Covered:41244 Images
- Rainy:14907 Images
- Total Number of Images: 84855 Images

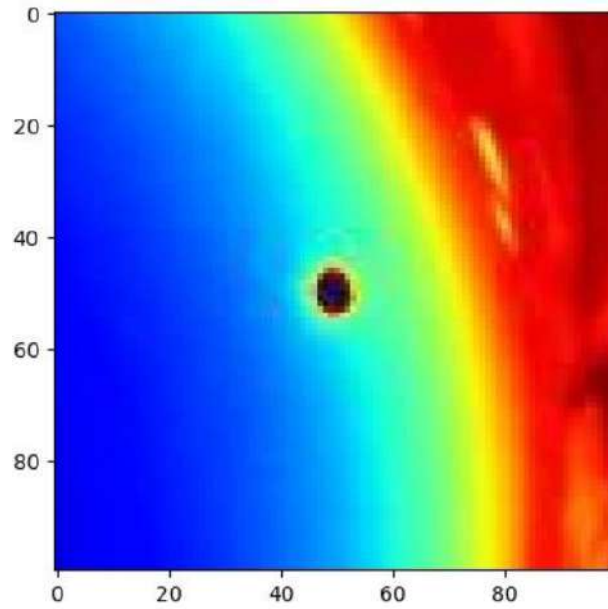


Figure 5.31: Image Labeled as Clear sky on September 19, 2019, at 5:41 PM

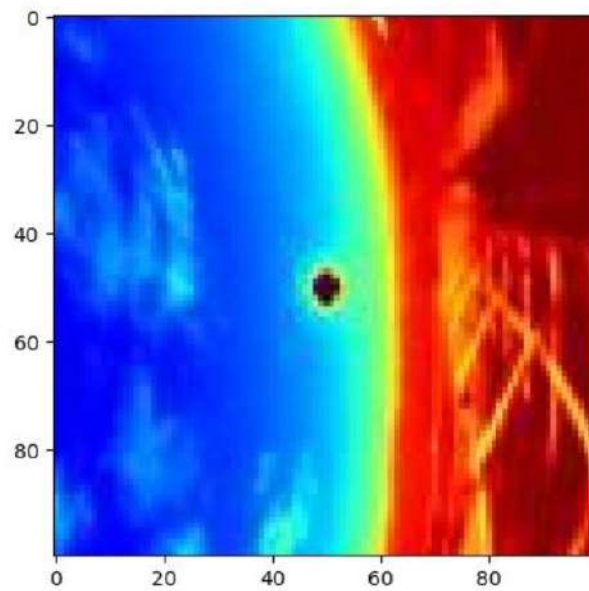


Figure 5.32: Image Labeled as Covered on March 31, 2020, at 6:46 PM

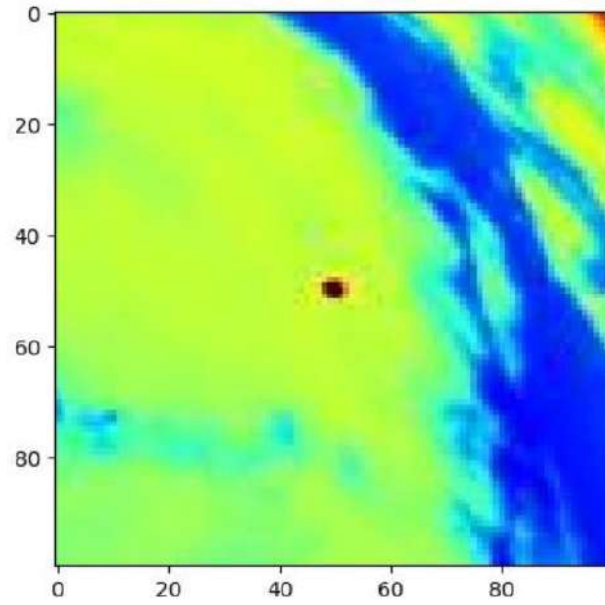


Figure 5.33: Image Labeled as Covered on April 1, 2020, at 4:07 PM

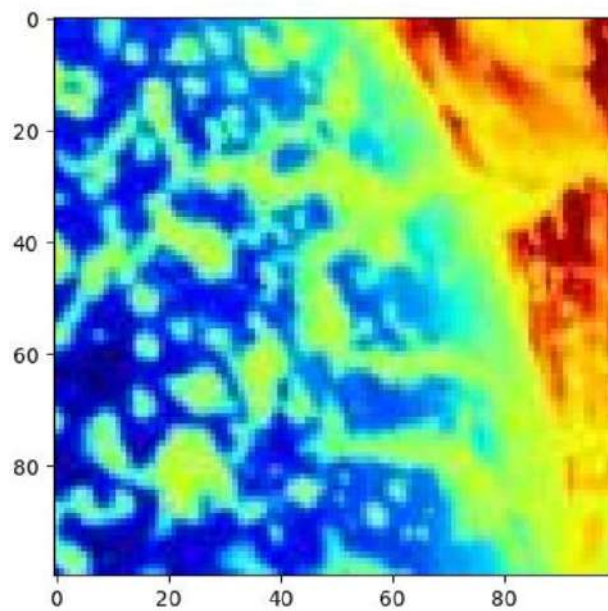


Figure 5.34: Image Labelled as Rainy on September 22, 2019, at 5:27 PM

5.9 Last Classification

As can be seen in figure 5.35, for the next classifier, the Rainy label is removed because in rainy weather, the GHI is very low, and the produced energy from solar cells is negligible.

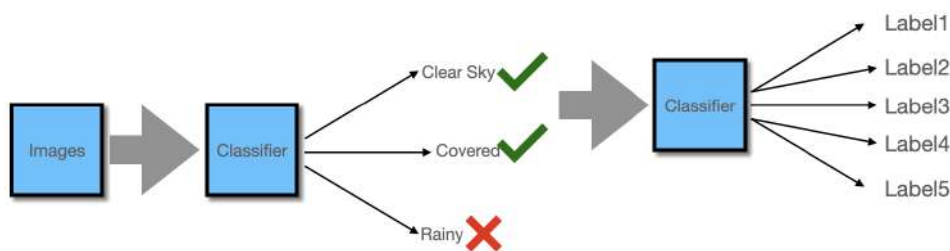


Figure 5.35: Classification Road Map

The second classifier is based on CSI value.

CSI is the ratio of measured irradiance at ground level to estimated irradiance in clear sky conditions. It is also utilized in a clear sky model and information, on atmospheric input parameters is available [60].

In figure 5.36, the ".mat" file of labeling based on CSI can be seen:

dates	files	GHI_meas	GHI_clearSky	Kt	labels_CSI
18-Sep-2019 10:25:00	20190918/img_20190918112500.jpg	527	739.051343989679	0.713076302865583	3
18-Sep-2019 10:26:00	20190918/img_20190918112600.jpg	506	740.852488607599	0.682996963337473	3
18-Sep-2019 10:27:00	20190918/img_20190918112700.jpg	541	742.638853483459	0.728483296372602	3
18-Sep-2019 10:28:00	20190918/img_20190918112800.jpg	771	744.410403391003	1.03571900189448	5
18-Sep-2019 10:29:00	20190918/img_20190918112900.jpg	612	746.167103404411	0.820191612854186	4
18-Sep-2019 10:30:00	20190918/img_20190918113000.jpg	507	747.90891889919	0.677890030708857	3
18-Sep-2019 10:31:00	20190918/img_20190918113100.jpg	424	749.635815552271	0.565607980840178	3
18-Sep-2019 10:32:00	20190918/img_20190918113200.jpg	529	751.347759342097	0.70406811416222	3
18-Sep-2019 10:33:00	20190918/img_20190918113300.jpg	632	753.044716549537	0.839259589916299	4
18-Sep-2019 10:34:00	20190918/img_20190918113400.jpg	442	754.726653757951	0.585642494271514	3
18-Sep-2019 10:35:00	20190918/img_20190918113500.jpg	505	756.393537853325	0.667641875197943	3
18-Sep-2019 10:36:00	20190918/img_20190918113600.jpg	385	758.045336025138	0.507885190638826	3
18-Sep-2019 10:37:00	20190918/img_20190918113700.jpg	361	759.68201576648	0.475198823333694	3

Figure 5.36: ".Mat" file of classification based on CSI value

As can be seen in figure 5.36, the "Kt" column is a CSI value that is equal to "GHI-meas/GHI-clearsky."

Concerning the value of CSI, images are classified into five different labels, which can be seen below:

- Label1: $0 < Kt < 0.2$
- Label2: $0.2 < Kt < 0.45$
- Label3: $0.45 < Kt < 0.75$
- Label4: $0.75 < Kt < 1$
- Label5: $Kt > 1$

Also, in the ".Mat" file, some images with label 0 exist related to images near Sunset and Sunrise when the measured GHI and ClearSky GHI are very close to each other and have small values. Since the values of both Measured GHI and ClearSky GHI are shallow, so they are ignored.

After removing the Rainy labeled images and "Label 0" images, the number of images in each label is listed below:

- Class1: 6427 Images
- Class2: 12678 Images
- Class3: 9936 Images
- Class 4: 26077 Images
- Class 5: 1635 Images

The total number of images is 56753.

Chapter 6

Deep Learning Model Application

In this chapter, first, we prepare an introduction about CNN and the used model with a description of each layer. Another image modification was also used to prepare them for Deep Learning, which will be discussed in this chapter.

6.1 Why Deep Learning?

In figure 6.1 [61], the difference between ML and Deep Learning can be seen:

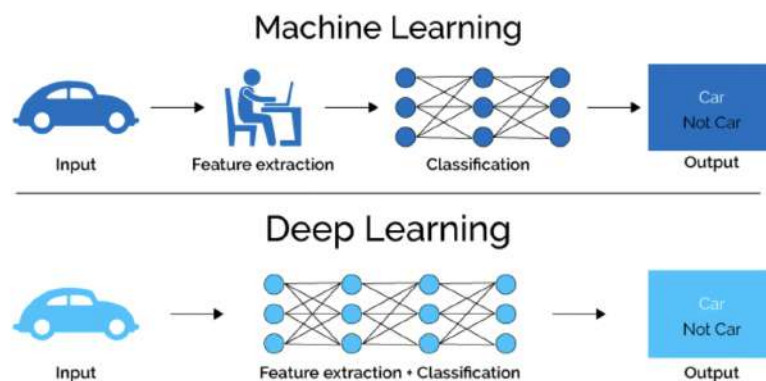


Figure 6.1: Difference between ML and Deep Learning

Deep Learning is used when:

- Dealing with Unstructured Data
- Do not need easily Interpretable Result
- Benefit from a known architecture

6.2 How does ANN work?

In figure 6.2 [62], the architecture of ANN can be seen. $W1$, $W2$, and $W3$ are the strength of the input signals. Also, as can be seen in the figure:

$$F = W1 \times X1 + W2 \times X2 + W3 \times X3$$

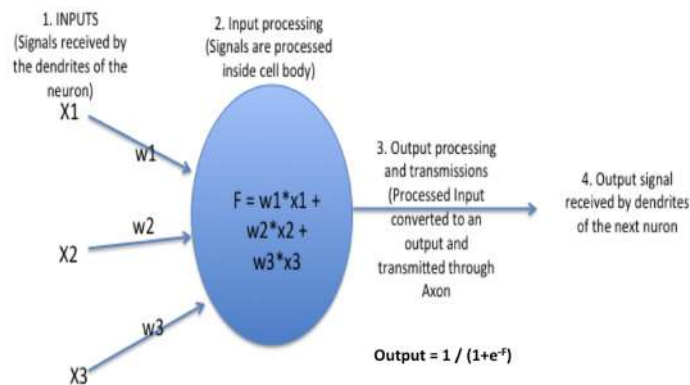


Figure 6.2: ANN Architecture

In figure 6.3 [62], a complete architecture of the ANN model can be seen:

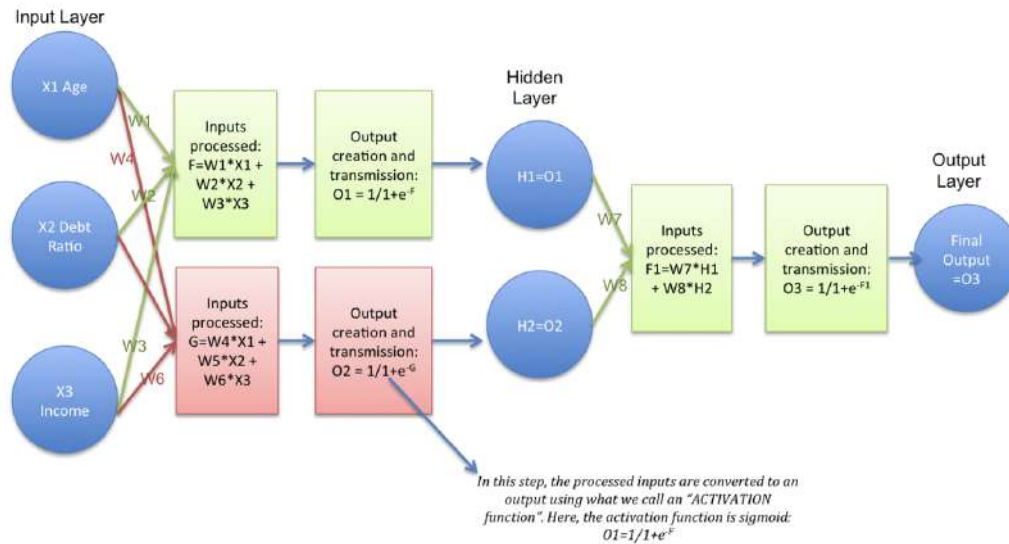


Figure 6.3: ANN Functional Architecture

Key points related to this architecture:

- There are three main layers in the architecture; An input layer, hidden layers (usually more than 1), and the output layer equal to the classes' number.
- Filtering the inputs and passing them to the next layer is the hidden layer's duty, which makes the network faster and efficient.
- The activation function serves to detect the non-linear relation between the inputs. Conversion of the inputs to a more valuable output is the other activation function's duty.

The Key advantages of Neural Network are [63]:

- Learning and modeling non-linear and complex relationships which is similar to the non-linear inputs and outputs in real-life.
- Generalizing is the other advantage of ANNs. They can infer the unknown and new relationships after learning and modeling the initial inputs and outputs and making a model which can generalize and predict the new and unseen data.
- There are no restrictions in the input variables in ANN, unlike the other methods.

6.3 Important Activation Functions

Many activation functions can be used in ANN. The list of these functions is listed below. Then a short description of the more critical activation functions is discussed.

The activation functions used in ANN are [64]:

- Binary Step Function
- Linear
- Sigmoid
- Tanh
- ReLU
- Leaky ReLU
- Parametrized ReLU
- Exponential Linear Unit
- Swish
- SoftMax

In figures 6.4 to 6.6 the images of mostly used activation functions are shown. The other activation functions are represented in Appendix B.

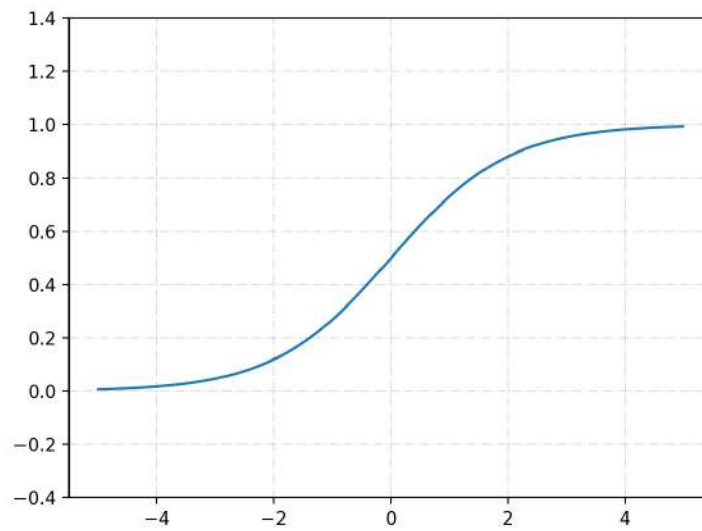


Figure 6.4: Sigmoid Activation Function

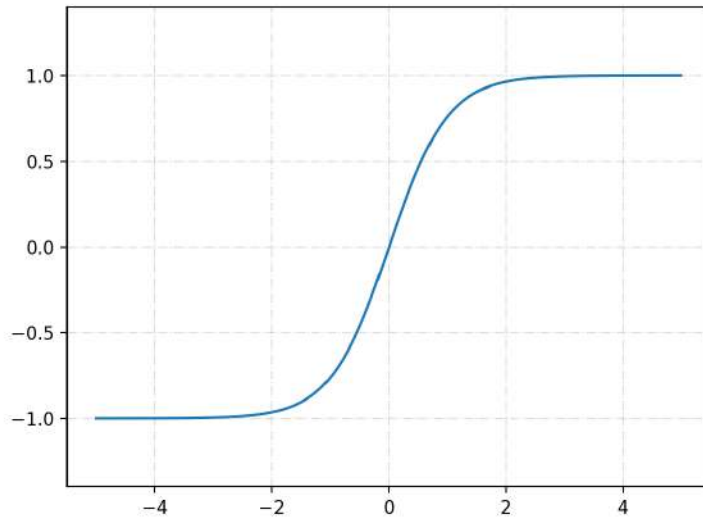


Figure 6.5: Tanh Activation Function

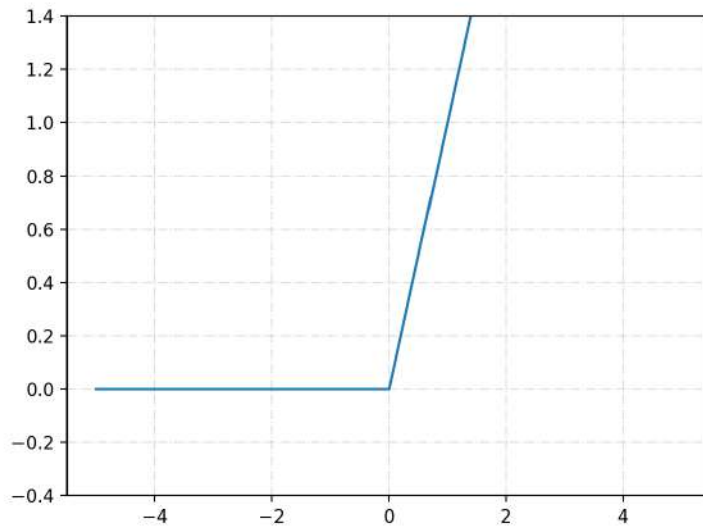


Figure 6.6: ReLU Activation Function

The mostly used activation functions are:

- Sigmoid
- Tanh
- ReLU
- Softmax

6.3.1 Sigmoid

The sigmoid function includes values between 0 and 1. Concerning this, sigmoid can be useful in cases where the goal is probability. Sigmoid is a differentiable function which means that its slope can be found at any two points [64].

6.3.2 Tanh

Sigmoid and Tanh activation functions are very similar. The difference between these functions is that Tanh is also available for the values 0 to -1. So, Tanh is mainly used for classification between two output classes [65].

6.3.3 ReLU

The most used activation function which is used in both Deep Learning and Machine Learning models is ReLU. In ReLU, there are no negative values since the negative values become zero. So, the model's ability is decreased to fit or train the data correctly. It means that for the negative data, ReLU is not working properly.

6.3.4 SoftMax

Softmax converts a vector of numbers into a probabilities vector proportional to the relative scale of each value in the vector as a mathematical function. It is used in multiple outputs models [66].

In the next section, we introduce the model used for Deep Learning on our image dataset, following a short introduction about the layers.

6.4 Deep Learning Model:

In figure 6.7, the summary of the model can be seen. As indicated in this image, the layers used for Deep Learning are listed below:

- Convolutional Layer
- Maxpooling Layer
- Flatten Layer
- Fully Connected Layer (Dense)

6.4.1 Convolutional Layer

There are several filters in Convolutional Layers. These filters contain different parameters which need to be learned. The activation map of neurons is calculated with the convolution of each filter and the input volume. In other words, the filter is slid across the input's width and height, and the dot products between the input and filter are computed at every spatial position. The convolutional layer's output volume is obtained by stacking all filters' activation maps along the depth dimension. Since each filter's width and height are designed to be smaller than the input, each neuron in the activation map is only connected to a small local region of the input volume. In other words, each neuron's receptive field size is small and equal to the filter size [67].

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792
block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928
block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0
block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856
block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584
block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0
block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168
block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080
block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080
block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0
block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160
block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808
block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808
block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0
block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808
block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808
block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808
block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0
flatten (Flatten)           (None, 25088)              0
fc1 (Dense)                 (None, 4096)                102764544
fc2 (Dense)                 (None, 4096)                16781312
dense (Dense)               (None, 5)                   20485
-----
Total params: 134,281,029
Trainable params: 20,485
Non-trainable params: 134,260,544

```

Figure 6.7: Deep Learning Model Layers

6.4.2 MaxPooling Layer

Maxpooling Layer reduces the spatial size of the image. It converts the image to a matrix and then, concerning the size of the kernel, creates a smaller matrix so that each argument of the smaller matrix is the maximum of the selected part of the main matrix. For example, suppose the image is a 224 by 224 matrix, and the kernel size is 2. In that case, there is a two by two matrix, which passes through all the primary matrix arguments and selects the maximum of the four arguments in that selected part. It means that the pooling layer reduces the size of the image by the kernel size. In figure 6.8, the process of max pool layer can be seen:

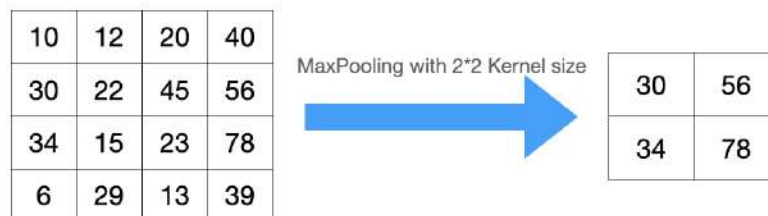


Figure 6.8: MaxPooling Layer Process

6.4.3 Flatten Layer

Flatten Layer converts the pooled layer's output to a simple vector that is ready to enter to the Fully Connected Layer. As shown in the model, there is a Flatten Layer after all the pooling layers before the last Fully Connected Layer (Dense). Flatten Layer reshapes the input vector to a simple vector.

6.4.4 Fully Connected Layer

It makes a full connection between the inputs neuron and the outputs neuron. The Dense layer creates a complete connection to all the previous activation functions and neurons.

In the used model, all the activation functions are ReLU, and for the last layer, the activation function is softmax. As mentioned in the introduction,

for multilabel cases, Softmax activation is used. Also, the last layer's output neurons are five, which is equal to the number of labels in our dataset.

6.5 Training Deep Learning Model on the Dataset

The first step in training the Deep Learning model is to split the dataset into Training Set, Validation Set, and Test Set. In order to split the data set, the code which can be seen in figure 6.9 is used.

```
import os
import numpy as np
import shutil
import random

# Creating Train / Val / Test folders

root_dir = "."
classes_dir = ['/label1', '/label2', '/label3', '/label4', '/label5']

val_ratio = 0.15
test_ratio = 0.10

for cls in classes_dir:
    os.makedirs(root_dir + '/train' + cls)
    os.makedirs(root_dir + '/val' + cls)
    os.makedirs(root_dir + '/test' + cls)

# Folders to copy images from
src = root_dir + cls

allFileNames = os.listdir(src)
allFileNames.sort()
# allFileNames.remove(".DS_Store")
# np.random.shuffle(allFileNames)
train_FileNames, val_FileNames, test_FileNames = np.split(np.array(allFileNames),
                                                           [int(len(allFileNames)* (1 - (val_ratio + test_ratio))),
                                                            int(len(allFileNames)* (1 - test_ratio))])

train_FileNames = [src+'/'+ name for name in train_FileNames.tolist()]
val_FileNames = [src+'/'+ name for name in val_FileNames.tolist()]
test_FileNames = [src+'/'+ name for name in test_FileNames.tolist()]

print('Total images: ', len(allFileNames))
print('Training: ', len(train_FileNames))
print('Validation: ', len(val_FileNames))
print('Testing: ', len(test_FileNames))

# Copy-pasting images
for name in train_FileNames:
    shutil.copy(name, root_dir + '/train' + cls)

for name in val_FileNames:
    shutil.copy(name, root_dir + '/val' + cls)

for name in test_FileNames:
    shutil.copy(name, root_dir + '/test' + cls)
```

Figure 6.9: Code for Splitting the Dataset

As can be seen in the figure, the dataset is split by the percentage listed below:

- Training Set: 75%
- Validation Set: 15%
- Test Set: 10%

Also, to avoid Class Imbalance concerning larger number of images in Label 4, the number of images is reduced to 10000 images.

The number of images in each Set is:

- Training Set: 30506 Images
- Validation Set: 6101 Images
- Test Set: 4069 Images

6.6 Image Preprocessing

As mentioned in the previous section, image processing is done to all the images to prepare them for the Deep Learning method. For this part, the white square around all the images is removed. As shown in figures 6.10 and 6.11, the main image contains a white square around the core image.

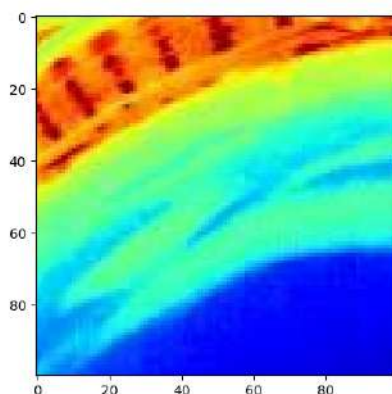


Figure 6.10: The Main Image before Removing the White Square

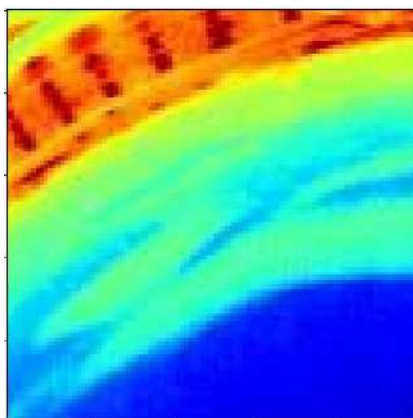


Figure 6.11: The Main Image after Removing the White Square

After this step, the images should be prepared to enter the Deep Learning Method. So, they are passed into preprocessing function of Keras. In figure 6.12, the code to make the images ready to enter the Deep Learning model can be seen:

```

train_batches = ImageDataGenerator(preprocessing_function=tf.keras.applications.vgg16.preprocess_input) \
    .flow_from_directory(directory=train_path, target_size=(224,224), classes=['label1', 'label2', 'label3', 'label4', 'label5'], batch_size=32)

test_batches = ImageDataGenerator(preprocessing_function=tf.keras.applications.vgg16.preprocess_input) \
    .flow_from_directory(directory=test_path, target_size=(224,224), classes=['label1', 'label2', 'label3', 'label4', 'label5'], batch_size=32, shuffle=False)

valid_batches = ImageDataGenerator(preprocessing_function=tf.keras.applications.vgg16.preprocess_input) \
    .flow_from_directory(directory=valid_path, target_size=(224,224), classes=['label1', 'label2', 'label3', 'label4', 'label5'], batch_size=32)

```

Figure 6.12: Image PreProcessing

The batch size is the images' number that the model selects for the Deep Learning Process in each step. So, in each step, 32 images are processed.

After preprocessing, the images are ready to enter the model. The final result of this process can be seen in the confusion matrix in figure 6.13.

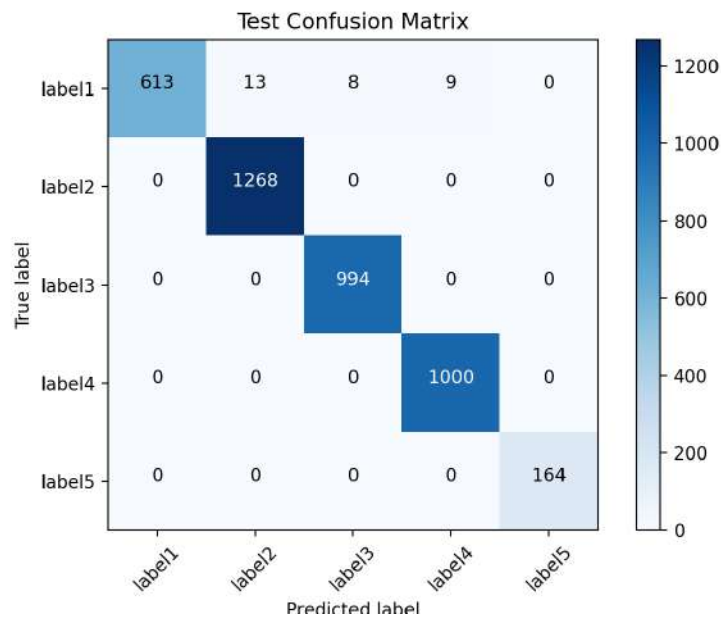


Figure 6.13: Confusion Matrix

The rows of the confusion matrix show True Label, and the columns show the Predicted Label. For example, in Label1, 613 images are correctly predicted as Label1, 13 images are incorrectly labeled as Label2, eight images are incorrectly labeled as Label3, nine images are incorrectly labeled as Label4.

For the other labels, there is no error in prediction. This matrix's accuracy,

which shows the test set's prediction accuracy, is equal to dividing the diagonal elements' sum by the sum of all the elements. By calculating the accuracy, it can be seen that the accuracy on the test set or the accuracy of the model is 99.26%.

Chapter 7

Forecasting

This chapter focuses on the Forecasting process in three different time horizons, including 5, 10, and 15 minutes. First, the dataset-making procedure has been introduced. Then, the model used for forecasting has been explained. Finally, the results of the forecasting part, such as the accuracy, the best day, the worst day, etc are introduced.

7.1 Dataset Preparation Procedure

The forecasting method that is used in this research work is based on the value of CSI. As introduced in the previous chapter, the images were split into five different labels concerning the CSI value. Since the value of CSI is available for every minute, by replacing the next value of CSI (the next value with respect to the time horizon) with the current value and running the Deep Learning model, forecasting accuracy is achieved. To be more precise, the value of $CSI(t+dt)$ is replaced with the value of $CSI(t)$, which "dt" is the time horizon.

In figure 7.1, an example of how making the dataset for 5 Minutes forecasting is shown.

	dates	files	labels_CSI		dates	files	labels_CSI
0	9/18/2019 10:25	img_20190918112500.jpg	3	0	9/18/2019 10:25	img_20190918112500.jpg	3
1	9/18/2019 10:26	img_20190918112600.jpg	3	1	9/18/2019 10:26	img_20190918112600.jpg	3
2	9/18/2019 10:27	img_20190918112700.jpg	3	2	9/18/2019 10:27	img_20190918112700.jpg	3
3	9/18/2019 10:28	img_20190918112800.jpg	5	3	9/18/2019 10:28	img_20190918112800.jpg	4
4	9/18/2019 10:29	img_20190918112900.jpg	4	4	9/18/2019 10:29	img_20190918112900.jpg	3
5	9/18/2019 10:30	img_20190918113000.jpg	3	5	9/18/2019 10:30	img_20190918113000.jpg	3
6	9/18/2019 10:31	img_20190918113100.jpg	3				
7	9/18/2019 10:32	img_20190918113200.jpg	3				
8	9/18/2019 10:33	img_20190918113300.jpg	4				
9	9/18/2019 10:34	img_20190918113400.jpg	3				
10	9/18/2019 10:35	img_20190918113500.jpg	3				




Figure 7.1: 5 Minutes Forecasting Dataset Preparation. As can be seen, the first five rows are replaced by the next five rows. So the new dataset shows the CSI values for the next five minutes.

As shown in the figure, the first five rows of the dataset are replaced by the next five rows. So, the label of the image at 10:25 is replaced by that of at 10:30. This process is done for all days separately to make sure that replacing the labels are done correctly. Then, with respect to the new labels, the images are split into five labels. This process is repeated for 10 and 15 Minutes forecasting as well. The next step is to run the Deep Learning model. Figure 7.2 is another example of dataset preparation of 10 Minutes forecasting.

	dates	files	labels_CSI		dates	files	labels_CSI	
	0	9/18/2019 10:25	img_20190918112500.jpg	3	0	9/18/2019 10:25	img_20190918112500.jpg	3
	1	9/18/2019 10:26	img_20190918112600.jpg	3	1	9/18/2019 10:26	img_20190918112600.jpg	3
	2	9/18/2019 10:27	img_20190918112700.jpg	3	2	9/18/2019 10:27	img_20190918112700.jpg	3
	3	9/18/2019 10:28	img_20190918112800.jpg	5	3	9/18/2019 10:28	img_20190918112800.jpg	3
	4	9/18/2019 10:29	img_20190918112900.jpg	4	4	9/18/2019 10:29	img_20190918112900.jpg	3
	5	9/18/2019 10:30	img_20190918113000.jpg	3	5	9/18/2019 10:30	img_20190918113000.jpg	3
	6	9/18/2019 10:31	img_20190918113100.jpg	3	6	9/18/2019 10:31	img_20190918113100.jpg	3
	7	9/18/2019 10:32	img_20190918113200.jpg	3	7	9/18/2019 10:32	img_20190918113200.jpg	3
	8	9/18/2019 10:33	img_20190918113300.jpg	4	8	9/18/2019 10:33	img_20190918113300.jpg	3
	9	9/18/2019 10:34	img_20190918113400.jpg	3	9	9/18/2019 10:34	img_20190918113400.jpg	3
	10	9/18/2019 10:35	img_20190918113500.jpg	3	10	9/18/2019 10:35	img_20190918113500.jpg	4
	11	9/18/2019 10:36	img_20190918113600.jpg	3				
	12	9/18/2019 10:37	img_20190918113700.jpg	3				
	13	9/18/2019 10:38	img_20190918113800.jpg	3				
	14	9/18/2019 10:39	img_20190918113900.jpg	3				
	15	9/18/2019 10:40	img_20190918114000.jpg	3				
	16	9/18/2019 10:41	img_20190918114100.jpg	3				
	17	9/18/2019 10:42	img_20190918114200.jpg	3				
	18	9/18/2019 10:43	img_20190918114300.jpg	3				
	19	9/18/2019 10:44	img_20190918114400.jpg	3				
	20	9/18/2019 10:45	img_20190918114500.jpg	4				

Figure 7.2: 10 Minutes Forecasting Dataset Preparation. As can be seen, the first ten rows are replaced by the next ten rows. So the new dataset shows the CSI values for the next ten minutes.

7.2 Forecasting Model

In this step, the forecasting model has been run on our image dataset for our three different forecasting time horizons. We have tried ten different and independent trials on each time horizon to verify the robustness and reliability of the Forecasting model. In figures 7.3 and 7.4, the first two confusion matrices for 5' forecasting can be seen. Figures 7.5 and 7.6, represent the first two confusion matrices for 10' forecasting. In figures 7.7 and 7.8, the first two confusion matrices for 15' forecasting are shown. The confusion matrices for the other trials are represented in Appendix C.

Confusion matrix

Predicted	label1	5 0.09%	83 1.46%	1 0.02%			89 5.62%	94.38%
	label2	485 8.54%	1010 17.79%	327 5.76%		17 0.30%	1839 94.92%	45.08%
	label3		27 0.48%	71 1.25%		4 0.07%	102 65.61%	30.39%
	label4	155 2.73%	141 2.48%	606 10.67%	2605 45.89%	140 2.47%	3647 71.43%	28.57%
	label5						0 0.00%	0.00%
	sum_col	645 0.78%	1261 66.10%	1005 7.06%	2605 100%	161 0.90%	5677 65.02%	99.22%
		label1	label2	label3	label4	label5	sum_row	
								Actual

Figure 7.3: First confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the First trial and is 65.02%.

Predicted	Actual label1	Actual label2	Actual label3	Actual label4	Actual label5	sum_row
label1	180 3.17%	239 4.21%	90 1.59%	0 0.00%	0 0.00%	509 8.96% 64.42%
label2	100 1.76%	606 10.67%	155 2.73%	0 0.00%	3 0.05%	864 15.35% 28.86%
label3	0 0.00%	90 1.59%	50 0.88%	0 0.00%	4 0.07%	144 2.55% 65.28%
label4	365 6.43%	326 5.74%	710 12.51%	2605 45.89%	154 2.71%	4160 73.87% 27.38%
label5	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
sum_col	645 11.41% 12.88%	1261 22.28% 51.84%	1005 17.93% 45.52%	2605 46.11% 60.61%	161 2.87% 39.39%	5677
	label1	label2	label3	label4	label5	sum_col
	Actual					

Figure 7.4: Second confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Second trial and is 60.61%.

Confusion matrix

Predicted	label1	197 3.55%	194 3.50%	67 1.21%	2 0.04%	460 82.33%	37.17%	
	label2	372 6.71%	774 13.95%	226 4.07%	37 0.67%	1409 25.37%	45.37%	
	label3	2 0.04%	185 3.33%	109 1.96%	9 0.16%	305 5.50%	84.20%	
	label4	50 0.90%	74 1.33%	564 10.17%	2570 46.32%	3368 60.31%	23.89%	
	label5		1 0.02%	5 0.09%		6 0.11%	100.00%	
	sum_col	621 11.14%	1228 22.40%	971 17.54%	2570 46.60%	158 0.28%	5548 65.79%	34.21%
		label1	label2	label3	label4	label5	sum__lin	
		Actual						

Figure 7.5: First confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the First trial and is 65.79%.

Confusion matrix

Predicted	label1	70 1.26%	69 1.24%	2 0.04%			141 0.40%	88.35%	
	label2	395 7.12%	731 13.18%	149 2.69%		34 0.61%	1309 36.84%	44.16%	
	label3		136 2.45%	65 1.17%		7 0.13%	208 5.83%	68.75%	
	label4	156 2.81%	292 5.26%	755 13.61%	2570 46.32%	117 2.11%	3890 107.37%	33.99%	
	label5						0 0.00%	0.00%	
	sum_col	621 11.27%	1228 22.44%	971 17.68%	2570 46.80%	158 2.83%	5548 155.42%	61.93%	
		label1	label2	label3	label4	label5	sum_lim		
		88.71%	40.47%	93.31%	0.00%	100.00%	38.07%		
		Actual							

Figure 7.6: Second confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Second trial and is 61.93%.

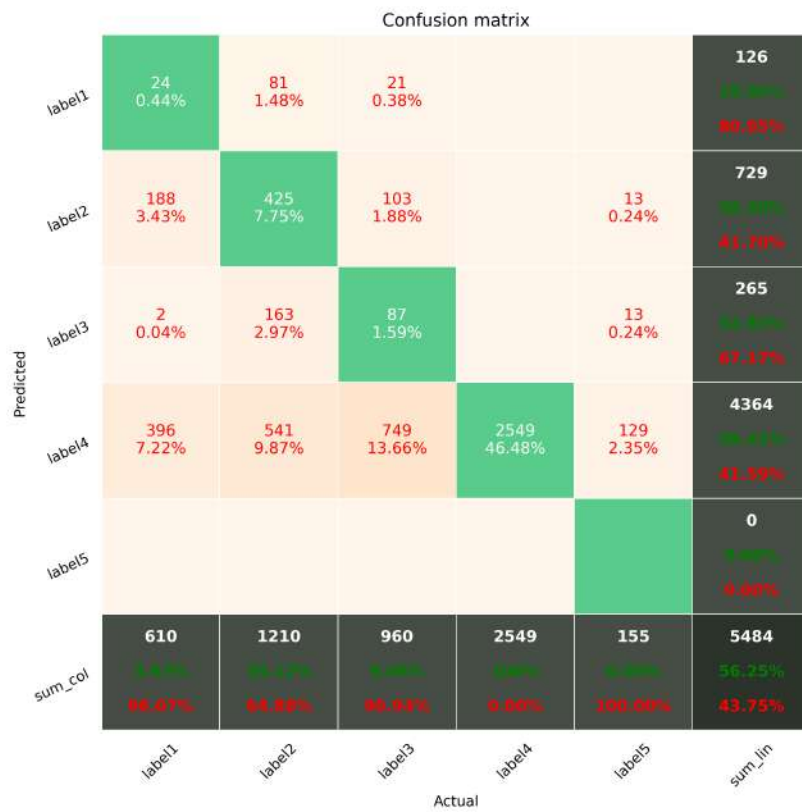


Figure 7.7: First confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the First trial and is 56.25%.

Confusion matrix

Predicted	label1	20 0.36%	130 2.37%	35 0.64%		185 36.41% 88.19%	
	label2	238 4.34%	599 10.92%	165 3.01%		1024 20.48% 41.30%	
	label3		102 1.86%	72 1.31%		183 36.60% 60.40%	
	label4	352 6.42%	379 6.91%	688 12.55%	2549 46.48%	124 2.26%	4092 81.70% 37.71%
	label5						0 0.00% 0.00%
	sum_col	610 11.86% 86.72%	1210 23.80% 50.58%	960 18.82% 62.50%	2549 50.96% 0.00%	155 3.08% 100.00%	5484 109.68% 40.92%
		label1	label2	label3	label4	label5	sum_lim
		Actual					

Figure 7.8: Second confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Second trial and is 59.08%.

7.3 Forecasting Results

In figures 7.9 to 7.11, the accuracy of the confusion matrices for each time horizon and for all trials are plotted with the average value. As can be seen in these figures, the forecasting accuracies for 5, 10, and 15 Minutes forecasting are 62.7%, 65.48%, and 58.913% respectively.

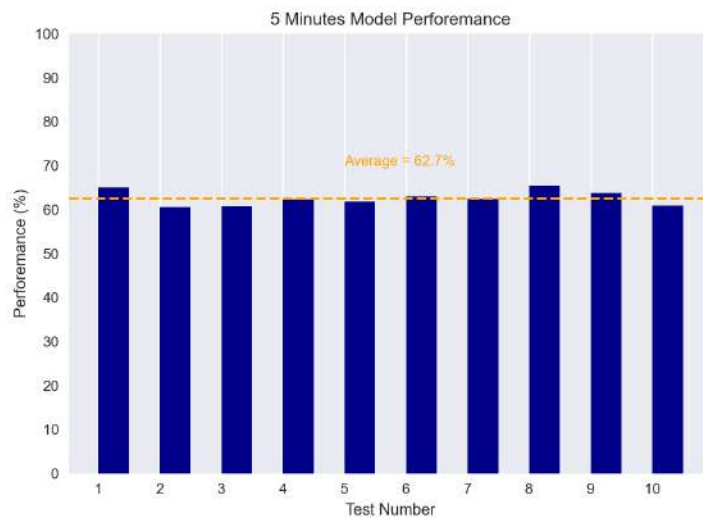


Figure 7.9: Model Accuracy in 10 different and independent trials for 5 Minutes Forecasting. The average forecasting accuracy for this time horizon is 62.7%.

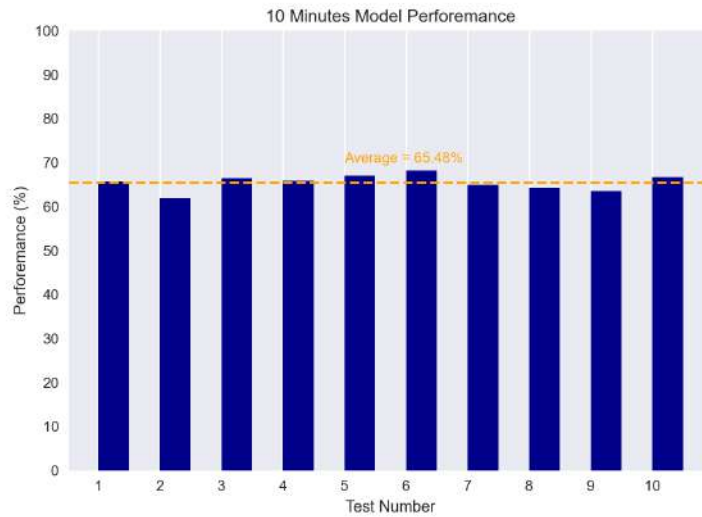


Figure 7.10: Model Accuracy in 10 different and independent trials for 10 Minutes Forecasting. The average forecasting accuracy for this time horizon is 65.48%.

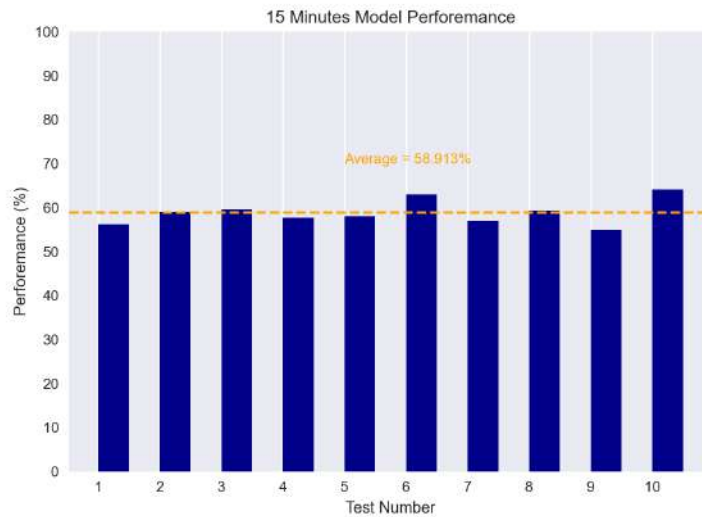


Figure 7.11: Model Accuracy in 10 different and independent trials for 15 Minutes Forecasting. The average forecasting accuracy for this time horizon is 58.9%.

Figures 7.12 to 7.17 represent thirty six samples of correctly classified and misclassified samples for both the best and worst days through all time horizons. As can be seen in these figures, the correctly classified values are plotted in blue, and the misclassified values are plotted in red. Also, the label of each classified samples are separated.

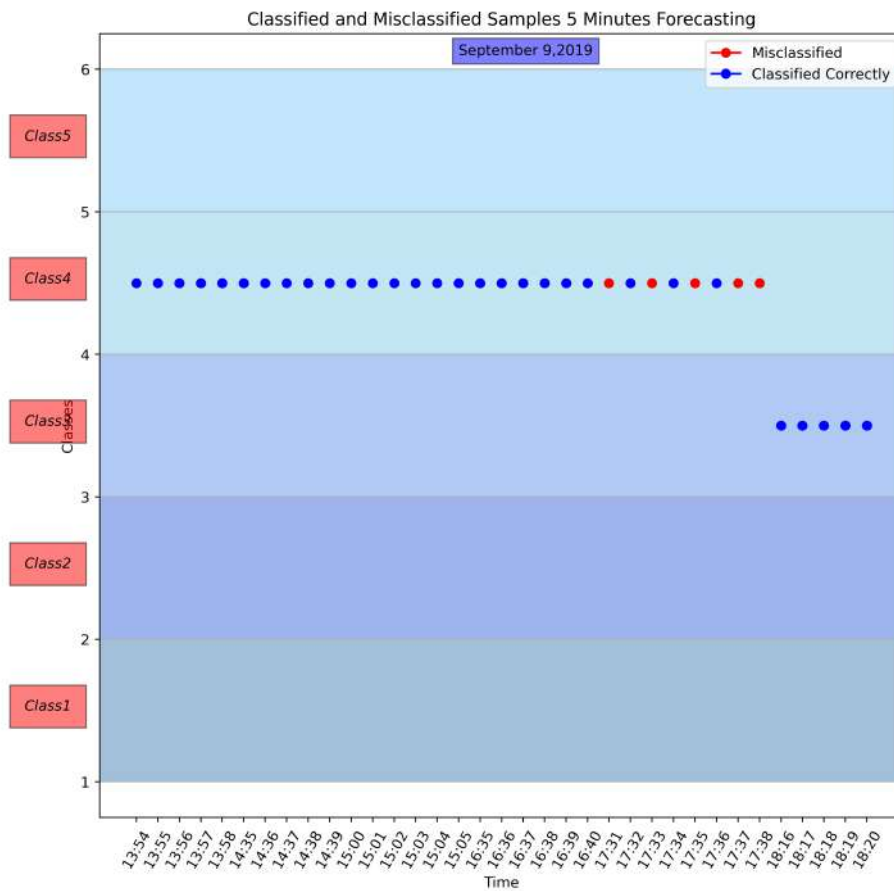


Figure 7.12: Classified and Misclassified samples for one of the best day in 5 Minutes Forecasting in September 9,2019. The Correctly Classified images are represented by Blue dots, whereas the misclassified samples are represented by Red dots. The horizontal axis of this figure is divided to five sections representing each label.

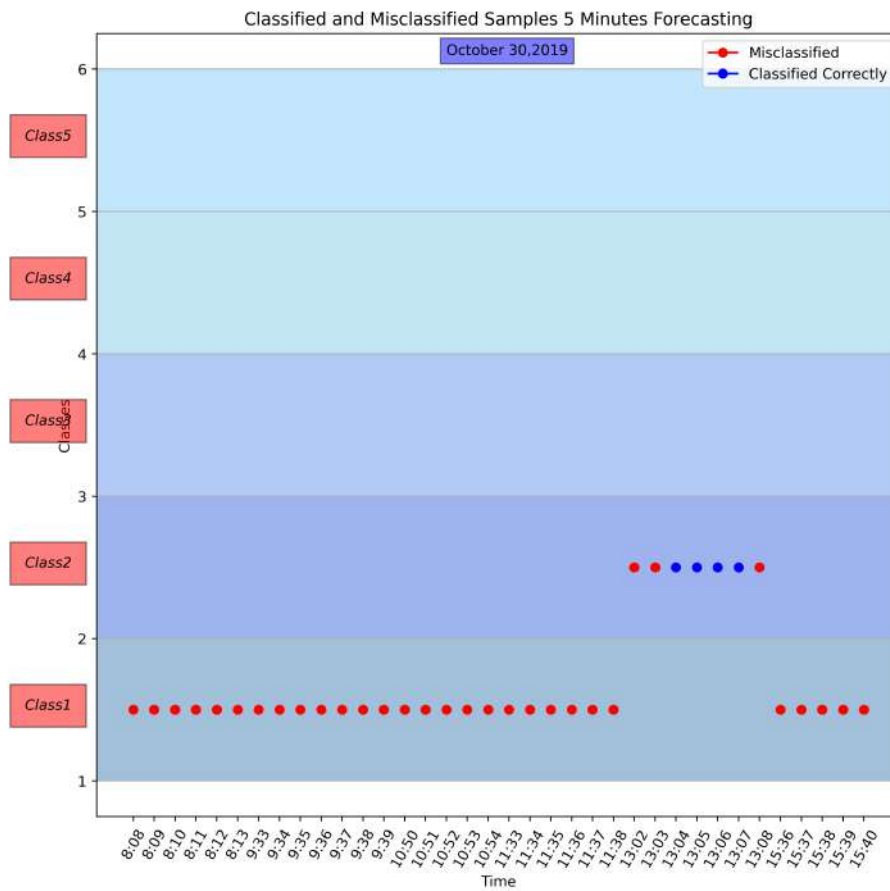


Figure 7.13: Classified and Misclassified samples for one of the worst day in 5 Minutes Forecasting in December 30,2019. The Correctly Classified images are represented by Blue dots, whereas the misclassified samples are represented by Red dots. The horizontal axis of this figure is divided to five sections representing each label.

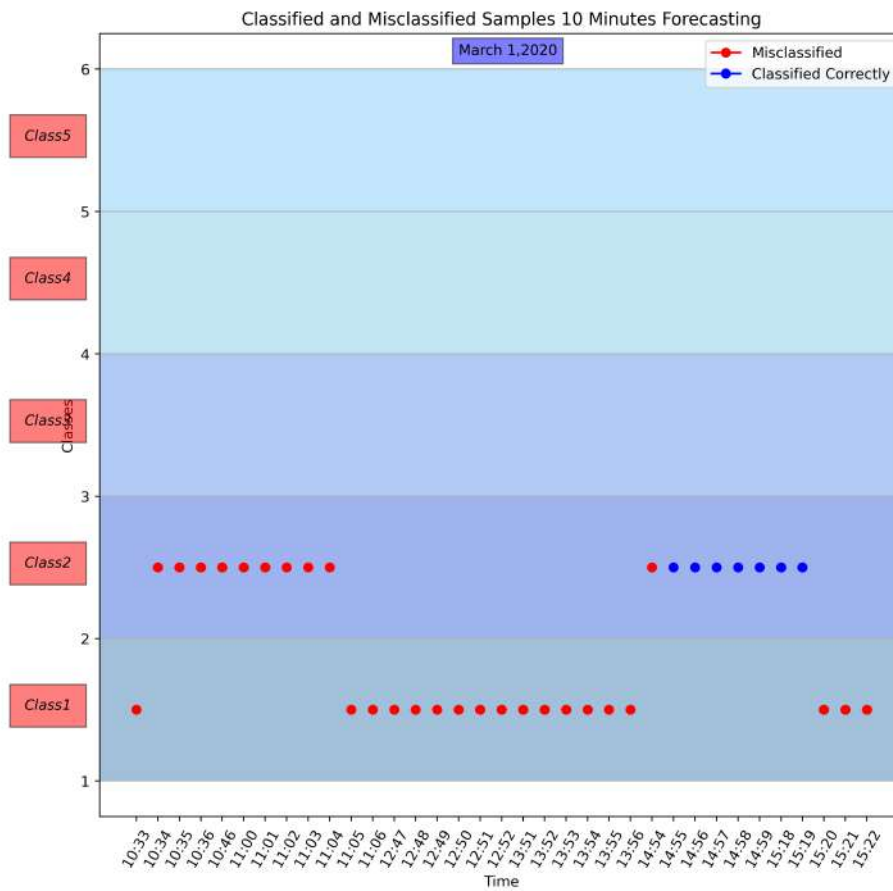


Figure 7.15: Classified and Misclassified samples for one of the worst day in 10 Minutes Forecasting in March 1,2020. The Correctly Classified images are represented by Blue dots, whereas the misclassified samples are represented by Red dots. The horizontal axis of this figure is divided to five sections representing each label.

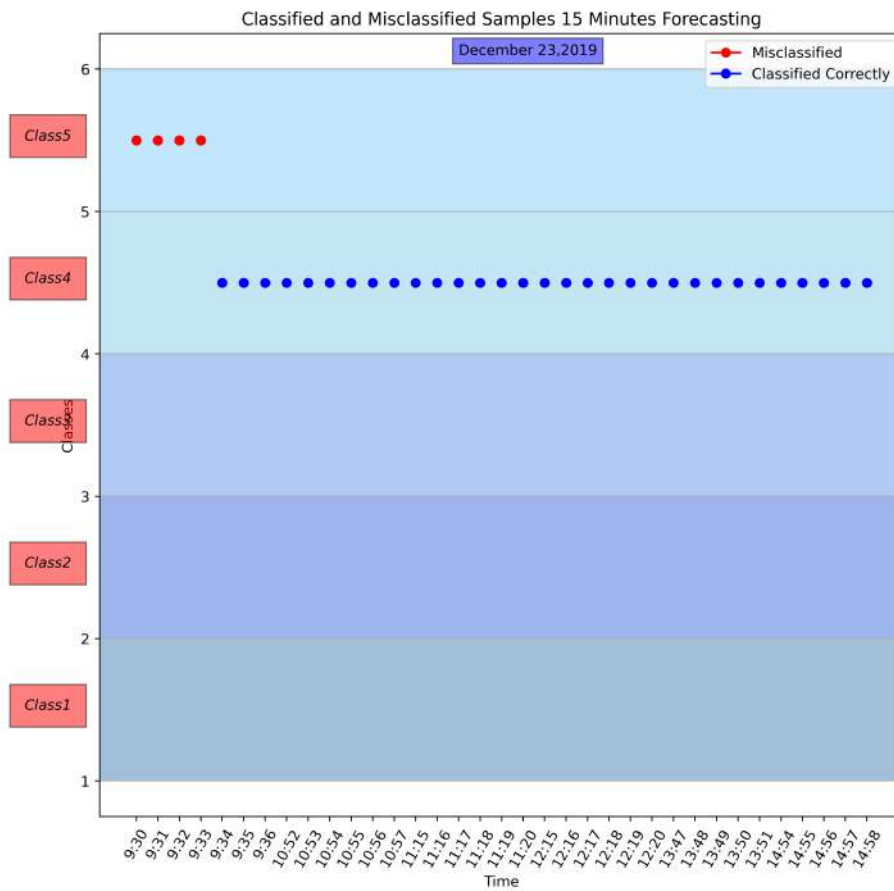


Figure 7.16: Classified and Misclassified samples for one of the best day in 15 Minutes Forecasting in December 23,2019. The Correctly Classified images are represented by Blue dots, whereas the misclassified samples are represented by Red dots. The horizontal axis of this figure is divided to five sections representing each label.

Figures 7.18 to 7.23 indicate the convergence curves of the Deep Learning model in each forecasting time horizon. In these graphs, we can see the prediction and classification accuracy in both Training Set and Validation Set.

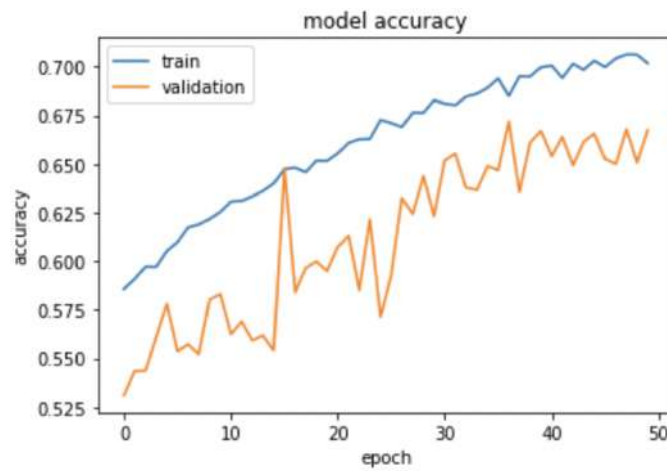


Figure 7.18: The accuracy of the model on Train set and Validation set through 50 epochs for 5 Minutes Forecasting. The blue line represents the performance of the train set in forecasting. While the orange line shows the validation set performance for each epoch.

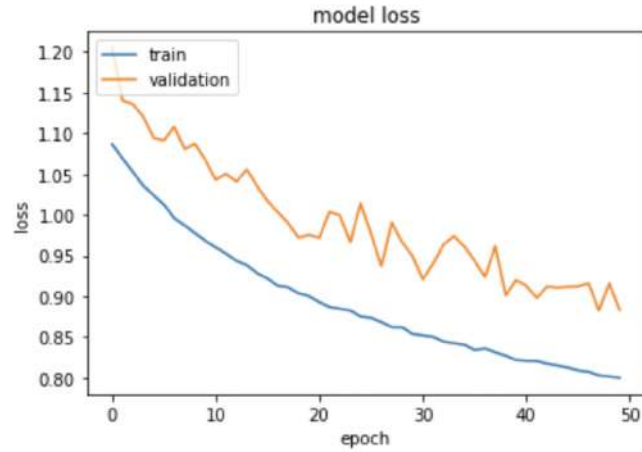


Figure 7.19: The Loss of the model on Train set and Validation set through 50 epochs for 5 Minutes Forecasting. The blue line represents the performance of the train set in forecasting. While the orange line shows the validation set performance for each epoch.

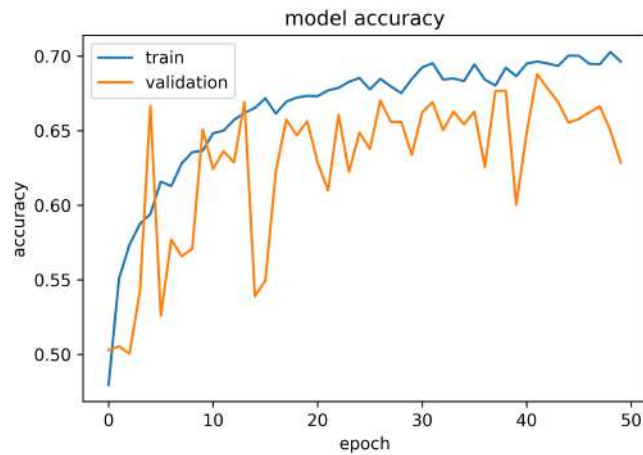


Figure 7.20: The accuracy of the model on Train set and Validation set through 50 epochs for 10 Minutes Forecasting. The blue line represents the performance of the train set in forecasting. While the orange line shows the validation set performance for each epoch.

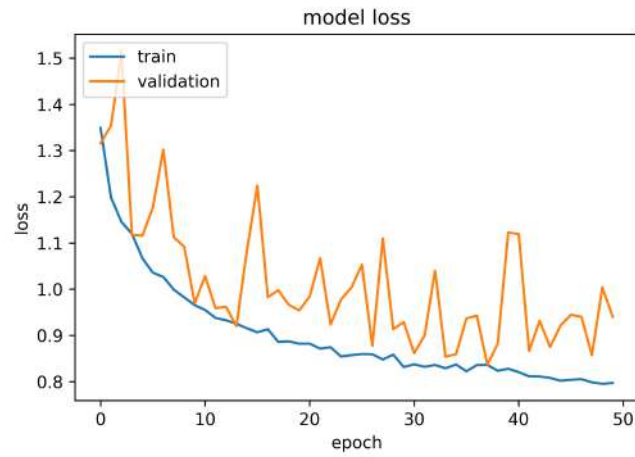


Figure 7.21: The Loss of the model on Train set and Validation set through 50 epochs for 10 Minutes Forecasting. The blue line represents the performance of the train set in forecasting. While the orange line shows the validation set performance for each epoch.

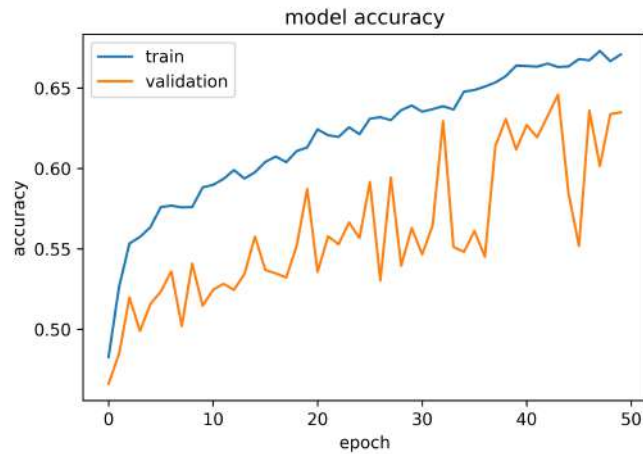


Figure 7.22: The accuracy of the model on Train set and Validation set through 50 epochs for 15 Minutes Forecasting. The blue line represents the performance of the train set in forecasting. While the orange line shows the validation set performance for each epoch.

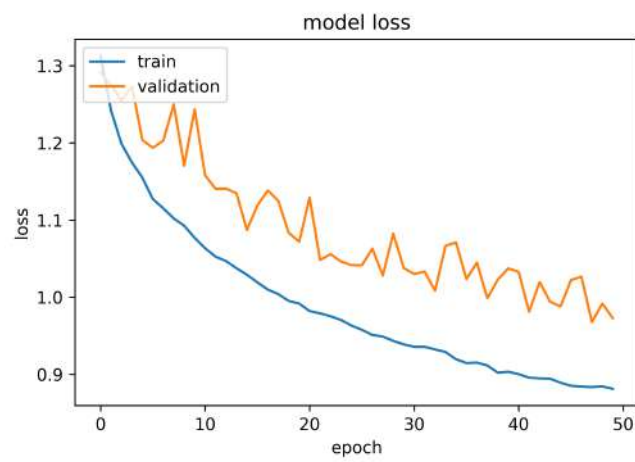


Figure 7.23: The Loss of the model on Train set and Validation set through 50 epochs for 15 Minutes Forecasting. The blue line represents the performance of the train set in forecasting. While the orange line shows the validation set performance for each epoch.

In figures 7.24 to 7.26, the sparsity curves of each time horizon are represented. These curves show how much the misclassified samples are different from the correctly classified samples based on a logarithmic scale. The process of Deep learning is that the model predicts the label presence probability for each image. The prediction result of each image is available in an numpy array. For example we have: [0.0007, 0.0090, 0.01, 0.9997, 0.009]

This array shows that this image belongs 99.97% to label4. Then the model validates this probability by checking the dataset. If the image exists in label 4, the image is correctly classified. Otherwise it is misclassified. In the sparsity curves, these probabilities are used. If the images are classified correctly, they were plotted as a linear ($y=x$) line. Otherwise, the probability of the label in which the image exists in that label in our dataset will multiply by 10000 and then be plotted with a logarithmic scale. The predictions are multiplied by 10000 since the probability values are minimal in this case.

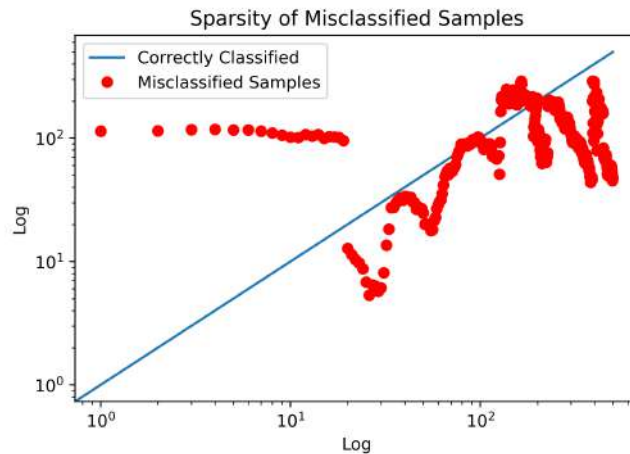


Figure 7.24: Sparsity Curve for 5 Minutes Forecasting: Red dots show the misclassified samples and Correctly classified samples are represented by blue linear line

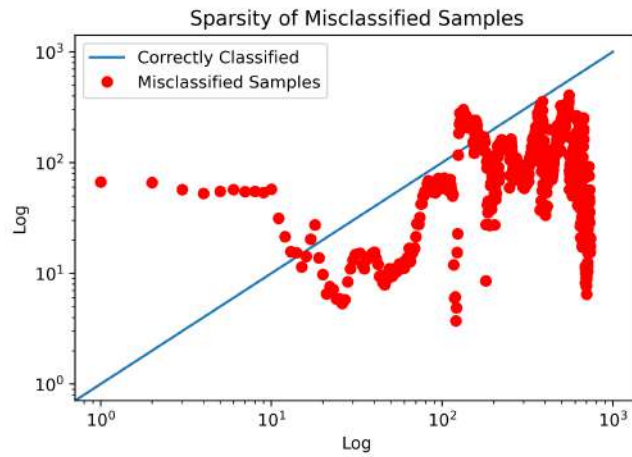


Figure 7.25: Sparsity Curve for 10 Minutes Forecasting: Red dots show the misclassified samples and Correctly classified samples are represented by blue linear line

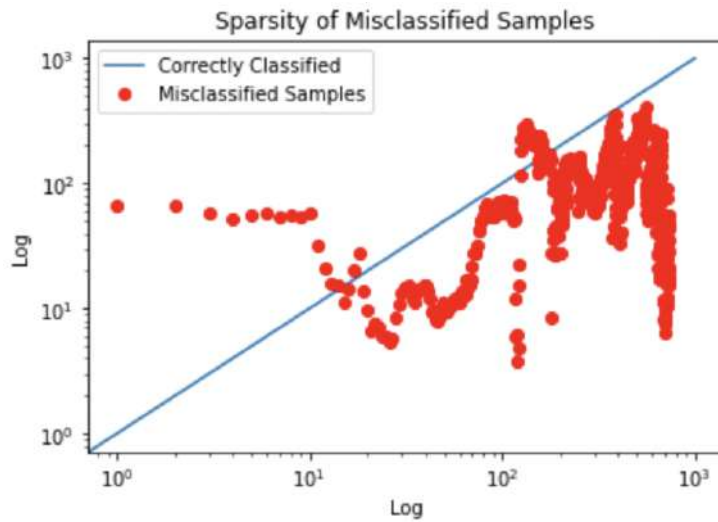


Figure 7.26: Sparsity Curve for 15 Minutes Forecasting: Red dots show the misclassified samples and Correctly classified samples are represented by blue linear line

As seen in these figures, the red dots indicate the misclassified samples, and the blue line indicates the correctly classified samples. Also, as seen in these figures, the misclassified samples are not placed at a long distance from the blue line, representing that the model was close to predicting the correct label for most of these misclassified samples. Whereas, for the other dots placed in a long distance of the blue line, it shows that the model was confused for the correct label of those images.

In figures 7.27 to 7.29, the sorted daily performance of the forecasting model for all days through the dataset is represented. In these figures, the performance is sorted from the best day to the worst day.

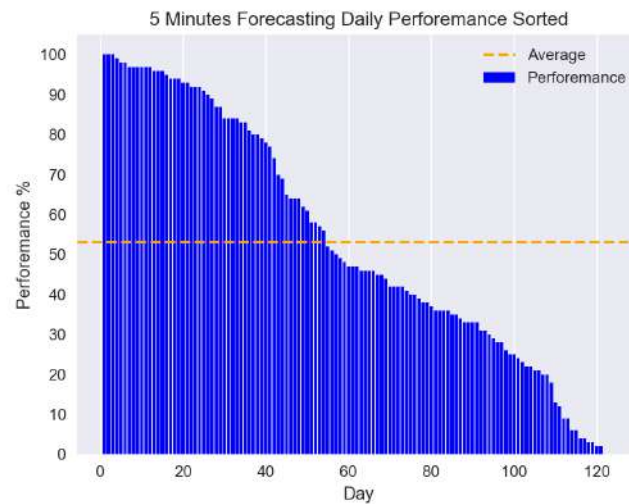


Figure 7.27: Sorted Prediction Performance in Descending order for 5 Minutes Forecasting

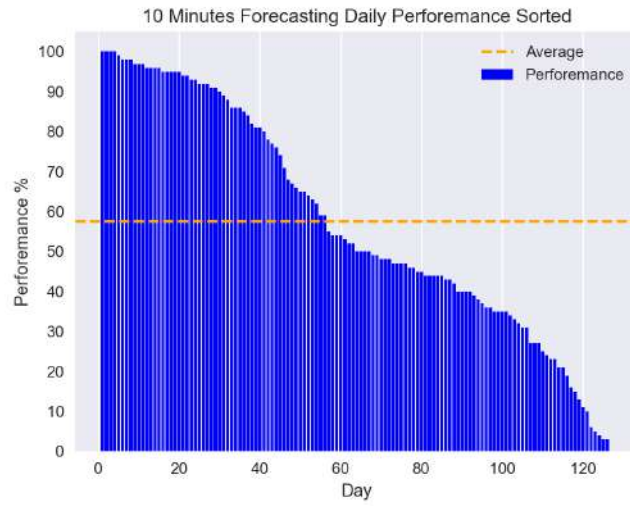


Figure 7.28: Sorted Prediction Performance in Descending order for 10 Minutes Forecasting

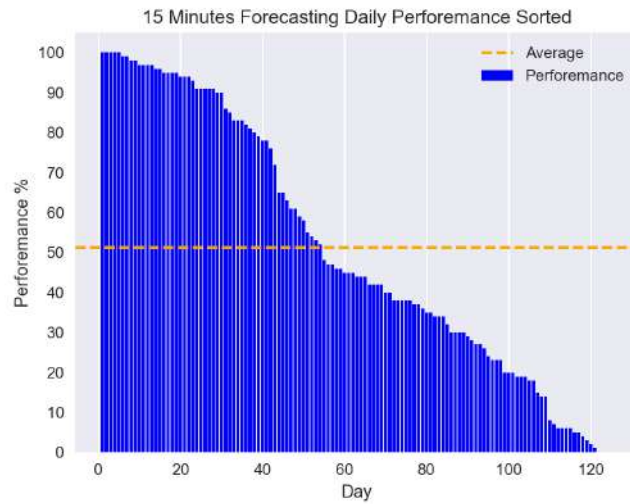


Figure 7.29: Sorted Prediction Performance in Descending order for 15 Minutes Forecasting

As can be seen in these figures, there are some days with 100% prediction accuracy and some days with 0% accuracy.

Chapter 8

Conclusion and Future Work

This thesis represents the inclusive study of Solar Energy Nowcasting employing Computer Vision to identify the advantages and drawbacks of this method and the benefits to the other possible methods.

This method is based on implementing Computer vision or in a specific way Deep Learning on the Images dataset and with respect to the CSI value to first classify the images and then forecast the CSI value for the next 5', 10', and 15'.

The benefits of this method with respect to the other possible methods are speed and accuracy. Through Computer Vision and, in a specific case Deep Learning, the predictions can be much faster than the NWP or the other traditional methods.

As mentioned, the images captured from the sky using ASI play a vital role in this study. These images are needed to be processed by different Image Processing and visualizing methods to be provided for the Deep Learning model. The model has to process various features of the images like the edges, the colors, the difference in colors by area, and the different shapes to understand the difference between the images and group them based on a reference value. Various Image processing and Image visualizing tests were performed to find the best-fitted images for the Deep Learning model. Next, grouping the images are done based on CSI, which is a function of GHI. Since the output power of solar panels is directly proportional to Irradiation, this classification, the times with the most and least Irradiation can be identified.

Also, by predicting the changes in Irradiance, the clouds movements, and the weather conditions, the future changes of Irradiation can be forecasted, and the generated power by solar panels can be maximized.

By the forecasted results in this research work, the solar panels can be tilted at an angle that can receive the most Irradiation. The new method now being used is to use a rotating motor to tilt the panel to the optimum angle by the data obtained from forecasting.

Chapter 9

Appendix: Irradiation Data

In this appendix the figures related to the chapter of Irradiation data are represented. Figures 9.1 to 9.6 indicate the identified ramps in Minutely scales from November 2019 to April 2020.

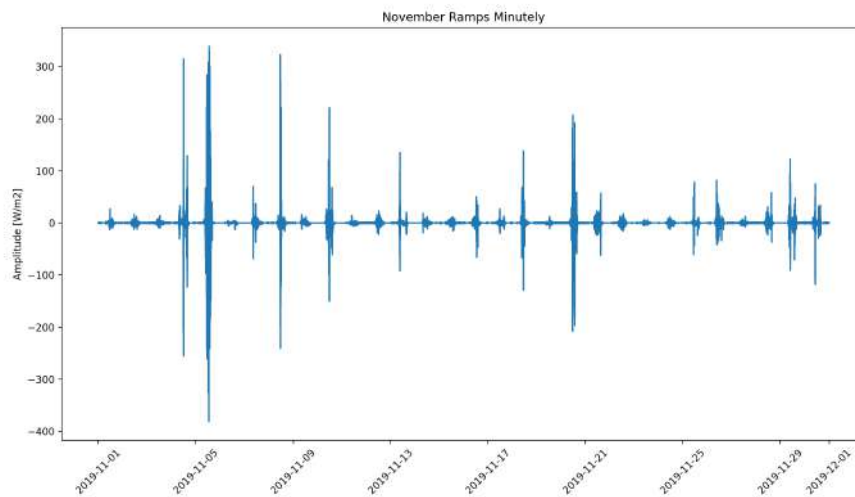


Figure 9.1: November Ramps Minutely

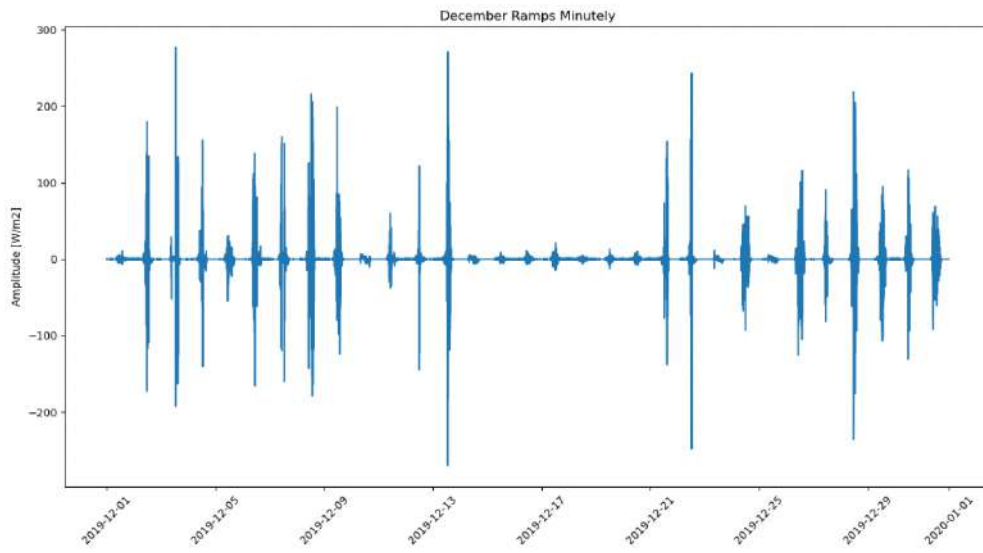


Figure 9.2: December Ramps Minutely

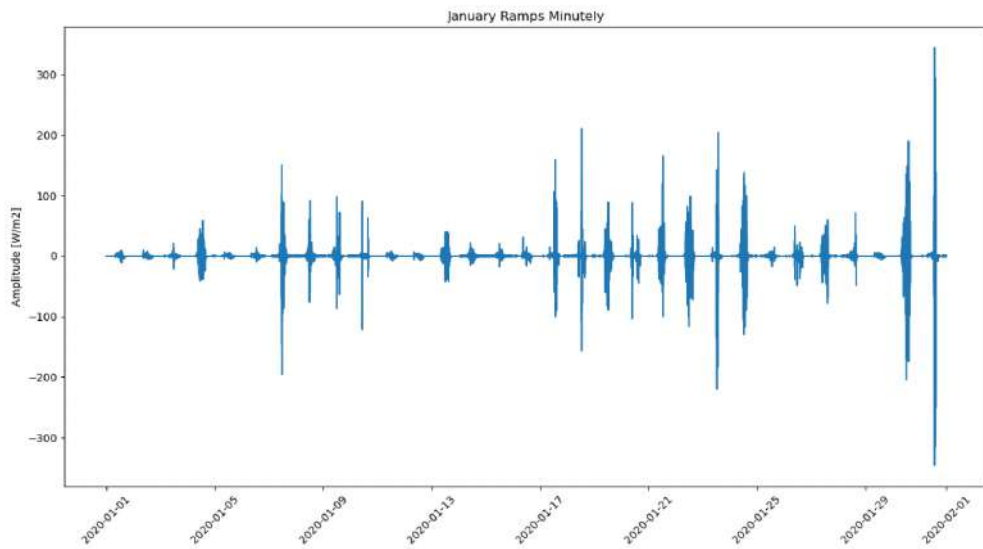


Figure 9.3: January Ramps Minutely

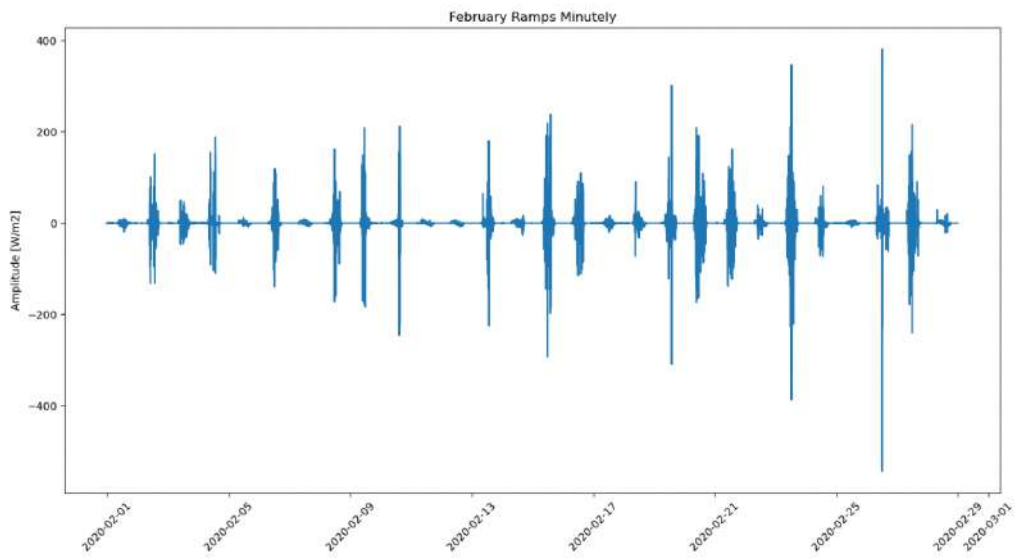


Figure 9.4: February Ramps Minutely

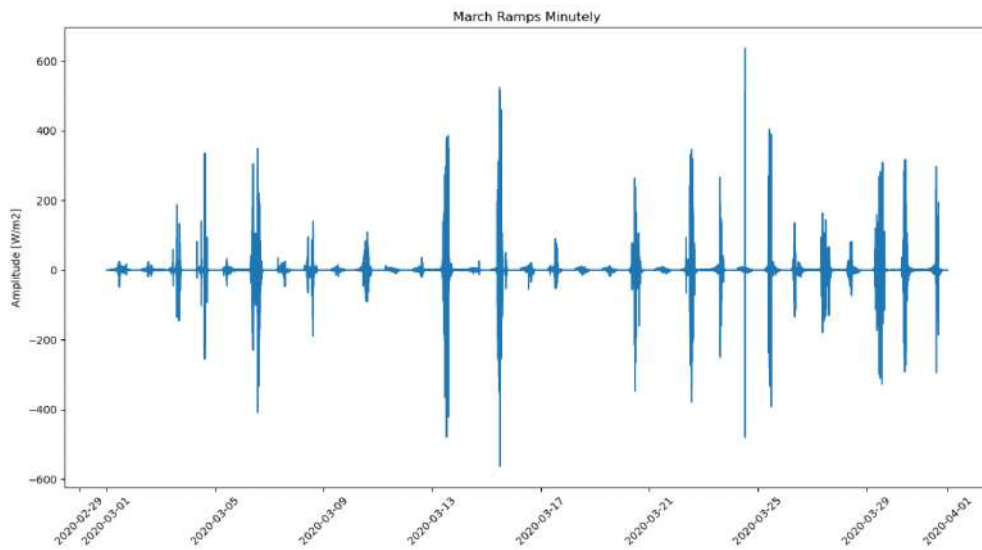


Figure 9.5: March Ramps Minutely

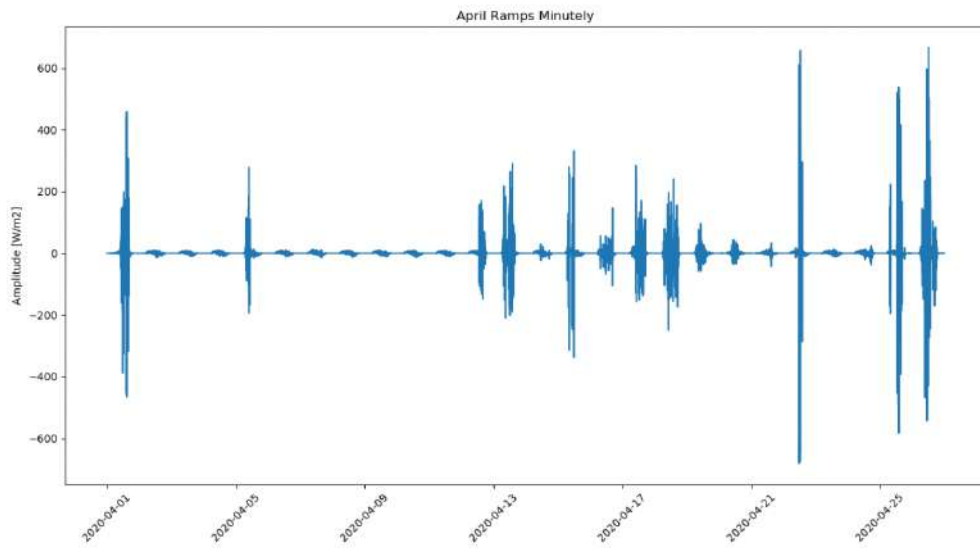


Figure 9.6: April Ramps Minutely

In figures 9.7 to 9.30, the ramp identification images for 5', 10', 15', and 20' time horizons from November to April are represented respectively.

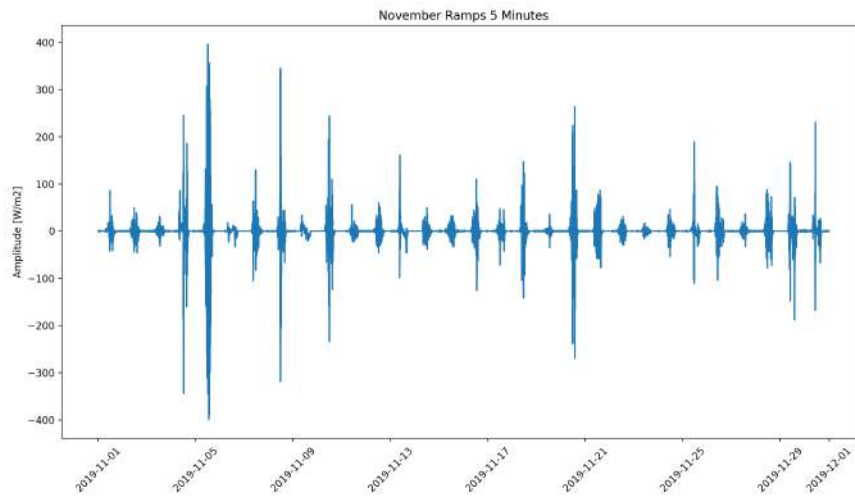


Figure 9.7: November Ramps 5 Minutes Time Horizons

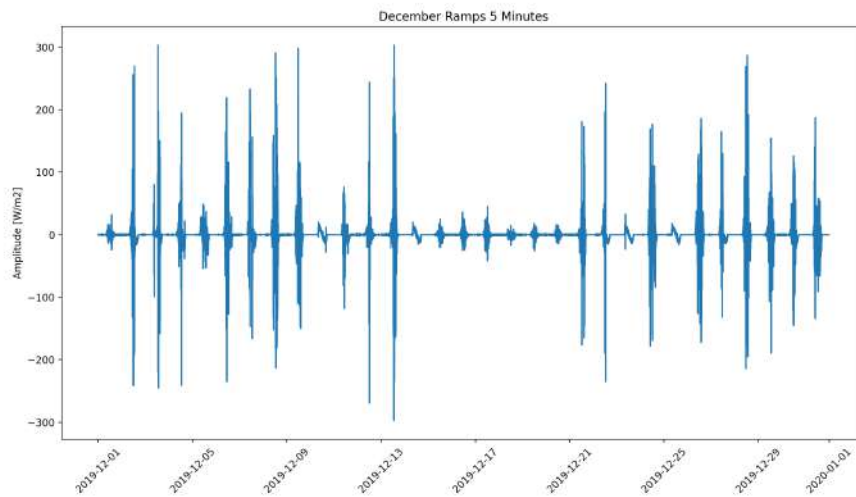


Figure 9.8: December Ramps 5 Minutes Time Horizons

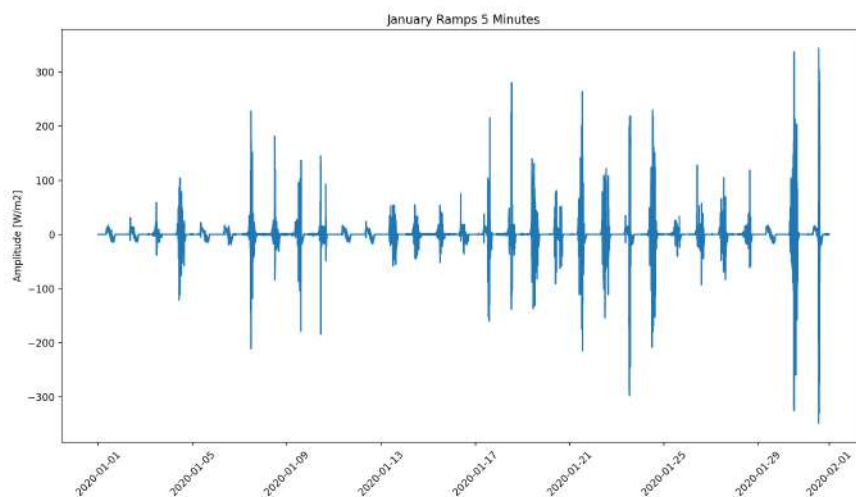


Figure 9.9: January Ramps 5 Minutes Time Horizons

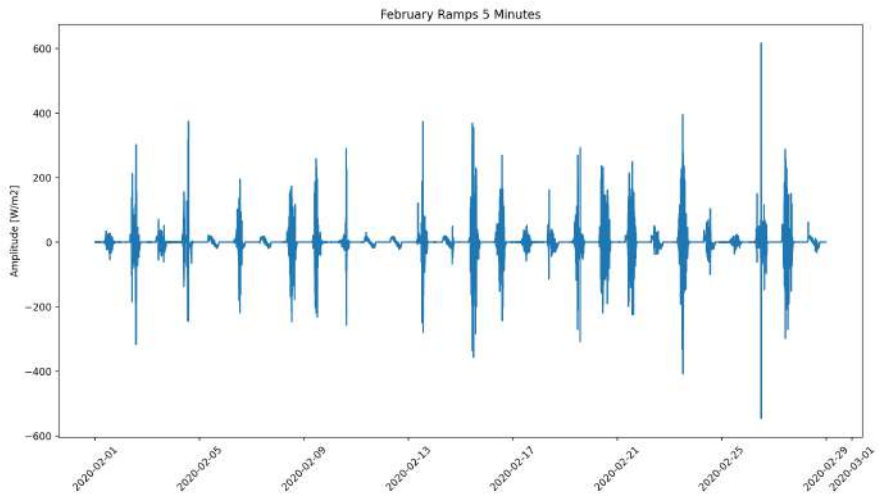


Figure 9.10: February Ramps 5 Minutes Time Horizons

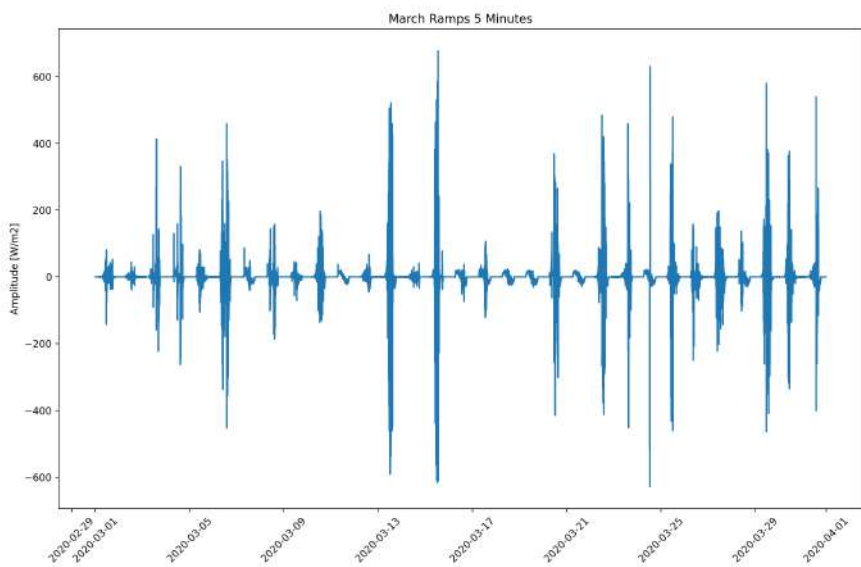


Figure 9.11: March Ramps 5 Minutes Time Horizons

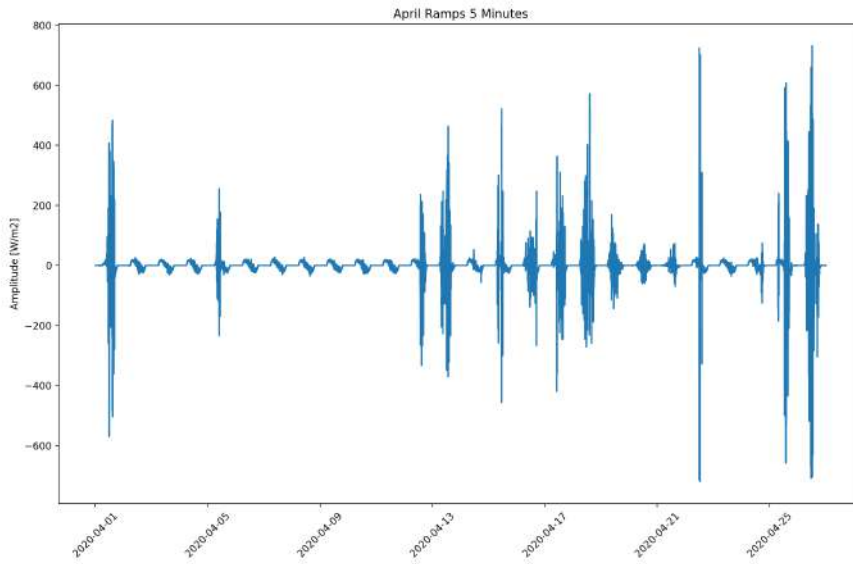


Figure 9.12: April Ramps 5 Minutes Time Horizons

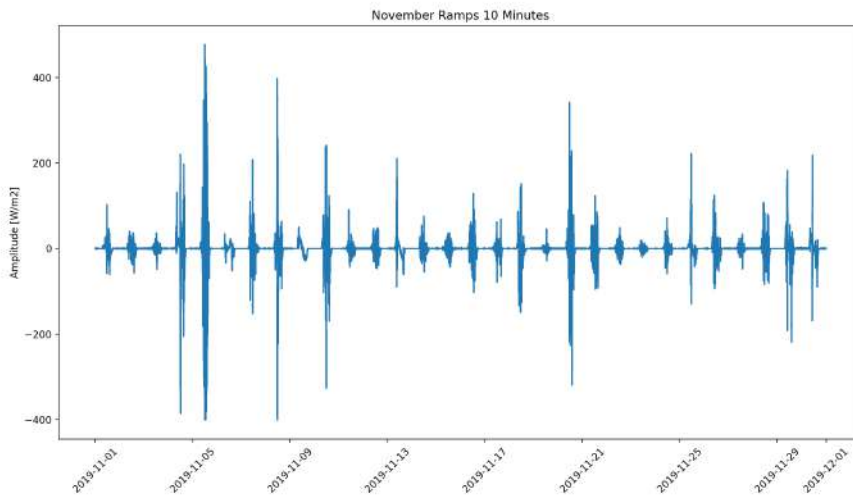


Figure 9.13: November Ramps 10 Minutes Time Horizons

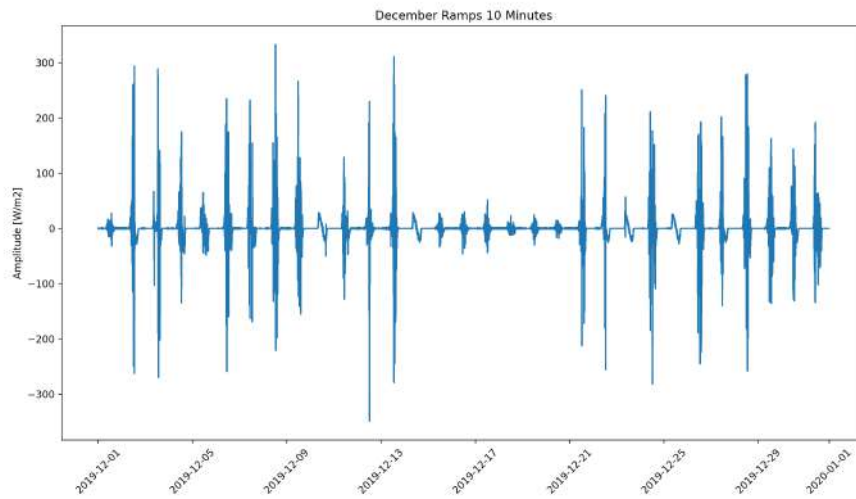


Figure 9.14: December Ramps 10 Minutes Time Horizons

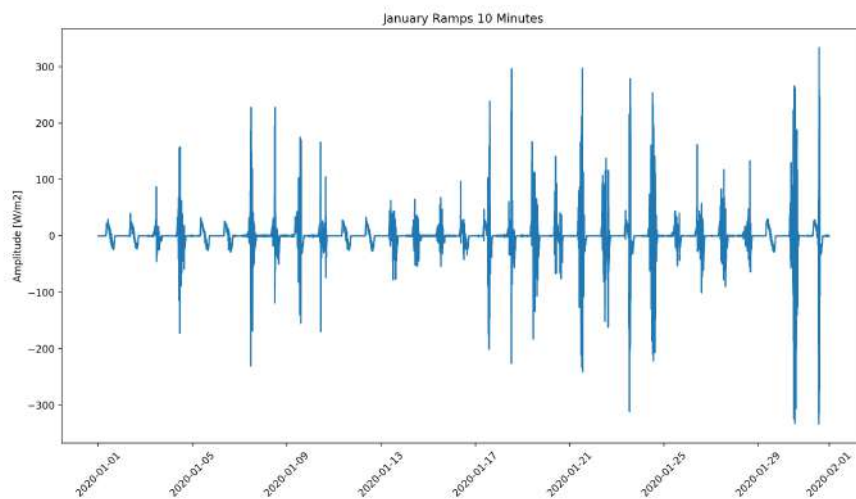


Figure 9.15: January Ramps 10 Minutes Time Horizons

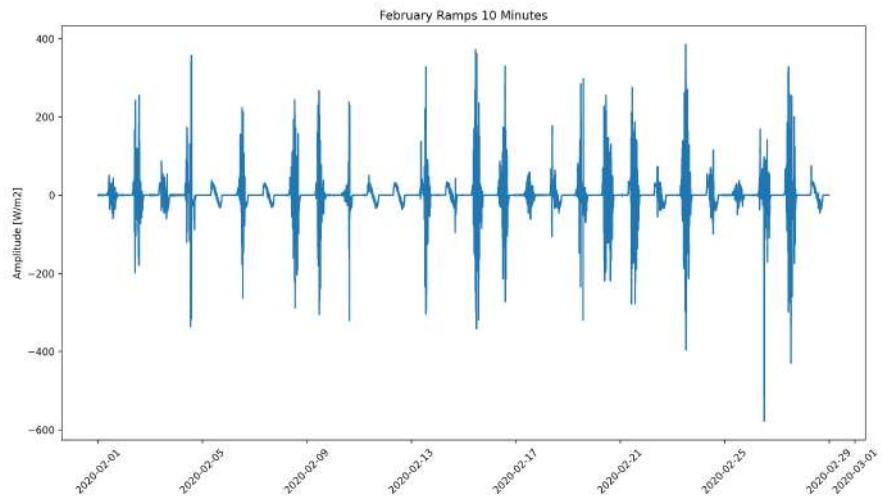


Figure 9.16: February Ramps 10 Minutes Time Horizons

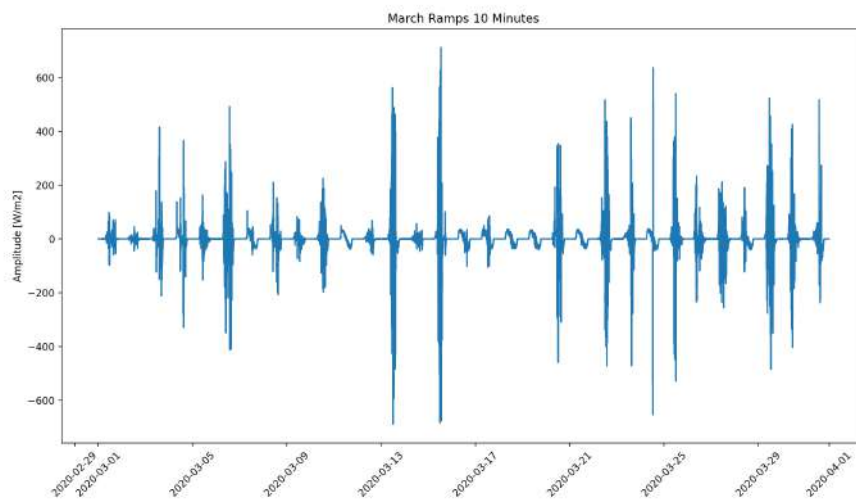


Figure 9.17: March Ramps 10 Minutes Time Horizons

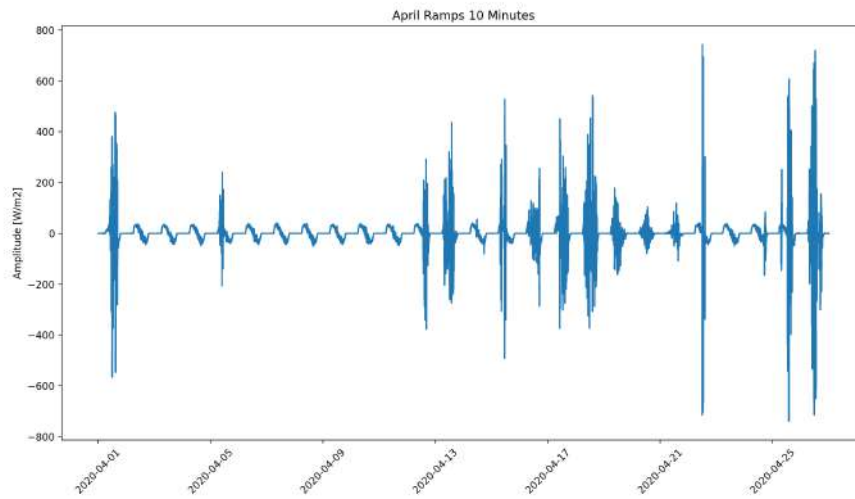


Figure 9.18: April Ramps 10 Minutes Time Horizons

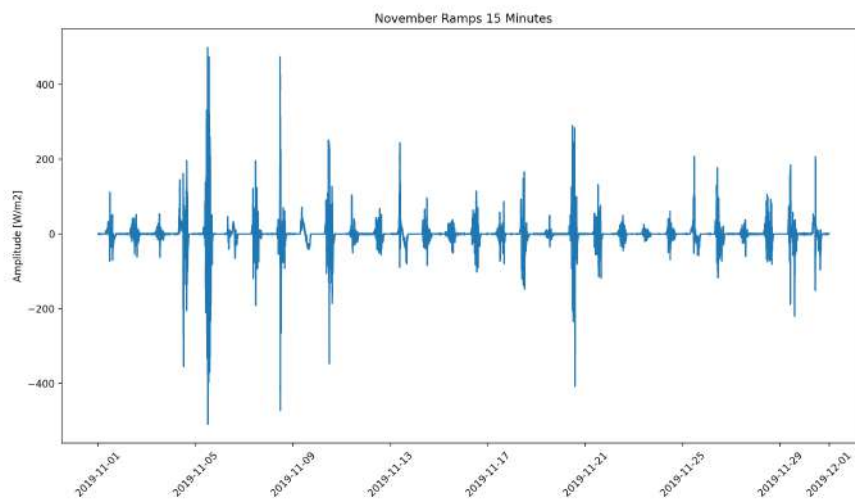


Figure 9.19: November Ramps 15 Minutes Time Horizons

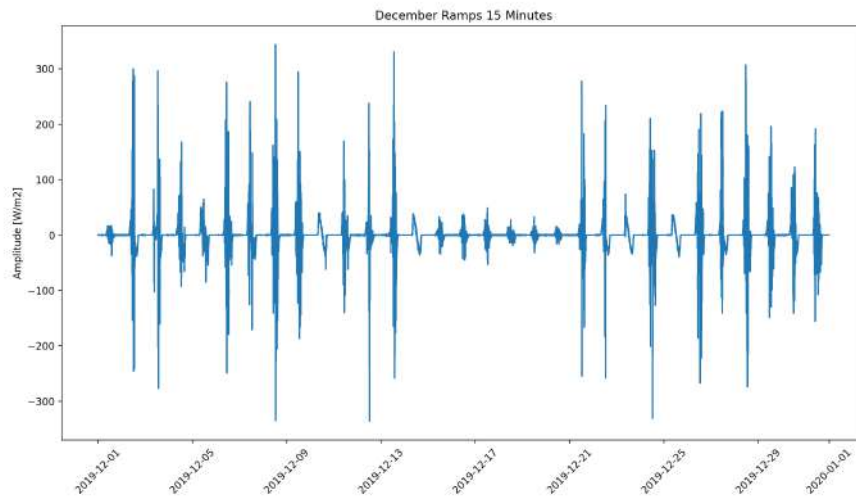


Figure 9.20: December Ramps 15 Minutes Time Horizons

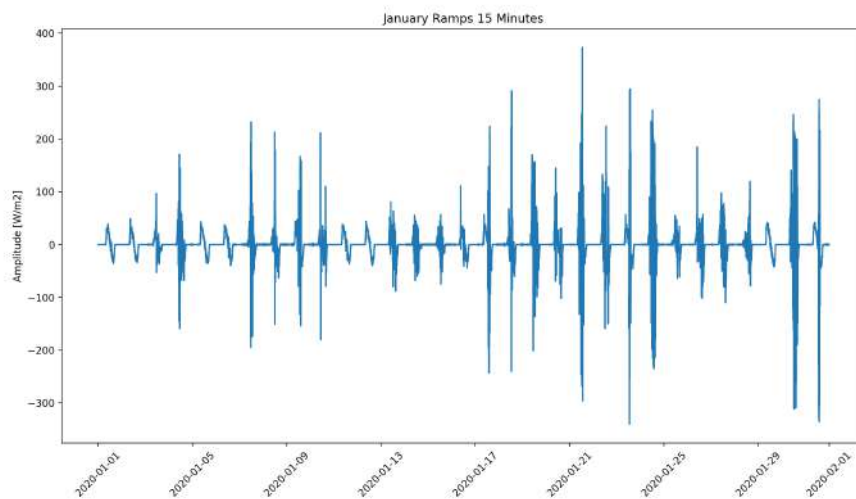


Figure 9.21: January Ramps 15 Minutes Time Horizons

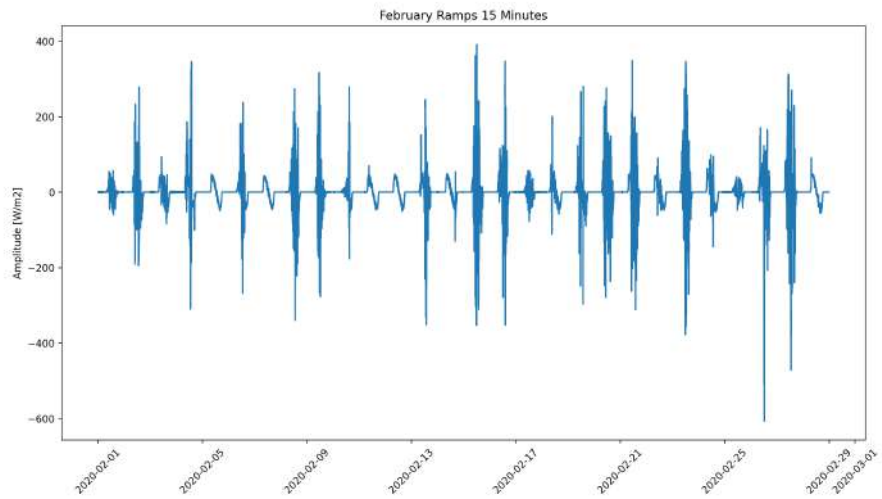


Figure 9.22: February Ramps 15 Minutes Time Horizons

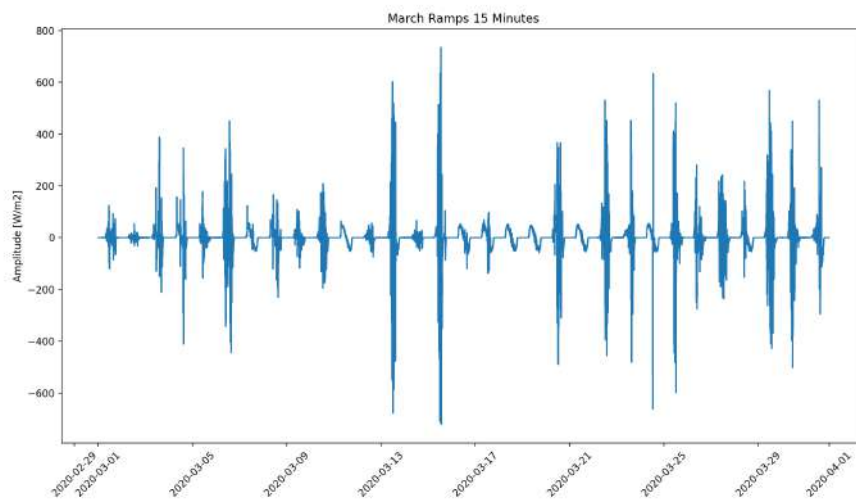


Figure 9.23: March Ramps 15 Minutes Time Horizons

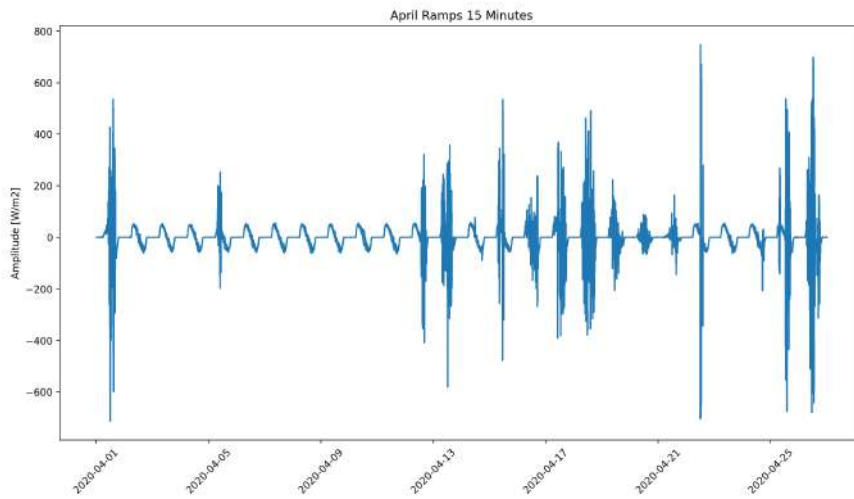


Figure 9.24: April Ramps 15 Minutes Time Horizons

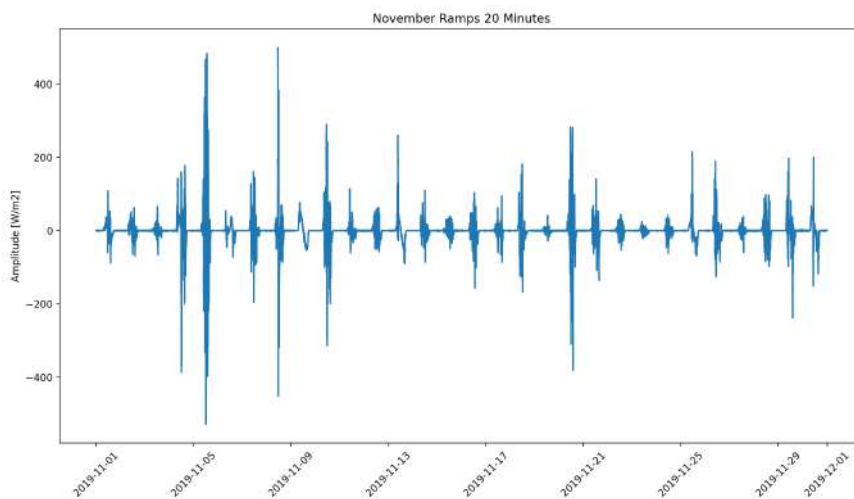


Figure 9.25: November Ramps 20 Minutes Time Horizons

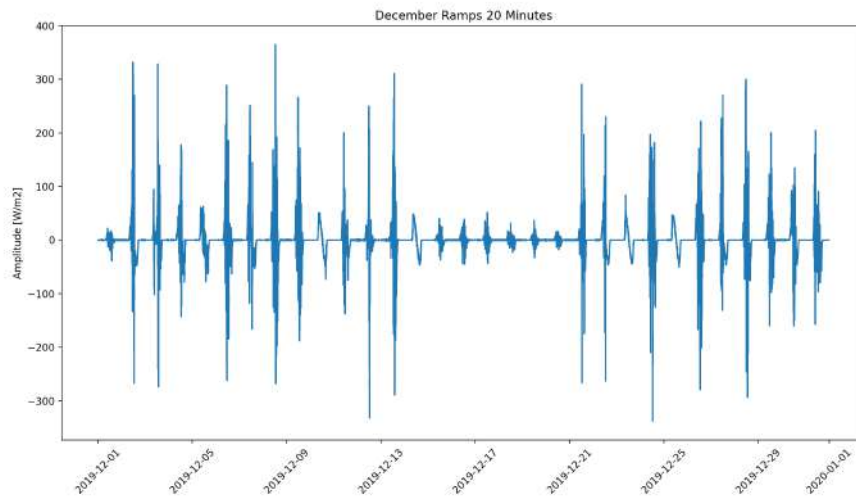


Figure 9.26: December Ramps 20 Minutes Time Horizons

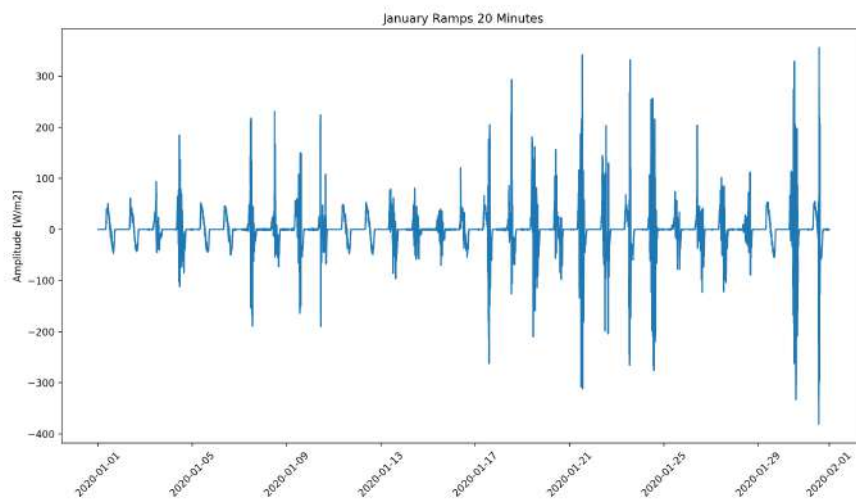


Figure 9.27: January Ramps 20 Minutes Time Horizons

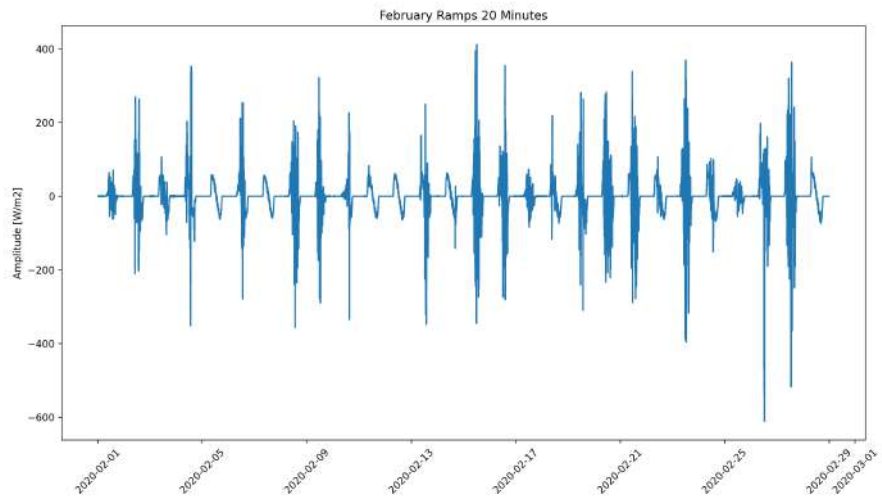


Figure 9.28: February Ramps 20 Minutes Time Horizons

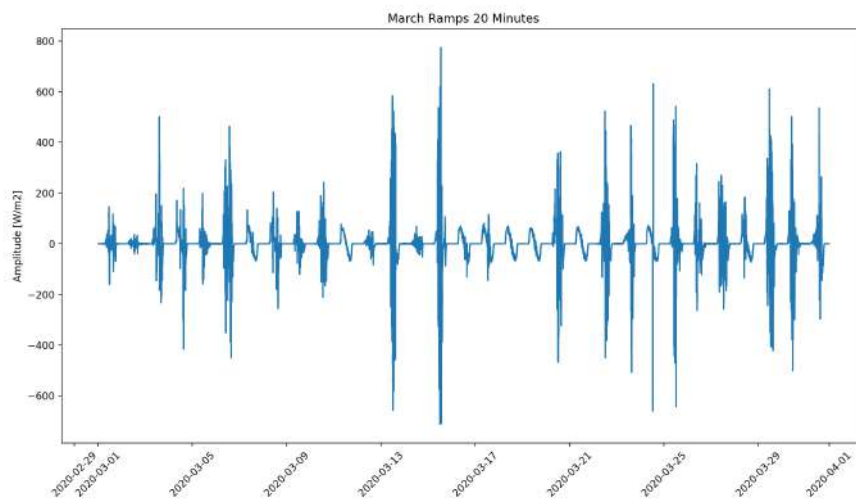


Figure 9.29: March Ramps 20 Minutes Time Horizons

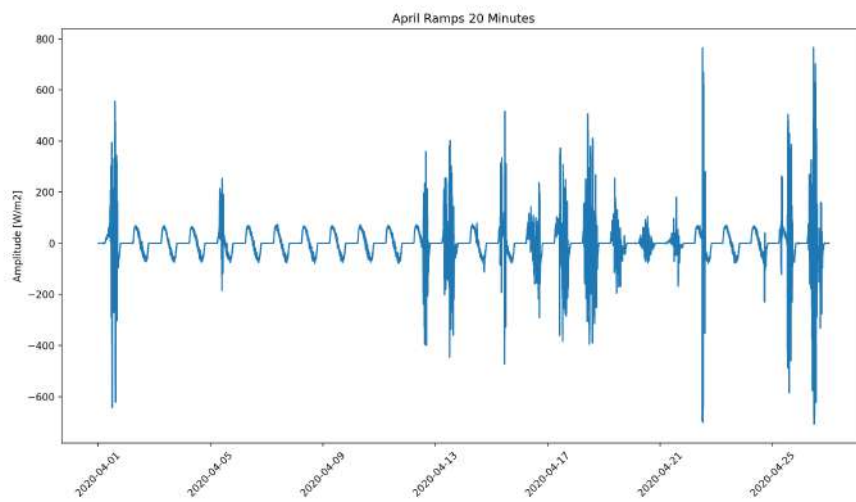


Figure 9.30: April Ramps 20 Minutes Time Horizons

Chapter 10

Appendix: Deep Learning Model Application

In figures 10.1 to 10.5, the other activation functions being used in Deep Learning models are represented.

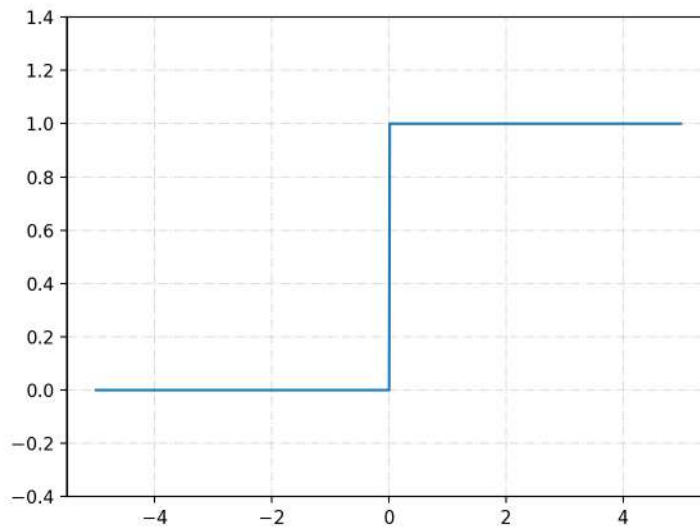


Figure 10.1: Binary Step Activation Function

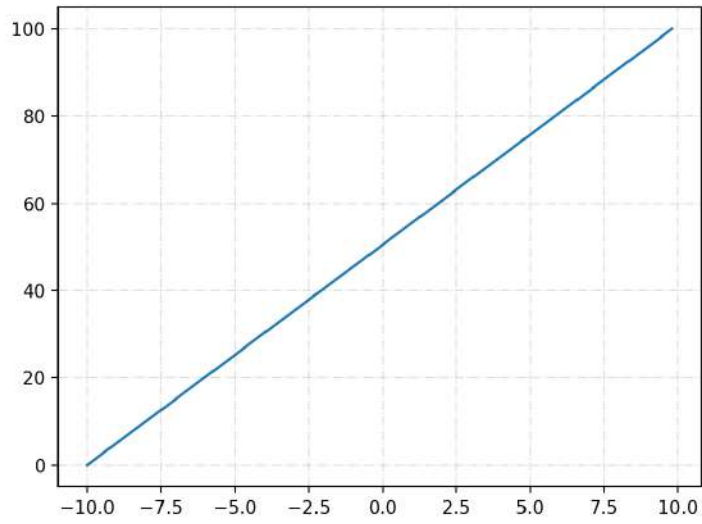


Figure 10.2: Linear Activation Function

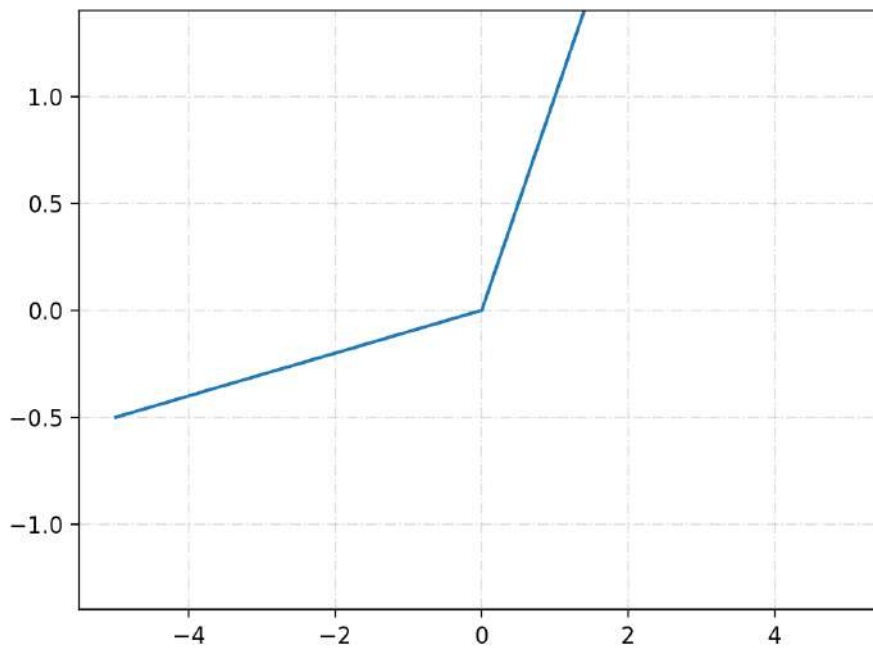


Figure 10.3: Leaky ReLU

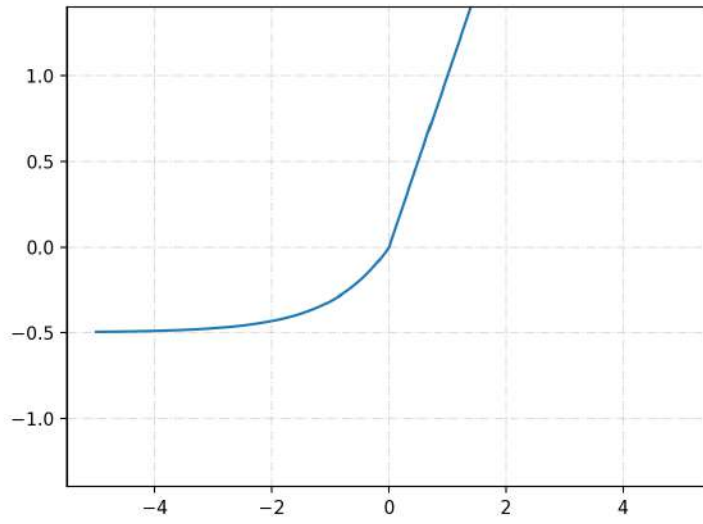


Figure 10.4: Exponential Linear Unit Activation Function

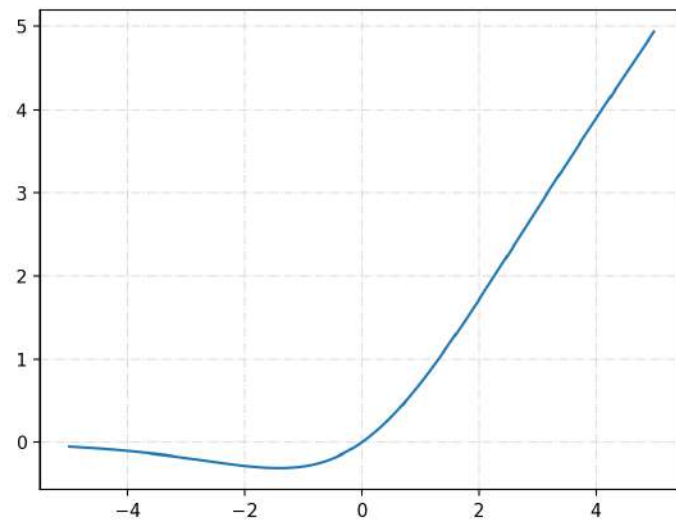


Figure 10.5: Swish Activation Function

Chapter 11

Appendix: Deep Learning Model Application

In figures 11.1 to 11.24, the remaining confusion matrices of 10 different trials of the model for 5', 10', and 15' are represented respectively.

Confusion matrix

Predicted	label1	198 3.49%	243 4.28%	85 1.50%		526 89.84%	62.38%	
	label2	96 1.69%	563 9.92%	148 2.61%		807 99.99%	31.82%	
	label3		177 3.12%	89 1.57%		266 99.99%	67.53%	
	label4	351 6.18%	278 4.90%	683 12.03%	2605 45.89%	133 2.34%	4050 99.99%	39.48%
	label5					0 0.00%	0.00%	
	sum_col	645 99.99%	1261 99.99%	1005 9.99%	2605 99.99%	161 99.99%	5677 60.86%	39.14%
		label1	label2	label3	label4	label5	sum__lin	
							Actual	

Figure 11.1: Third confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Third trial and is 60.86%.

Confusion matrix

Predicted	label1	204 3.59%	196 3.45%	70 1.23%	1 0.02%	471 82.41% 38.89%	
	label2	100 1.76%	626 11.03%	122 2.15%	4 0.07%	852 15.47% 26.32%	
	label3		143 2.52%	101 1.78%	6 0.11%	250 4.46% 39.60%	
	label4	341 6.01%	296 5.21%	712 12.54%	2605 45.89%	4104 74.41% 34.32%	
	label5					0 0.00% 0.00%	
	sum_col	645 11.87% 68.37%	1261 22.86% 54.36%	1005 18.25% 66.89%	2605 46.96% 0.00%	161 2.91% 100.00%	5677 100.00% 62.29% 37.71%
		label1	label2	label3	label4	label5	sum_lim
		Actual					

Figure 11.2: Fourth confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Fourth trial and is 62.29%.

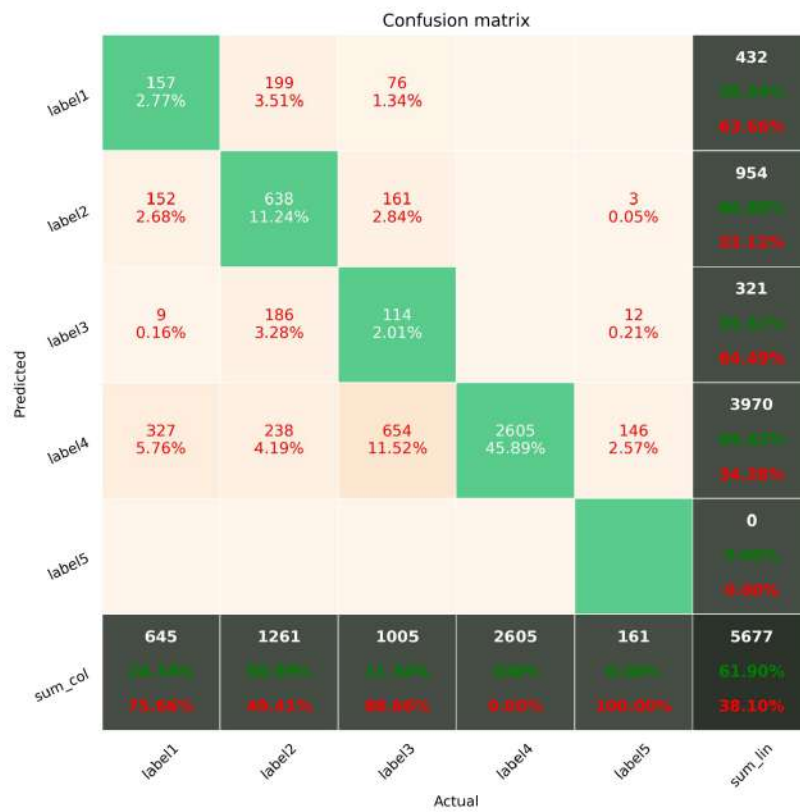


Figure 11.3: Fifth confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Fifth trial and is 61.90%.

		Actual					sum_col
		label1	label2	label3	label4	label5	
Predicted	label1	122 2.15%	146 2.57%	60 1.06%			328 5.78% 63.80%
	label2	219 3.86%	833 14.67%	244 4.30%		29 0.51%	1325 23.53% 37.18%
	label3		43 0.76%	22 0.39%		2 0.04%	67 1.19% 67.16%
	label4	304 5.35%	239 4.21%	679 11.96%	2605 45.89%	130 2.29%	3957 70.41% 34.17%
	label5						0 0.00% 0.00%
sum_col		645 11.81% 81.88%	1261 22.80% 33.84%	1005 18.21% 97.81%	2605 46.99% 0.00%	161 2.92% 100.00%	5677 100.00% 63.10% 36.90%

Figure 11.4: Sixth confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Sixth trial and is 63.10%.

Confusion matrix

Predicted	label1	1 0.02%	50 0.88%				51 0.90%	88.04%
	label2	224 3.95%	887 15.62%	308 5.43%		15 0.26%	1434 25.37%	38.15%
	label3	4 0.07%	21 0.37%	55 0.97%		2 0.04%	82 1.46%	32.83%
	label4	416 7.33%	303 5.34%	642 11.31%	2605 45.89%	144 2.54%	4110 73.10%	34.62%
	label5						0 0.00%	0.00%
	sum_col	645 11.30%	1261 22.44%	1005 18.01%	2605 46.00%	161 2.85%	5677 100.00%	62.50%
	label1	label2	label3	label4	label5	sum_lim		
								Actual

Figure 11.5: Seventh confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Seventh trial and is 62.50%.

		Confusion matrix					
Predicted	label1	177 3.12%	232 4.09%	90 1.59%		4 0.07%	503 75.18% 84.81%
	label2	221 3.89%	853 15.03%	263 4.63%		15 0.26%	1352 20.96% 36.31%
	label3		31 0.55%	75 1.32%		3 0.05%	109 16.41% 31.19%
	label4	247 4.35%	145 2.55%	577 10.16%	2605 45.89%	139 2.45%	3713 56.12% 29.84%
	label5						0 0.00% 0.00%
	sum_col	645 10.00% 12.58%	1261 19.80% 32.38%	1005 15.70% 27.54%	2605 40.00% 69.68%	161 2.40% 6.65%	5677 85.35% 34.65%
		label1	label2	label3	label4	label5	sum__lin
		Actual					

Figure 11.6: Eighth confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Eighth trial and is 65.35%.

Confusion matrix

Predicted	label1	84 1.48%	109 1.92%	16 0.28%		209 3.61%	59.81%	
	label2	349 6.15%	744 13.11%	181 3.19%		15 0.26%	1289 22.17%	41.38%
	label3		245 4.32%	188 3.31%		19 0.33%	452 7.86%	38.41%
	label4	212 3.73%	163 2.87%	620 10.92%	2605 45.89%	127 2.24%	3727 64.86%	90.10%
	label5						0 0.00%	0.00%
	sum_col	645 11.32%	1261 21.99%	1005 17.51%	2605 45.89%	161 2.80%	5677 98.51%	63.78%
		label1	label2	label3	label4	label5	sum__lin	
								Actual

Figure 11.7: Ninth confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Ninth trial and is 63.78%.

Confusion matrix

Predicted	label1	221 3.89%	402 7.08%	122 2.15%		745 60.89%	79.34%	
	label2	96 1.69%	588 10.36%	177 3.12%		865 70.38%	81.02%	
	label3		60 1.06%	43 0.76%		106 8.67%	39.47%	
	label4	328 5.78%	211 3.72%	663 11.68%	2605 45.89%	154 2.71%	3961 60.17%	34.22%
	label5						0 0.00%	0.00%
	sum_col	645 8.67%	1261 16.44%	1005 13.24%	2605 34.60%	161 2.10%	5677 60.89%	39.11%
		label1	label2	label3	label4	label5	sum__lin	
		Actual						

Figure 11.8: Ninth confusion Matrix for 5 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Tenth trial and is 60.89%.

Confusion matrix

Predicted	label1	134 2.42%	110 1.98%	26 0.47%	3 0.05%	273 48.98% 58.82%	
	label2	396 7.14%	804 14.49%	246 4.43%	17 0.31%	1463 26.38% 43.34%	
	label3	10 0.18%	210 3.79%	180 3.24%	12 0.22%	412 7.48% 36.31%	
	label4	81 1.46%	104 1.87%	519 9.35%	2570 46.32%	3400 61.48% 24.41%	
	label5					0 0.00% 0.00%	
	sum_col	621 11.36% 18.42%	1228 22.47% 34.53%	971 17.63% 21.44%	2570 46.32% 66.66%	158 2.85% 100.00%	5548 100.00% 66.47% 33.53%
		label1	label2	label3	label4	label5	sum__lin
		Actual					

Figure 11.9: Third confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Third trial and is 66.47%.

Confusion matrix

Predicted	label1	65 1.17%	55 0.99%				120 0.17%	41.67%
	label2	504 9.08%	976 17.59%	378 6.81%		45 0.81%	1903 2.47%	46.71%
	label3	3 0.05%	105 1.89%	46 0.83%		10 0.18%	164 0.21%	21.95%
	label4	49 0.88%	92 1.66%	547 9.86%	2570 46.32%	103 1.86%	3361 4.37%	21.57%
	label5						0 0.00%	0.00%
	sum_col	621 10.17%	1228 18.48%	971 14.14%	2570 36.00%	158 0.21%	5548 65.92%	34.08%
		label1	label2	label3	label4	label5	sum_lim	
								Actual

Figure 11.10: Fourth confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Fourth trial and is 65.92%.

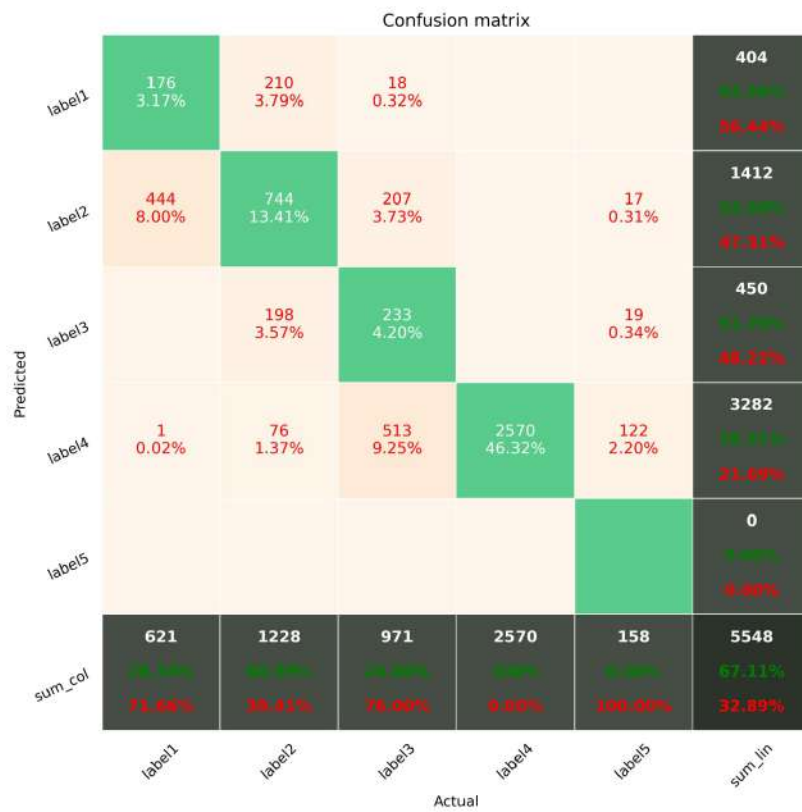


Figure 11.11: Fifth confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Fifth trial and is 67.11%.

Confusion matrix

Predicted	label1	450 8.11%	363 6.54%	91 1.64%	2 0.04%	906 16.31%	26.17%	
	label2	130 2.34%	528 9.52%	103 1.86%	15 0.27%	776 14.04%	31.84%	
	label3	19 0.34%	260 4.69%	240 4.33%	23 0.41%	542 9.89%	33.72%	
	label4	22 0.40%	77 1.39%	537 9.68%	2570 46.32%	118 2.13%	3324 59.32%	23.48%
	label5					0 0.00%	0.00%	
	sum_col	621 11.06%	1228 22.06%	971 17.71%	2570 46.32%	158 2.83%	5548 68.28%	31.72%
		label1	label2	label3	label4	label5	sum__lin	
							Actual	

Figure 11.12: Sixth confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Sixth trial and is 68.28%.

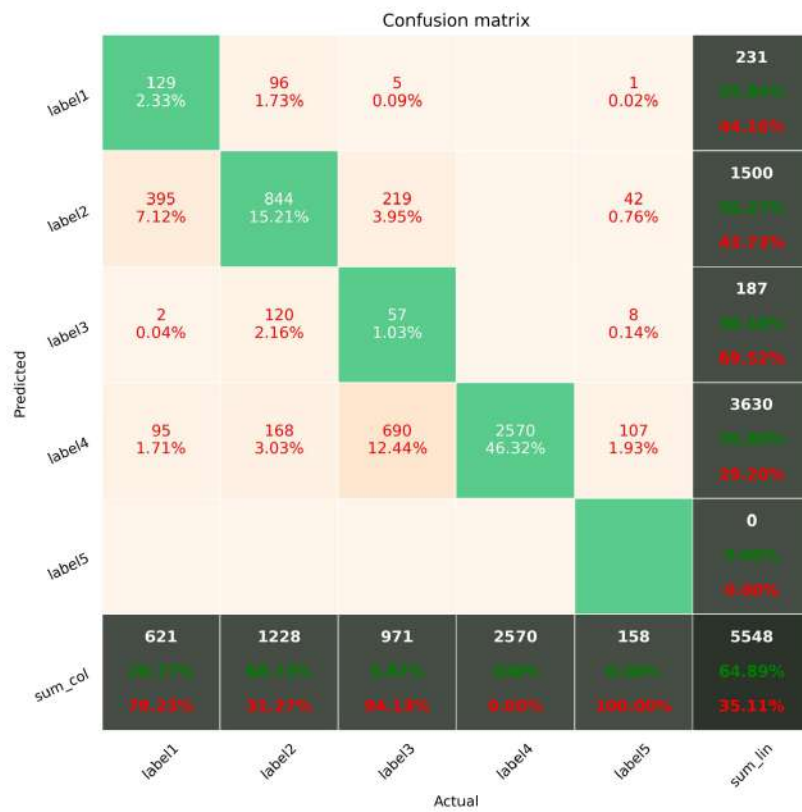


Figure 11.13: Seventh confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Seventh trial and is 64.89%.

Confusion matrix

Predicted	label1	208 3.75%	236 4.25%	62 1.12%			506 64.19%	38.89%
	label2	314 5.66%	747 13.46%	205 3.70%		18 0.32%	1284 64.19%	41.87%
	label3		90 1.62%	36 0.65%		4 0.07%	130 21.88%	32.31%
	label4	99 1.78%	155 2.79%	668 12.04%	2570 46.32%	136 2.45%	3628 70.34%	29.16%
	label5						0 0.00%	0.00%
	sum_col	621 64.88%	1228 66.41%	971 9.71%	2570 5.08%	158 0.32%	5548 64.19%	64.19%
		label1	label2	label3	label4	label5	sum_lim	
								Actual

Figure 11.14: Eighth confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Eighth trial and is 64.19%.

Confusion matrix

Predicted	label1	72 1.30%	105 1.89%	10 0.18%		187 0.34%	61.50%	
	label2	405 7.30%	790 14.24%	236 4.25%		32 0.58%	1463 26.38%	48.00%
	label3		155 2.79%	89 1.60%		9 0.16%	253 4.63%	84.82%
	label4	144 2.60%	178 3.21%	636 11.46%	2570 46.32%	117 2.11%	3645 66.31%	29.49%
	label5						0 0.00%	0.00%
	sum_col	621 11.38%	1228 22.43%	971 17.71%	2570 46.98%	158 2.90%	5548 100.00%	63.46%
		label1	label2	label3	label4	label5	sum__lin	
								Actual

Figure 11.15: Ninth confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Ninth trial and is 63.46%.

Confusion matrix

Predicted	label1	210 3.79%	150 2.70%	35 0.63%		395 70.18%	46.34%	
	label2	383 6.90%	684 12.33%	177 3.19%		1277 23.18%	48.34%	
	label3	19 0.34%	330 5.95%	240 4.33%		621 11.31%	61.37%	
	label4	9 0.16%	60 1.08%	511 9.21%	2570 46.32%	93 1.68%	3243 59.14%	29.75%
	label5		4 0.07%	8 0.14%			12 0.22%	100.00%
	sum_col	621 11.31%	1228 22.48%	971 17.73%	2570 46.99%	158 2.88%	5548 100.00%	66.76%
		label1	label2	label3	label4	label5	sum__lin	
							Actual	

Figure 11.16: Tenth confusion Matrix for 10 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Tenth trial and is 66.76%.

Predicted	Actual label1	Actual label2	Actual label3	Actual label4	Actual label5	sum_col
label1	3 0.05%	68 1.24%	33 0.60%			104 1.88% 97.12%
label2	291 5.31%	701 12.78%	254 4.63%		49 0.89%	1295 23.17% 45.87%
label3		31 0.57%	17 0.31%		3 0.05%	51 0.93% 66.67%
label4	316 5.76%	410 7.48%	656 11.96%	2549 46.48%	103 1.88%	4034 73.13% 34.81%
label5						0 0.00% 0.00%
sum_col	610 11.14% 84.51%	1210 22.04% 43.07%	960 17.43% 48.23%	2549 46.48% 9.60%	155 0.28% 100.00%	5484 59.63% 40.37%
	Actual label1	Actual label2	Actual label3	Actual label4	Actual label5	sum_row

Figure 11.17: Third confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Third trial and is 59.53%.

Confusion matrix

Predicted	label1	1 0.02%	35 0.64%	15 0.27%			51 0.94%	88.04%
	label2	142 2.59%	546 9.96%	237 4.32%		24 0.44%	949 17.37%	43.47%
	label3		106 1.93%	71 1.29%		6 0.11%	183 3.36%	61.30%
	label4	467 8.52%	523 9.54%	637 11.62%	2549 46.48%	125 2.28%	4301 79.27%	40.73%
	label5						0 0.00%	0.00%
	sum_col	610 11.40%	1210 22.47%	960 17.68%	2549 46.96%	155 2.88%	5484 100.00%	57.75%
		label1	label2	label3	label4	label5	sum_lim	
								Actual

Figure 11.18: Fourth confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Fourth trial and is 59.75%.

Confusion matrix

Predicted	label1	62 1.13%	60 1.09%	21 0.38%		143 00.00% 00.00%	
	label2	150 2.74%	446 8.13%	103 1.88%		751 00.00% 40.81%	
	label3	4 0.07%	165 3.01%	130 2.37%		302 00.00% 00.00%	
	label4	394 7.18%	539 9.83%	706 12.87%	2549 46.48%	4288 00.00% 40.90%	
	label5					0 0.00% 0.00%	
	sum_col	610 00.00% 00.00%	1210 00.00% 00.00%	960 00.00% 00.00%	2549 00.00% 00.00%	155 00.00% 100.00%	5484 58.11% 41.89%
		label1	label2	label3	label4	label5	sum_row
		label1	label2	label3	label4	label5	sum_col
							Actual

Figure 11.19: Fifth confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Fifth trial and is 58.11%.

Predicted	Actual label1	Actual label2	Actual label3	Actual label4	Actual label5	sum_row
label1	13 0.24%	91 1.66%	1 0.02%	1 0.02%	0	106 18.48% 87.74%
label2	252 4.60%	898 16.37%	402 7.33%	0	37 0.67%	1589 26.91% 43.49%
label3	0	17 0.31%	17 0.31%	0	0	17 0.29% 100.00%
label4	345 6.29%	204 3.72%	557 10.16%	2548 46.46%	118 2.15%	3772 63.80% 32.49%
label5	0	0	0	0	0	0 0.00% 0.00%
sum_col	610 10.14% 87.87%	1210 20.41% 25.78%	960 16.67% 100.00%	2549 43.84% 0.84%	155 0.26% 100.00%	5484 93.07% 36.93%
	label1	label2	label3	label4	label5	sum_col
	Actual	Actual	Actual	Actual	Actual	Actual

Figure 11.20: Sixth confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Sixth trial and is 63.07%.

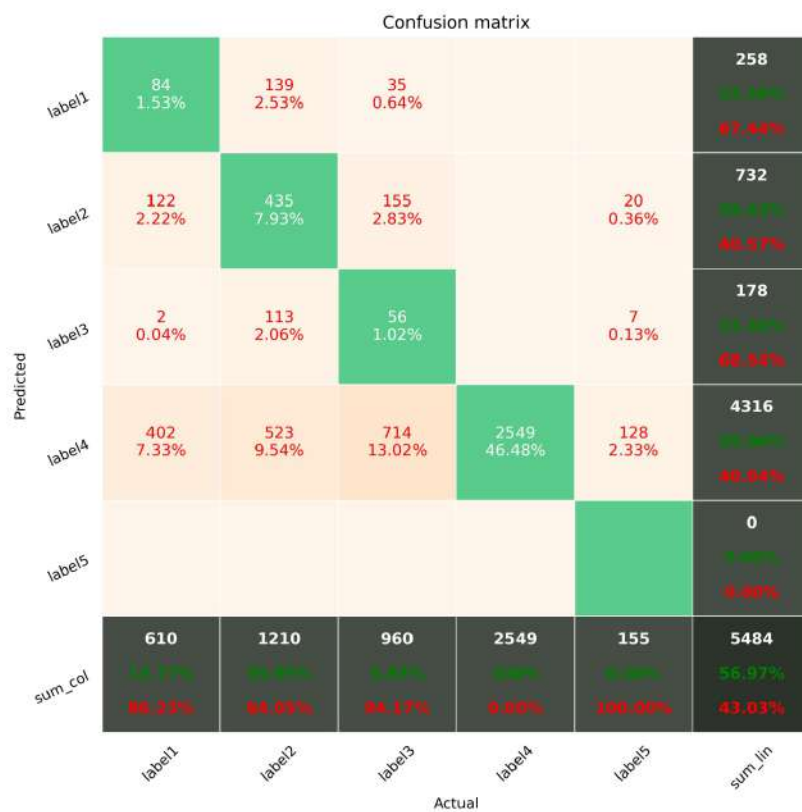


Figure 11.21: Seventh confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Seventh trial and is 56.97%.

Confusion matrix

Predicted	label1	6 0.11%	85 1.55%	18 0.33%			109 1.38% 84.30%
	label2	226 4.12%	674 12.29%	232 4.23%		25 0.46%	1157 14.71% 41.75%
	label3		13 0.24%	20 0.36%		1 0.02%	34 0.44% 41.18%
	label4	378 6.89%	438 7.99%	690 12.58%	2549 46.48%	129 2.35%	4184 53.51% 39.68%
	label5						0 0.00% 0.00%
	sum_col	610 8.48% 88.82%	1210 17.46% 44.38%	960 13.44% 67.82%	2549 35.61% 9.68%	155 0.21% 100.00%	5484 71.76% 40.75%
		label1	label2	label3	label4	label5	sum_lim
		Actual					

Figure 11.22: Eighth confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Eighth trial and is 59.25%.



Figure 11.23: Ninth confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Ninth trial and is 54.91%.

		Confusion matrix					
Predicted	label1	145 2.64%	192 3.50%	25 0.46%			362 66.98% 59.94%
	label2	422 7.70%	710 12.95%	203 3.70%		22 0.40%	1357 25.33% 47.88%
	label3	2 0.04%	97 1.77%	112 2.04%		8 0.15%	219 4.12% 40.80%
	label4	41 0.75%	211 3.85%	620 11.31%	2549 46.48%	125 2.28%	3546 66.30% 28.12%
	label5						0 0.00% 0.00%
	sum_col	610 11.71% 16.21%	1210 22.40% 41.33%	960 17.71% 28.33%	2549 47.60% 69.60%	155 0.28% 100.00%	5484 100.00% 35.89%
		label1	label2	label3	label4	label5	sum_row

Figure 11.24: Tenth confusion Matrix for 15 Minutes Forecasting. The columns of this matrix shows the predicted label of the images. The rows of this matrix shows the Actual Label of the images. The prediction accuracy of the Tenth trial and is 64.11%.

Bibliography

- [1] Dazhi Yang, Jan Kleissl, Christian A Gueymard, Hugo T C Pedro, and Carlos F M Coimbra. History and trends in solar irradiance and PV power forecasting: A preliminary assessment and review using text mining. *Solar Energy*, 168:60–101, 2018.
- [2] 2017 Int. Energy Agency. World Energy Outlook 2017. 2017.
- [3] S Leva, A Nespoli, S Pretto, M Mussetta, and E G C Ogliari. PV Plant Power Nowcasting: A Real Case Comparative Study With an Open Access Dataset. *IEEE Access*, 8:194428–194440, 2020.
- [4] L Bird, M Milligan, and D Lew. Integrating Variable Renewable Energy: Challenges and Solutions. Technical report, United States, 2013.
- [5] Alfredo Nespoli and Alessandro Niccolai. Solar Position Identification on Sky Images for Photovoltaic Nowcasting applications. pages 1–5, 2020.
- [6] Deep photovoltaic nowcasting. *Solar Energy*, 176:267–276, 2018.
- [7] Cyril Voyant, Gilles Notton, Soteris Kalogirou, Marie-Laure Nivet, Christophe Paoli, Fabrice Motte, and Alexis Fouilloy. Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*, 105:569–582, 2017.
- [8] Sophie Vorrath. HE New APVI solar tool shows a daily, time-based forecast for each state.
- [9] M J Sanjari and H B Gooi. Probabilistic Forecast of PV Power Generation Based on Higher Order Markov Chain. *IEEE Transactions on Power Systems*, 32(4):2942–2952, jul 2017.

- [10] Wikipedia. Numerical Weather Predictions, 2016.
- [11] A Catalina, C M Alaíz, and J R Dorransoro. Combining Numerical Weather Predictions and Satellite Data for PV Energy Nowcasting. 11(3):1930–1937, jul 2020.
- [12] V K Dehariya, S K Shrivastava, and R C Jain. Clustering of Image Data Set Using K-Means and Fuzzy K-Means Algorithms. In *2010 International Conference on Computational Intelligence and Communication Networks*, pages 386–391, nov 2010.
- [13] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, New York, NY, 2 edition, 2009.
- [14] M V Valueva, N N Nagornov, P A Lyakhov, G V Valuev, and N I Chervyakov. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177:232–243, 2020.
- [15] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, nov 1997.
- [16] Md Khalid, Naina Pal, and Kamiya Arora. Clustering of Image Data Using K-Means and Fuzzy K-Means. *International Journal of Advanced Computer Science and Applications*, 5, 2014.
- [17] Hsu-Yung Cheng. Cloud tracking using clusters of feature points for accurate solar irradiance nowcasting. *Renewable Energy*, 104:281–289, 2017.
- [18] Donatello Gallucci, Filomena Romano, Angela Cersosimo, Domenico Cimini, Francesco Di Paola, Sabrina Gentile, Edoardo Gherardi, Salvatore Larosa, Saverio T Nilo, Elisabetta Ricciardelli, and Mariassunta Viggiano. Nowcasting Surface Solar Irradiance with AMESIS via Motion Vector Fields of MSG-SEVIRI Data. *Remote Sensing*, 10(6), 2018.
- [19] B Nouri, S Wilbert, L Segura, P Kuhn, N Hanrieder, A Kazantzidis, T Schmidt, L Zarzalejo, P Blanc, and R Pitz-Paal. Determination of cloud transmittance for all sky imager based solar nowcasting. *Solar Energy*, 181:251–263, 2019.

- [20] Bijan Nouri. *Solar irradiance nowcasting system to optimize the yield in parabolic trough power plants = Solarstrahlungs-Kürzestfrist-Vorhersagesystem für die Ertragsoptimierung eines Parabolrinnenkraftwerks*. PhD thesis, 2020.
- [21] Shan Zeng, Celine Cornet, Frederic Parol, J Riedi, and F Thieuleux. A better understanding of cloud optical thickness derived from the passive sensors MODIS/AQUA and POLDER/PARASOL in the A-train constellation. *Atmospheric Chemistry Physics Discussions*, 12:11733–11764, 2012.
- [22] H Moosmüller and C M Sorensen. Small and large particle limits of single scattering albedo for homogeneous, spherical particles. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 204:250–255, 2018.
- [23] B Nouri, P Kuhn, S Wilbert, N Hanrieder, C Prah, L Zarzalejo, A Kazantzidis, P Blanc, and R Pitz-Paal. Cloud height and tracking accuracy of three all sky imager systems for individual clouds. *Solar Energy*, 177:213–228, 2019.
- [24] Salvador Matamoros, Josep-Abel González, and Josep Calbó. A Simple Method to Retrieve Cloud Properties from Atmospheric Transmittance and Liquid Water Column Measurements. *Journal of Applied Meteorology and Climatology*, 50, 2011.
- [25] Pascal Kuhn, Bijan Nouri, Stefan Wilbert, Christoph Prah, Nora Kozonek, Thomas Schmidt, Zeyad Yasser, Lourdes Ramirez, Luis Zarzalejo, Angela Meyer, Laurent Vuilleumier, Detlev Heinemann, Philippe Blanc, and Robert Pitz-Paal. Validation of an all-sky imager-based nowcasting system for industrial PV plants. *Progress in Photovoltaics: Research and Applications*, 26(8):608–621, 2018.
- [26] Y Marif, Y Chiba, M M Belhadj, M Zerrouki, and M Benhammou. A clear sky irradiation assessment using a modified Algerian solar atlas model in Adrar city. *Energy Reports*, 4:84–90, 2018.
- [27] Andrea Cipollina, Eftihia Tzen, Vicente Subiela, Michael Papapetrou, and J. Koschikowski. Renewable energy desalination: Performance analysis and operating data of existing res-desalination plants. volume 55, 05 2014.

- [28] Tutorials Point. Artificial Intelligence, 2021.
- [29] Tom M Mitchell. *Machine Learning*. McGraw-Hill, New York, NY, 1 edition, 1997.
- [30] John R Koza, Forrest H Bennett, David Andre, and Martin A Keane. *Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming*, pages 151–170. Springer Netherlands, 1996.
- [31] Simon Tavasoli. The Importance of Machine Learning for Data Scientists, 2021.
- [32] Erik G Learned-Miller. Introduction to supervised learning. *I: Department of Computer Science, University of Massachusetts*, 2014.
- [33] Peter Dayan, Maneesh Sahani, and Grégoire Deback. Unsupervised learning. *The MIT encyclopedia of the cognitive sciences*, pages 857–859, 1999.
- [34] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.
- [35] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin. Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12):14413–14423, 2020.
- [36] L Kaelbling, M Littman, and A Moore. Reinforcement Learning: A Survey. *J. Artif. Intell. Res.*, 4:237–285, 1996.
- [37] Orhan Yalchin. 4 Machine Learning Approaches that Every Data Scientist Should Know, 2020.
- [38] F. Q. Lauzon. An introduction to deep learning. In *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pages 1438–1439, 2012.
- [39] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012.

- [40] Andrej Krenker, Janez Bešter, and Andrej Kos. Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech*, pages 1–18, 2011.
- [41] Facundo Bre, Juan Gimenez, and Víctor Fachinotti. Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. *Energy and Buildings*, 158, 2017.
- [42] Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.
- [43] Valeria Maeda-Gutiérrez, Carlos E Galván-Tejada, Laura A Zanella-Calzada, José M Celaya-Padilla, Jorge I Galván-Tejada, Hamurabi Gamboa-Rosales, Huizilopoztli Luna-García, Rafael Magallanes-Quintanar, Carlos A Guerrero Méndez, and Carlos A Olvera-Olvera. Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases. *Applied Sciences*, 10(4), 2020.
- [44] Ahmed Tealab. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, 3(2):334–340, 2018.
- [45] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.
- [46] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2009.
- [47] H Sak, Andrew Senior, and F Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 338–342, 2014.
- [48] Xiangang Li and Xihong Wu. Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. pages 4520–4524, 2015.

- [49] Milos Miljanovic. Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction. *Indian Journal of Computer Science and Engineering*, 3, 2012.
- [50] Jason Brownlee. A Gentle Introduction to Generative Adversarial Networks (GANs), 2019.
- [51] Adit Burner, Jon Deshpan. Generative Adversarial Networks for Beginners, 2017.
- [52] Google. Tensorflow, 2021.
- [53] Francois Chollet. Keras, 2021.
- [54] Pandas. Pandas DataFrame, 2014.
- [55] Solar Tech Lab(<http://www.solartech.polimi.it>). Solar Tech Lab.
- [56] Skimage. Scikit-Image Python Library.
- [57] Numpy(Numpy.org). Numpy Python Library.
- [58] OpenCV. OpenCV Image Processing Library.
- [59] PIL. Python Imaging Library.
- [60] Ozan Erdnic. *Optimization in Renewable Energy Systems*. Elsevier, Turkey, first edition, 2017.
- [61] Miguel Esteban(Datacamp.com). Introduction to Deep Learning with Keras, 2021.
- [62] Deeplearning Track. Introduction to NEURAL NETWORKS, Advantages and Applications, 2017.
- [63] Jahnavi Mahanta. Introduction to Neural Networks, Advantages and Applications, 2017.
- [64] Sagar Sharma. Activation functions in neural networks. *towards data science*, 6, 2017.
- [65] Machine Learning Mastery. How to Choose an Activation Function for Deep Learning, 2021.
- [66] Machine Learning Mastery. Softmax Activation Function with Python, 2021.

- [67] Marco Leo and Giovanni Maria Farinella. *Computer Vision For Assistive Healthcare*. 2018.