Prediction of Photovoltaic Power Generation from Cloud Imaging

Master Thesis

Amrollah Seifoddini

March 31, 2016

Advisors: Prof. Marc Pollefeys, Dr. Jan Poland

Department of Computer Science, ETH Zürich

Abstract

Managing fluctuation in photo-voltaic power plants which is frequent in cloudy days, is one of the big challenges that need to be solved in order to significantly increase its penetration into the power grid. One possible approach to predict short term variations is vision-based which includes a fish eye camera pointing into the sky, taking image sequences. Cloud states are predicted for near future using cloud segmentation and optical flow. In this context, we investigate the role of the cloudiness in shielding the direct sun irradiation and also reflecting the sunlight which increases the diffuse irradiation. Specifically, by analyzing our irradiation sensor measurements and image features we learn a soft sensor regressor for inferring irradiance of a given image. Direct component of irradiation is inferred using sun detection algorithm and cloud map. Finally the diffuse irradiation is estimated using several features derived from cloud state and date and time of image. Several regression algorithms are compared and Support Vector Ordinal Regression Machine delivers the best result with $34W/m^2$ error and $33W/m^2$ error standard deviation.

Acknowledgements

This work would not have been feasible without the clever ideas and the patient explanations of my supervisor Dr. Poland. Also, I would like to thank ABB for providing me with the infrastructure, data, and human support which allowed me to conduct my thesis in the best conditions.

Contents

Contents iii

List of Figures 1

List of Tables 2

1 Introduction 3
1.1 Power prediction approaches for a PV plant 3
1.1.1 Ground whole-sky imagery 4
1.2 Cloud segmentation, cloud tracking 4
1.3 Image irradiance estimation for power prediction 6
1.4 Irradiance components 6
1.5 Accuracy metrics 7

2 Related work 9
2.1 Estimate irradiance from zone types in sky images 9
2.2 Using clear sky irradiance model and binary cloud mask . . . 10
2.2.1 Cloud detection 11
2.2.2 Image un-distorion 11
2.2.3 Shadow mapping 12
2.2.4 Irradiance retrieval 12
2.2.5 Irradiance forecast 13
2.3 Retrieval of direct and diffuse irradiance from sky images . . 13
2.3.1 Image features 13
2.3.2 irradiance estimation 14

3 Estimating Diffuse Horizontal Irradiance (DHI) from sky image 15
3.1 Experimental Setup 15
3.2 Acquired Data 17
3.3 Sun positions and sun states in image 17
3.4 Clear-sky irradiance model 19

3.4.1 Ineichen method 20
3.4.2 McClear method 21
3.4.3 Comparison of clear-sky simulation results 21
3.4.4 Choosing the clear-sky model 25
3.5 Estimating diffuse from pyranometers 26
3.6 Key factors influencing DHI 32
3.6.1 Cloud coverage features 33
3.6.2 Saturation factor 33
3.7 Dataset 35
3.8 Learning the relation between image features and DHI 36
3.8.1 Linear Regression 36
3.8.2 K-nearest-neighbors (K-NN) Regression 36
3.8.3 Support Vector Machine Regression (SVR) 37
3.8.4 Training and test datasets 37

4 Results and Discussion **39**
4.1 Feature Selection 39
4.2 Linear Regression 40
4.3 K-NN 41
4.4 SVR 43
4.5 Comparison 46

5 Future Work **49**

6 Conclusion **51**

Bibliography **53**

List of Figures

1.1 Cloud segmentation sample 5
1.2 Irradiance components 7

2.1 Three different zones identified in images (sun, cloud, sky) . . . 10
2.2 Correlation between estimated DNI and recorded DNI. 10
2.3 Comparing estimated k^*_{DHI} and k^*_{DNI} to measured values. source:[29] 14

3.1 wide-angle camera system used at Carviglia site 16
3.2 Two pyranometers (one horizontal, one 45 degrees titled to north) located close to the camera location 16
3.3 Sun position angles. source:[1] 18
3.4 Different sun states, from left to right: sun flag=4, 3, 2, 1 respectively. 19
3.5 Variation of sun appearance in the images. 20
3.6 GHI estimation of McClear vs Ineichen vs observation 22
3.7 Correlation of GHI simulation of McClear and Ineichen with observations on selected days 23
3.8 DNI and DHI correlation of McClear to Ineichen 24
3.9 Comparison of DHI of McClear and Ineichen to observed GHI . 25
3.10 Variation of irradiance components during day 25
3.11 The sun path during a spring day at Cavriglia, Italy, location of ABB PV plant. 27
3.12 Irradiance comparison of horizontal sensor (1) and tilted sensor (2) for a summer day 27
3.13 Irradiance comparison of horizontal sensor (1) and tilted sensor (2) for a winter day 28
3.14 Calculated DHI 30
3.15 Calculated GHI based on DHI and DNI for both sensors 31
3.16 Regions of interest for extracting cloud coverage features 34
3.17 One example of saturated cloud pixels 34

3.18 Detected saturated area in a patch around sun 35

4.1 Linear regression result using only non-image features 40
4.2 Linear regression result using all the features 41
4.3 Error histogram for linear regression 42
4.4 K-NN estimation result using only non-image features 43
4.5 K-NN estimation result using all the features 44
4.6 Error histogram for K-NN estimation result 45
4.7 SVR result using only non-image features 45
4.8 SVR result using both feature types 46
4.9 Error histogram for SVR result 46

List of Tables

4.1 Regression errors of K-NN on non-image and full feature vectors 42
4.2 Regression errors of SVR on non-image and both feature types . 44
4.3 Comparison of RMSE for all three regression methods using both feature types 47

Chapter 1

Introduction

Solar energy is one of the key alternative energy sources. The recent developments in solar panels and different business models around them made the photo-voltaic(PV) power plants more economical. However, the variability in PV output power make the integration into main energy grid risky and slow[27]. These fluctuation comes from cloud states in sky, and it has two different effects, one decreasing the power, the other one increasing the power. Firstly, if the clouds cover the sun completely or partially, some area of plant is shaded and does not receive direct sunlight, causing a power drop. On the other hand, if the clouds are not blocking the the sun completely or at all, based on their type, height, position and time, they can re-reflect some part of the irradiation[1] which is reflected by ground, back into the power plant. In this case, the input irradiance[2] and consequently the output power increases. In the electricity grid, the stability of power is vital. Therefore, if we want to integrate a PV power source into the grid we need to compensate for any power drop by using other electricity sources, and also restrain any excessive power. That is why we need to predict these short-time power changes in advance to design better strategies for handling them and ultimately provide a guaranteed stable power in the grid. In this chapter, we first explain different approaches towards this prediction problem, then we describe overview of the setup used in this study, and finally we talk about accuracy metrics for the result.

1.1 Power prediction approaches for a PV plant

The large variety of cloud characteristics such as motion, height, opacity and spatial distribution makes the cloud-induced fluctuations difficult to

[1] Irradiation is the sum of irradiance over a time period, expressed in Wh/m^2

[2] Irradiance is understood as instantaneous density of solar radiation incident on a given surface, typically expressed in W/m^2

predict. However, according to these comprehensive survays[13, 19], solar irradiance forecasting techniques have been successfully developed. These include numerical weather models (NWPs)[22] using pattern recognition of meteorological data for irradiance prediction, satellite-based forecasts using cloud motion vectors to determine sun occlusion based on fixed velocity model and predict power [16, 20, 11], statistical methods based on machine learning applied on past several years trend[35] and time series analysis[25] which are mostly developed for intra-day and day-ahead forecasts. However, for very short-term power prediction applications, the interest horizon stretches up to 30 minutes ahead. And therefore, these methods fall begind the required spatial or temporal resolution required on cloud-induced irradiance variability[13].

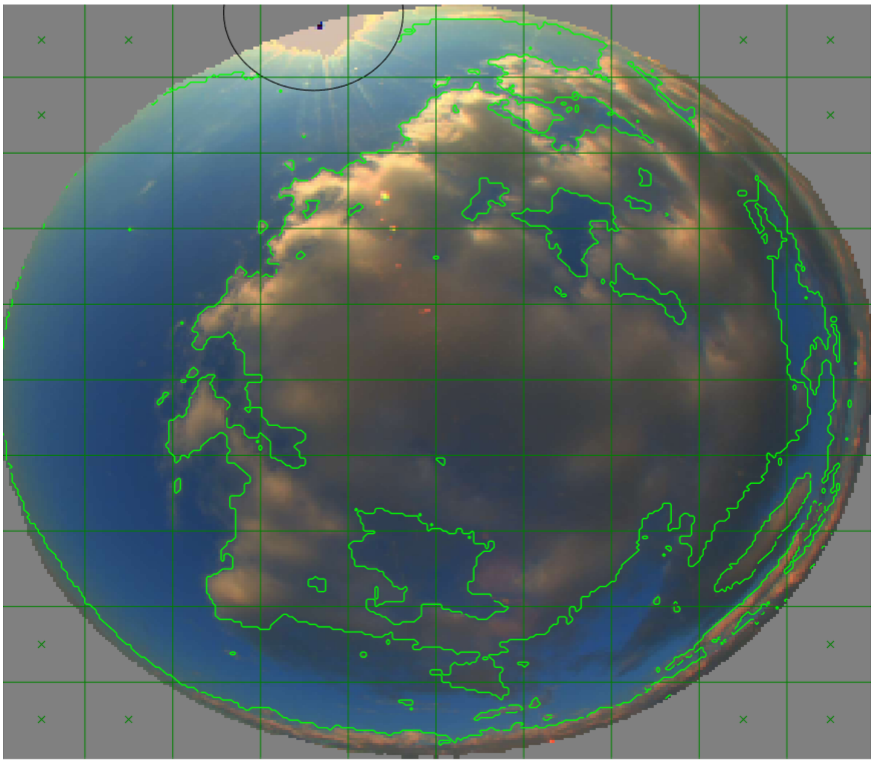
1.1.1 Ground whole-sky imagery

For acheiving this high resolution forecasts, vision-based methods using total-sky-cameras are developed. The earliest works use cameras for monitoring cloud cover characteristics[23, 7] and aerosol properties[21, 6]. In recent years, using sky cameras for solar irradiance forecast has grown rapidly and several successful works have been developed by analysing motion, optical and distribution of clouds in the whole-sky images captured by a fisheye lense camera.(Chu et al., 2015;West et al., 2014; Quesada-Ruiz et al., 2014; Chu et al., 2013; Fu and Cheng, 2013; Yang et al., 2014; Bernecker et al., 2014; Chow et al., 2011; Marquez and Coimbra, 2013). Using only one camera for data input, some methods[34] predict sun occlusions percieved by the camera and therefore, their forecast in only valid for the point very close to the camera, since areas further away might be sunny or cloudy and camera's point of view is not covering that area. However, one can incorporate cloud base height to calculate projection of clouds shadows on the ground. This information is usually acquired by using a laser based cloud base sensor (ceilometer) and make the area irradiance forecasts more accurate[36]. It is theoritically possible to use two or more cameras in the site mounted with a distance about 50m and by applying stereo algorithm, find the cloud height; however, this method has not been investigated in practice yet. In this research we focus on ground whole-sky imagery methods aimed for very short-term irradiance forecast.

1.2 Cloud segmentation, cloud tracking

For predicting the future state of clouds in the sky we need to first know where are the clouds. This is done by applying a dynamic-threshold segmentation algorithm on red to blue channel ratio of RGB images. Several studies has shown that the red to blue ratio is a good criterion for cloud segmentation, but the threshold for this ratio should be set in a way that discriminate clouds locally but be smooth globally. For this purspose, image pixels are mapped to a grid and threshold is determined for each grid separately while mainting the smoothness globally. This method handles cloud color variation for different cloud types very well. This detail of the algorithm for finding this optimal thresholds for every grid is out of the scope of this study. One sample result of such segmentation is given in Figure 1.1.

Figure 1.1: Cloud segmentation sample



This is a binary segmentation, meaning that classification result for every pixel is either cloud or not cloud (i.e. sky). If sun is visible in the image, a small circle around the sun is excluded for segmentation, thus does not have any class in the output. The same goes for pixels outside of sky mask. After detecting the clouds in a sequence of images, one can apply optical flow algorithm on some points of interest in the first image and extract the cloud movement as motion vectors of optical flow. This cloud tracking pipeline gives us the clouds position is the sky for very short time in future. The result of experiment on several future time horizon has shown that accuracy decreases considerably after 30 minutes, specially in high speed cloud motions.

1.3 Image irradiance estimation for power prediction

The final step in power prediction is to associate a potential power estimate to any time in several minutes ahead knowing the cloud positions and their characteristic in that time. The generated power of a PV plant depends on several factors including received irradiance, operational temperature and panels specification. However, the only factor which changes rapidly and has a huge impact on the output power is irradiance. Therefore, in our power prediction framework, we use a power prediction adaptation method with recorded data of previous minutes to derive the power output from future irradiance estimations. The scaling factor for the result is calculated by diving the recorded power output and irradiance sensor measurements. Thus, this power adaptation mechanism, separates and defines our main problem as estimating the received irradiance from a clouds attributes in a specific time and location.

1.4 Irradiance components

The total solar radiation -GHI[3]- which hits the surface of solar panels consists of three basic components, direct -DNI[4]-, diffuse -DHI[5]- and reflected. The direct part comes from the sunlight beams directly raying from sun direction to the solar module. While passing through atmosphere, some amount of sunlight scatters in every direction by dense particles. The portion of this scattered light which hits the module forms the diffuse irradiation for solar panels. Reflected irradiance represents sunlight that is reflected off the clouds or ground around the array of panels. The source of this reflected radiation can be DNI or DHI. Rate of the reflection depends on clouds coverage, size of the ground that is visible from the module and their albedo coefficient[6]. The albedo coefficient for ground is typically 0.2, though it can be higher during snowy periods in cold climates. The albedo coefficient for clouds depends on their type, density, temperature and etc. These components are shown in figure 1.2

Total irradiation is related to other three components with this formula:

$$GHI = DNI \times cos(Z) + DHI + reflected$$

where Z is the solar zenith angle-the angle between the direction of the sun and the line directly overhead. Since distinguishing between the reflected and diffuse irradiation is practically hard and also there is not any ground

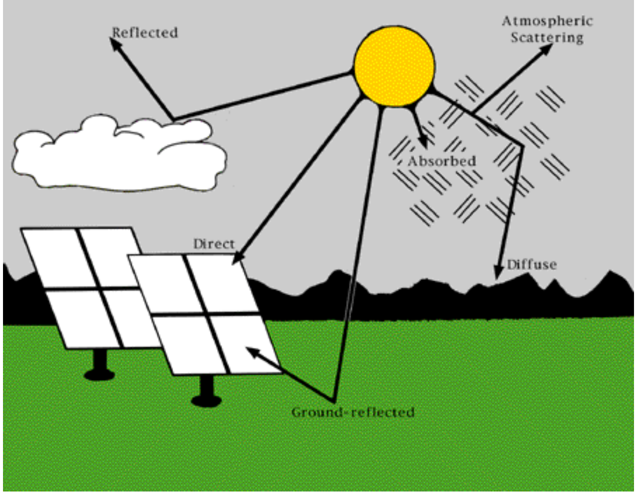
[3] Global Horizontal Irradiation

[4] Direct Normal Irradiance

[5] Diffuse Horizontal Irradiance

[6] The portion of the incident irradiance that is reflected

Figure 1.2: Irradiance components



truth value for training, we decided to combine both of them as the diffuse component. Thus, the formula changes to:

$$GHI = DNI \times cos(Z) + DHI \tag{1.1}$$

where DHI is sum of all non direct irradiations.

1.5 Accuracy metrics

For measuring accuracy of the result we can use popular error measures such as RMSE-Root mean square error. However, since there are not any other publicly available study on this specific problem to compare our RMSE with, we can define our own error measures which quantify our solution quality better. For example, apart from RMSE, we calculate a relative error as well that penalize errors for irradiances higher than a specific value more. And errors for irradiances lower than that value, will be penalized less. This cutting value is set as 100 and the scale is logarithmic, because when the irradiance is less than 100, the power output of plant is very low and practically not useful in grid. On the other hand, we want more precise result for very high irradiances. This means errors for small irradiances are less important than errors in big big ones. We also use MBE (mean bias error) and R^2 for correlation coefficient.

Chapter 2

Related work

In this chapter we survey the studies which focus on problem of irradiance estimation using sky images. The usage of ground-based cameras for studying effect of clouds on irradiation has a long history, as early as 1977 when Borkowski et al.[4] developed the first whole-sky camera system for investigating effects of clouds on middle ultraviolet global radiation. In this study, the degree of solar obstruction and cloud coverage were determined visually from the images. Later in 1998, Jeff Sabburg and Joe wong[26] developed and evalued the first automated, ground-based, sun-centered sky camera system for cloud assessment. However, since the purpose of study was the clouds effect on UVB[1] radiation they only considered a small area around the sun for cloud and sun obstruction detection which is of paramount importance for this rays. They use a threshold-based approach on gray scale pixel intensities for cloud detection. They also use solar radiation measurements in a image processing algorithm to reduce reflections from the sun on the camera system being mistaken for cloud in the images.

2.1 Estimate irradiance from zone types in sky images

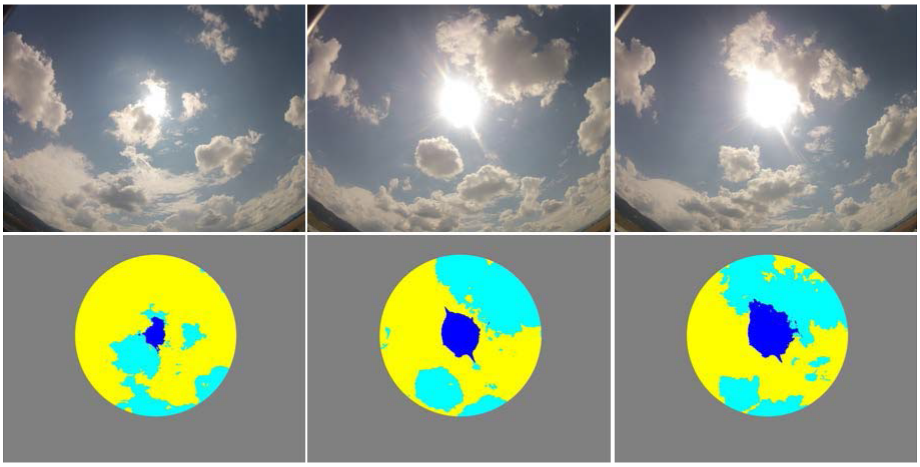
One of the recent researches done in this area is held as a collaboration between Universitatea Transilvania din Braşov in Romania and Cyprus University of Technology[32][33].This work uses sets of two consecutive images taken by wide-view angle GoPro Hero2 camera(one with normal exposure and the other one under-exposed) and extracts their RGB[2], HSV[3] components. Then by learning several intensity ranges, they segment four zone type in each image: sun, blue sky, thin clouds and thick clouds. One sample of segmentation is shown in figure2.1.

[1] Ultraviolet B

[2] (Red, Green, Blue)

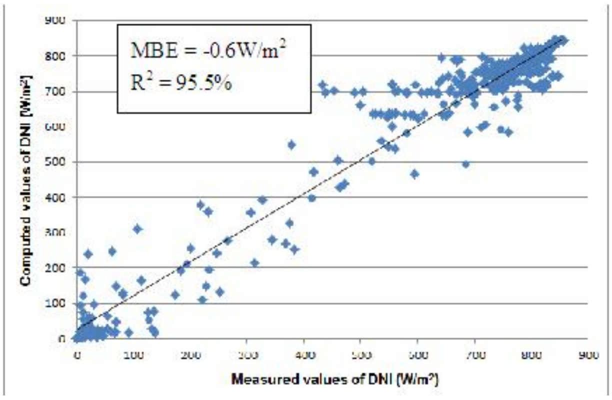
[3] (Hue, Saturation, Value)

Figure 2.1: Three different zones identified in images (sun, cloud, sky)



The irradiance (direct, diffuse, global) is recorded using the equipment Kipp & Zonen, Solys2 at the same time of image capturing. Finally, a regressor used to estimate direct irradiation (DNI) based on a feature vector consisting the number of pixels of different zone types in the images. The correlation in the result is shown in figure2.2.

Figure 2.2: Correlation between estimated DNI and recorded DNI.



2.2 Using clear sky irradiance model and binary cloud mask

The work done by T. Schmidt et al.[30] at University of Oldenburg in Germany is a very recent and relevant work on irradiance forecast using sky imager pictures. The experimental setup consists of a wide-view camera, one ceilometer (cloud base height sensor) located next to the camera, and a grid of 99 pyranometer distributed uniformly over 10km by 12km in the area close to camera. The aim is to forecast irradiance of every pyranometer up to 30 minutes. The training data is recorded from the pyranometers and the camera for two months every 10 seconds during daytime. In order to determine clouds projection on the ground, they apply a series of image processing algorithms.

2.2.1 Cloud detection

Firstly, they use Red-to-Blue Ratio (RBR) threshold for cloud detection which was first developed by Scripps Institution of Oceanography [14, 31] and is been used in many sky-imager-based forecast applications such as [9]. The RBR values close to 1 are usually cloud, and values very less than 1 are blue sky, since the blue channel which is in denominator dominates the red channel. However, since the RBR is not homogeneously distributed over the whole field of view, using a fixed global threshold for cloud detection brings a lot of misclassification for the areas close to the sun and also dark thick clouds or very transparent clouds. Therefore, they correct the RBR values based on clear-sky RBR values for each pixel. A Clear Sky Library (CSL) is created from images of one clear day. Then, the closest distance of current position and sun positions of CSL images is used to choose the reference RBR image map. This RBR map is used in correction formula 2.1 to decrease RBR threshold in circumsolar area to counter effect of sun saturation there. The correction also decreases RBR threshold for dark areas and increases it for bright pixels of image in order to detect thick and thin clouds.

$$R_{mod,i,j} = R_{orig,i,j} - R_{CSL,i,j} \times (a \times S - b \times (I_{i,j} - 200)) \tag{2.1}$$

Where $0 < S < 1$ is the average pixel intensity in circumsolar area. For more detailed discussion on results, they also apply a image-based cloud type classification using several visual cloud characteristics , and classify them into 7 different cloud types.

2.2.2 Image un-distorion

Since the raw image is from a fisheye lens, they apply a transformation to project it into geometric coordinates for convenience in other calculations. For that, intrinsic parameters of camera are determined using Scaramuzza Matlab toolbox [28] which solves a fifth-degree polynomial function of point-mapping between fisheye image and plain image. The extrinsic parameters are calculated as the best rotation which matches position of sun re-projection (derived mathematically) and sun position in the image. They calculate sun zenith and azimuth by using solar geometry2 algorithm[3].

2.2.3 Shadow mapping

In this step, shadow of cloud pixels are projected on the ground. For this, besides incidence and azimuth angle of every cloud pixel (which is derived using camera calibration function), cloud base height is needed. The cloud base height is estimated using a ceilometer for every point in time. However, to smooth th data, median of last 30 measurements is used. Even though the ceilometer supports multi-layer clouds as well, in this work they only use the lower-level cloud height. The distance of every cloud pixel to the camera is derived using $d_{i,j} = h \times tan(\theta_{i,j})$. Given the distance $d_{i,j}$, incidence angle $\theta i, j$, pixel's azimuth angle $\varphi_{i,j})$ and current sun position angles, horizontal distance of the cloud's show on the ground from the camera is calculated using Eq. 2.2.

$$\begin{aligned} dx_{i,j} &= h \times tan(\theta_{i,j}) \times sin(\varphi_{i,j}) + h \times tan(\theta_{sun}) \times sin(\varphi_{sun}) \\ dy_{i,j} &= h \times tan(\theta_{i,j}) \times cos(\varphi_{i,j}) + h \times tan(\theta_{sun}) \times cos(\varphi_{sun}) \end{aligned} \tag{2.2}$$

The shadow pixel points are mapped to a grid of 20km to 20km with resolution of 20m, and coordinates are interpolated if the shadow map resolution is lower than grid resolution, otherwise the central pixel of the dense shadow area is used for that grid point. Finlay, a Gaussian filter is applied to smooth the cloud edges for more realistic result.

2.2.4 Irradiance retrieval

Upon determining the shadowed and sunny grid points on the ground area of experiment, they use the histogram of clear-sky index (k^*) to estimate the GHI irradiance. The clear sky index is defined as ratio of measured global horizontal irradiance GHI_{meas} and a clear sky reference value GHI_{clear} (Eq. 2.3).

$$k^* = \frac{GHI_{meas}}{GHI_{clear}} \tag{2.3}$$

Clear sky irradiance is obtained from the mode of Dumortier [10] and turbidity values of Bourges [5] which is validated according to Ineichen's work [12]. For adapting to smooth changes of irradiance caused by factors other than clouds, this histogram is generated with measurements of last 30 minutes. As it is shown in figure **??** this histogram usually has two peaks which correspond to sunny and shadow states on the specified point of ground. Now, for every point on the ground, based on its state (shadow, no-shadow), the corresponding k^* is used from the peaks of the histogram to estimate GHI following Eq. 2.4. In case two distinct peaks could not be detected in histogram due to homogeneous irradiance condition, default values of 0.4 and 1, have been assigned for shadow and no-shadow states.

$$GHI = k^*_{hist} \times GHI_{clear} \tag{2.4}$$

2.2.5 Irradiance forecast

For forecasting the cloud map, they use optical flow algorithm (Lucas-Kanade) on cloud edges and corners (Shi-Tomasi method) in the past 2 minutes to extract clouds' motion vector. Then, by applying that motion vector to current cloud state, cloud map at different time horizons is estimated, and ray tracing from sun position at that times through cloud maps gives the shadow map on the ground. After determining shadow or no-shadow states for points on the ground, forecast GHI is estimated using Eq. 2.4 . They compare results of using GHI histogram from the nearest pyranometer station versus using only the pyranometer close to the camera as a representation for the whole area. These comparison is separately done for different cloud types, and the results shows for cumulus clouds using one pyranometer close to camera is enough to forecast irradiance for up to 2km radius. However, the forecast skill is highly varies depending on cloud types and overall does not do better than persistent model which uses median of past several minutes' irradiance.

2.3 Retrieval of direct and diffuse irradiance from sky images

In another recent work, T. Schmidt et al.[29] aims to estimate components of irradiance (direct, diffuse) instead of just GHI from the sky images using machine learning on image features. They hope this kind of irradiance detail helps in estimating GHI in cloudy and partial sunny states more accurately. The Experimental setup includes a fish-eye camera with sample rate of 10 sec and a pyranometers package next to it that records direct, diffuse and global irradiance every second.

2.3.1 Image features

As image features they calculate several local and global features including:

- Texture properties of the Grey Level Co-occurrence Matrix (GLCM)
- Color statistics (RGB space)
- Inter-color relations (e.g. Red-Blue-Ratio)
- Statistics of saturated pixels in circumsolar area in RGB and HSV color space
- Derived features like cloud coverage
- Solar elevation angle

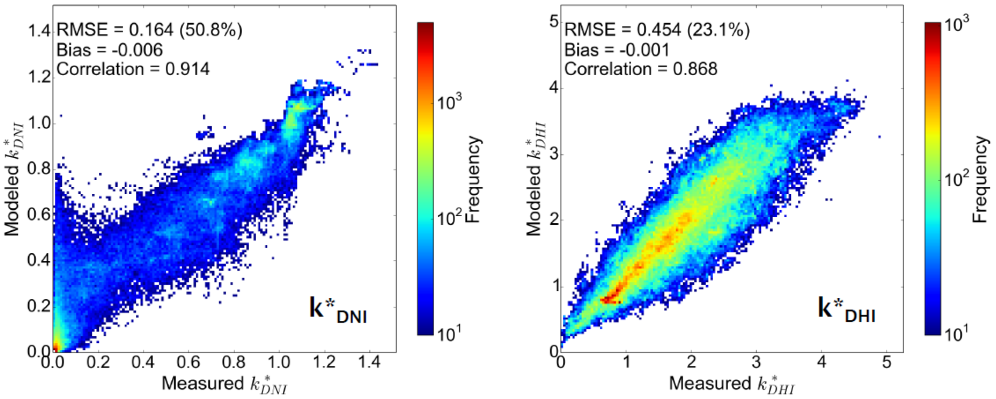
For prediction, two k-nearest-neighbor (kNN) models are trained that estimate the clear sky index of diffuse horizontal (k^*_{DHI}) and direct normal

(k^*_{DNI}) components which are defined as ratio of each component to their clear-sky values obtained from Ineichen's algorithm[12]. Since the initial feature list contained 37 features, for reducing computation time and avoiding over-fitting they apply a feature selection using decision tree feature ranking algorithm to choose the optimal feature set among them.

2.3.2 irradiance estimation

The result of KNN predication for DHI and DNI clear-sky indexes compared to measured values shows a correlation around .085 in Figure 2.3. In forecast applications, GHI can be derived from predicted irradiance components using 1.1. However, predictability of some of the used image features such as color statistics in this method is not robust enough. This leads to some errors in irradiation forecast.

Figure 2.3: Comparing estimated k^*_{DHI} and k^*_{DNI} to measured values. source:[29]



Chapter 3

Estimating Diffuse Horizontal Irradiance (DHI) from sky image

In this chapter, a new approach that we developed for estimating DHI from sky images is explained. First, our experimental setup and data is presented, then we talk about clear-sky model used here, and why estimating DHI is very important in predicting GHI. Furthermore, DHI estimation using the irradiance sensors and also sky-images is discussed. Afterwards, machine learning regression methods for obtaining DHI from sky-image are studies. Finally, the general strategy for power prediction in a photovoltaic power plant using the forecast irradiance components is proposed.

3.1 Experimental Setup

This study is conducted on one of the photovoltaic(solar) power plants operated by ABB company. This PV plant which is located in Cavriglia region in Italy is chosen as a pilot site for the "forecasting power prediction using sky-imagery" project. Therefore, it is equipped with the following instruments for recording irradiation and sky images:

- A customized wide-angle high resolution (4MP) camera system with a fisheye lens covering 185 degrees of field of view. The camera is in a packaging attached to a pole on rooftop of the building next to the site. Figure 3.1 shows the camera and its position next to the PV plates.
- Two GHI pyranometer (irradiance sensor); located close to the camera on rooftop. One of the sensors is horizontally facing sky, and the other one facing north with around 40 degrees angle to the horizontal plane. It's worth mentioning that the PV plates are titled to south with a fixed angle around 30 degrees to get more sun exposure. Th sensors are depicted in Figure 3.2.

- A thermometer for recording the temperature at the site.
- A PC which is connected to the camera, pyranometers and thermometer via their software interfaces in order to configure sample rates and store taken images and irradiance measurements. The data of power generated by the PV plant is also sampled and stored for every day. All the data recorded during each day is been automatically transfered to the company samba sever at midnight using a control software running on the PC.

Figure 3.1: wide-angle camera system used at Carviglia site

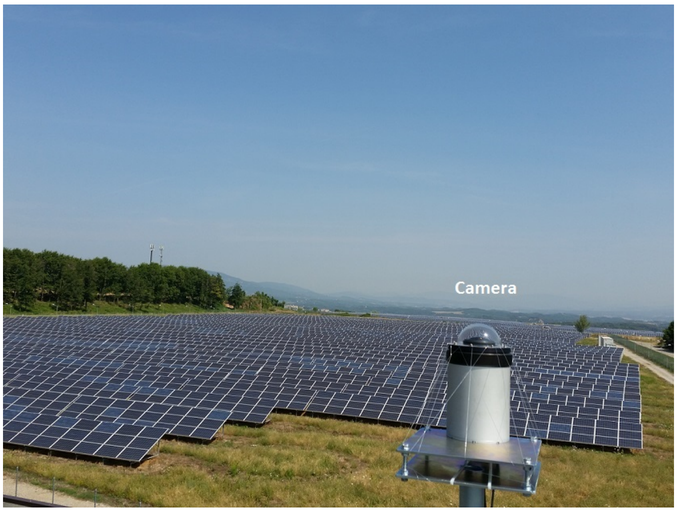
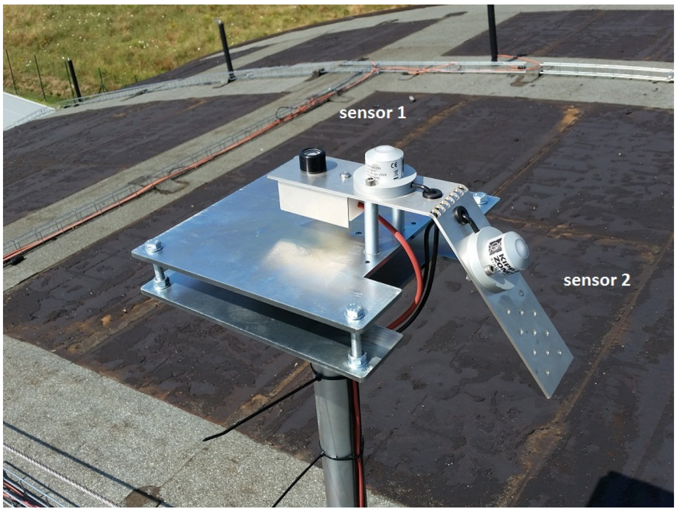


Figure 3.2: Two pyranometers (one horizontal, one 45 degrees titled to north) located close to the camera location



3.2 Acquired Data

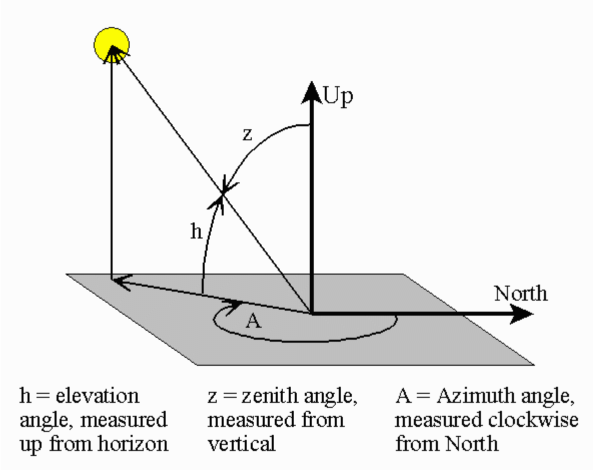
The camera system captures several images from the whole-sky every 8 seconds with different narrow exposure ranges. These images which are labeled according to capture time, are combined to create an HDR (High Dynamic Range) image for every sample time. The original narrow exposure images are generally deleted except the images at each hour time (i.e. around 7:00, 8:00, 9:00 etc.). Since capturing images at night is not useful for power prediction applications, camera is instructed to only take pictures during daytime (i.e. sunrise to sunset) which is obtained for that specific location for each day using mathematical models. Nevertheless, according to the captures images, this is not enough and still there are some black images taken at the minutes before sunrise and after sunset. Therefore, while processing the images on the application side, we exclude those images using a threshold on average pixel intensity of each image. This threshold is determined empirically. The images are further filtered against a sky mask which is been designed to exclude nearby mountains and buildings in the field of view and also limit the field of view to 170 degrees since transforming the points which are further in horizon is not accurate enough and the sun light is not negligible when the sun is in those points. It is worth mentioning that using HDR images in the image processing step is one of the key advantages of this study to related works specifically [29].

The pyranometers measure GHI values with the sample rate of 6 seconds. The temperature is also recorded with the same sample rate. However, the generate power is measured and logged every 3 seconds. These different sample rate make it necessary to interpolate the available data to find the estimated data for a time which there is no data available. Therefore, we can assign total irradiance, temperature and generated power to any given image using its capture time. This data acquisition setup has been running since 7th July 2015 until present. However, there are some short periods of time (usually lasting several days up to two weeks) which one of the sensors (pyranometers or the camera) had problems or the power plant was not working to produce power data. Considering the relatively small sample rate (less than 8 seconds), the amount of recorded data is big enough to make those off-days negligible in data processing steps. The data used in this study spans from 15th July to 10 February, meaning that many summer, autumn and winter days are available in the dataset to make it a good representation for the whole year.

3.3 Sun positions and sun states in image

Knowing position of sun is very important both in cloud segmentation and in irradiance estimation. First of all, since our images are from a wide-angle fisheye lens, they need to be transformed into geometric coordinates by an un-distortion algorithm to make them ready for further image processing steps including sun position, cloud segmentation and etc. This can be done by multiplying the raw image coordinates to camera transformation matrix which consists of intrinsic and extrinsic parameters of the camera system. As described in section 2.2.2, intrinsic parameters are calculated using image of a chessboard[28] and extrinsic parameters are estimated using Kabsch algorithm[15] based on position of the sun appeared in the image versus the expected position of sun in image. The theoretical sun positions are calculated for the location of our PV plant site at every image time-stamp using NREL algorithm [24] in Matlab represented in unit sphere polar coordinates. As shown in Figure 3.3 this position is described as two angles (zenith and azimuth) which are converted to Cartesian coordinates using sphere to Cartesian conversion and later on are scaled to the image size to correspond with an image pixel. That pixel will be assumed as center of the sun.

Figure 3.3: Sun position angles. source:[1]


Up
z
h
North
A
h = elevation angle, measured up from horizon
z = zenith angle, measured from vertical
A = Azimuth angle, measured clockwise from North

In cloud segmentation, which is not the focus of this study, sun position is used to treat pixels close to sun according to different threshold than other pixels. Furthermore, a sun state detection inspect the area around sun position to classify sun state in the image into following 4 categories:

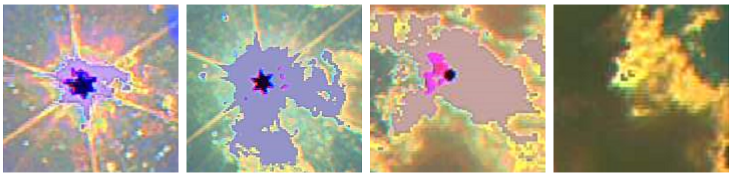
- sun_flag=4: indicating the sun is visible in the image and it appears as a star shape emitting 6 symmetric strong rays.
- sun_flag=3: indicating the sun is visible in the image and it appears as a star shape but with 5 or less symmetric strong rays.
- sun_flag=2: indicating the sun is visible in the image but it does not appear as a star shape. Instead, it appears as a small black dot with no

strong rays.

- sun_flag=1: indicating the sun is not visible in the image and it is either covered by clouds or the sun position is out of field of view in the image.
- sun_flag=-1: indicating there is an unexpected situation around sun position, for example star shape sun is detected far away from expected sun position which could be because of strange cloud formations.

One sample for each one of these categories is depicted in Figure 3.4. We are able distinguish between this states thanks to HDR images, otherwise this fine classification would not be possible.

Figure 3.4: Different sun states, from left to right: sun flag=4, 3, 2, 1 respectively.



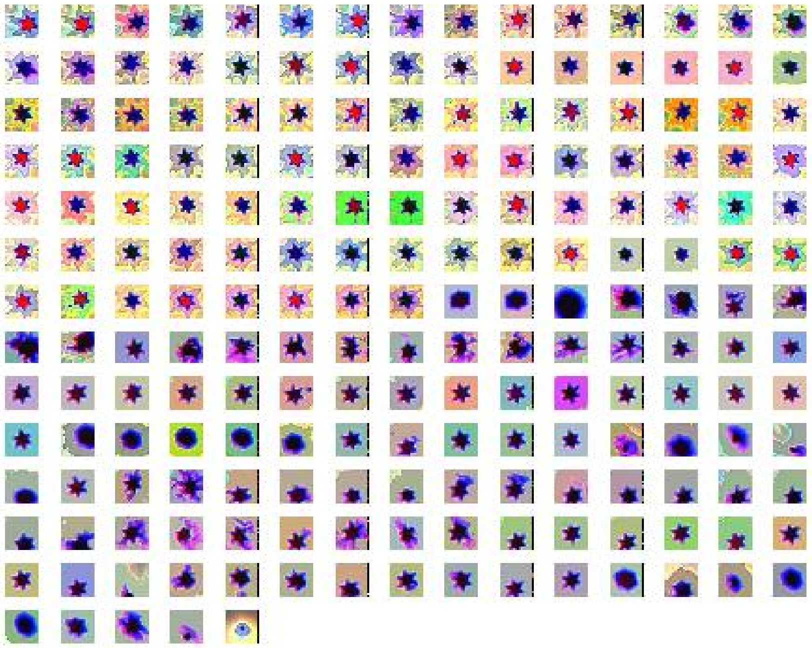
The variation of sun color in the images is so much that using Support Vector Machine for detecting sun is works very poorly. This issue is visible in Figure 3.5 which shows some clear sun samples from the images. Therefore, another approach using gray-scale images and geometrical symmetry detection is employed.

Using these sun states along with the sun position in cloud segmentation algorithm, will lead to better segmentation results close to sun that is particularly a difficult area for cloud segmentation due to highly saturated pixels with different sky or cloud colors. In irradiance estimation which is the main focus of this study, sun position is used to create two specific feature vectors in a bounding circle around the sun. These features are explained section 3.6 in detail.

3.4 Clear-sky irradiance model

Before dealing with the problem of irradiance estimation from cloudy images we should know first know how much the irradiance would be at any time in clear sky conditions when there is no cloud in the sky at all. Fortunately, there are a few number of models which provide irradiance components at each given time for many locations on the Earth. In this study we investigate two of the these methods, Ineichen [10] and McClear [18]. Both of these methods use location coordinates (latitude, longitude) and query time (consisting of year,day,month,day,minute,second) to calculate sun angles internally and return the irradiation based on optic formulas which

Figure 3.5: Variation of sun appearance in the images.



determine how much sun should reach the ground in as direct sunlight and how much should be scattered in hemisphere and forms the diffuse irradiance. The amount of scattered sunlight is varying throughout the year and also depends on the location, ground albedo[1] and aerosol parameters including pressure, ozone column content, water vapour column content, optical depth and Angstrom coefficient.

3.4.1 Ineichen method

Ineichen model statistically and physically relates some irradiance measurements to the aforementioned parameters as Linke Turbidity profiles which are available for different locations . We used the default Linke Turbidity values which come as a separate file in PV-Lib toolbox [17] in Matlab and is representative for most locations in Europe. However, one might need to use other appropriate Linke Turbidity profiles for other locations to get more accurate results.

[1] The fraction of solar energy (shortwave radiation) which is reflected from the Earth back into space. It is a measure of the reflectivity of the earth's surface. Ice, especially with snow on top of it, has a high albedo.

3.4.2 McClear method

On contrary to this approach, McClear uses a fully physical model that exploits recent aerosol properties, total column content in water vapour and ozone produced by the MACC project (Monitoring Atmosphere Composition and Climate). The MACC project, funded by the European Commission, uses data of many Numerical Weather Prediction (NWP) centers distributed around the world to provide a global aerosol property forecasts together with physically consistent total column content in water vapour and ozone. In other words, since McClear uses synthetic data of NWP's, it does not depend on any local atmospheric observations for irradiance prediction. For the sake of speed, McClear irradiance estimates are pre-computed for the the location of these measurement centers and are interpolated for all other point on th Earth using a look-up table approach. Of course the closer we are to one of these measurement centers, the more accurate McClear estimate will be. The McClear irradiance estimates are available worldwide for every minutes from 2004 to present with 2 days lag under this web service [2]. This means that if we want to get the irradiance for today, we need to interpolate data of several days or weeks before to get an estimate for the current time. Nevertheless, one can use the original data of past 2 days for current time as well, since irradiance does not change considerably in two days.

3.4.3 Comparison of clear-sky simulation results

To evaluate performance of these two clear-sky models we choose several days which have a significant clear part during the day (just because complete clear days are very rare). The chosen days should be from different months of year in order to represent performance of models for the whole year better. After observing the irradiance logs and images for verification, the following days were selected for comparison: 2015/07/19, 2015/08/03, 2015/09/21, 2015/10/24, 2015/11/24 and 2016/02/05. As an example, simulated GHI of three days are plotted in Figure 3.6 next to the observation.

As it can be seen, both methods can predict the shape of irradiance curve very accurately around the year, but both have biases in the result such that the simulated values are always smaller than observed ones. During the summer days, McClear and Ineichen results are biased almost equally, and as we get closer to winter days, the bias of McClear gets smaller and bias of Ineichen get slightly bigger. Figure 3.7 shows the correlation of simulated GHI of both models plotted with respect to the actual measurements of all of the examined days.

This plot again shows that the bias of McClear and Ineichen for large values of GHI is the same, and for small GHI values McClear result is closer to the observed irradiance. however, it also suggests that since Ineichen methods

Figure 3.6: GHI estimation of McClear vs Ineichen vs observation

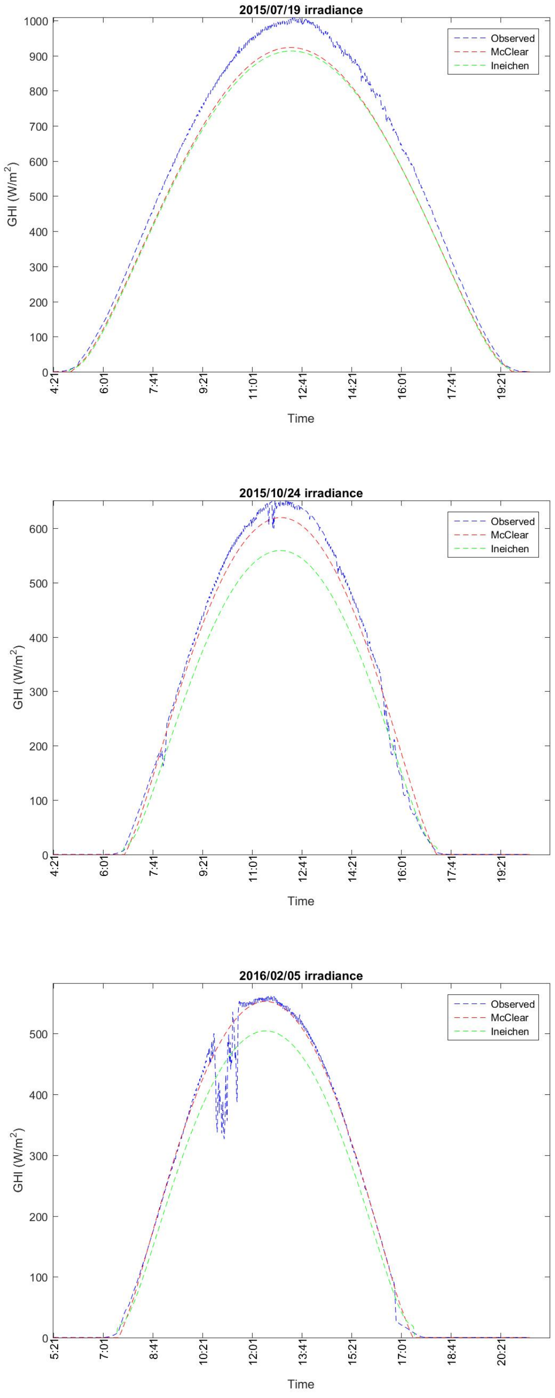
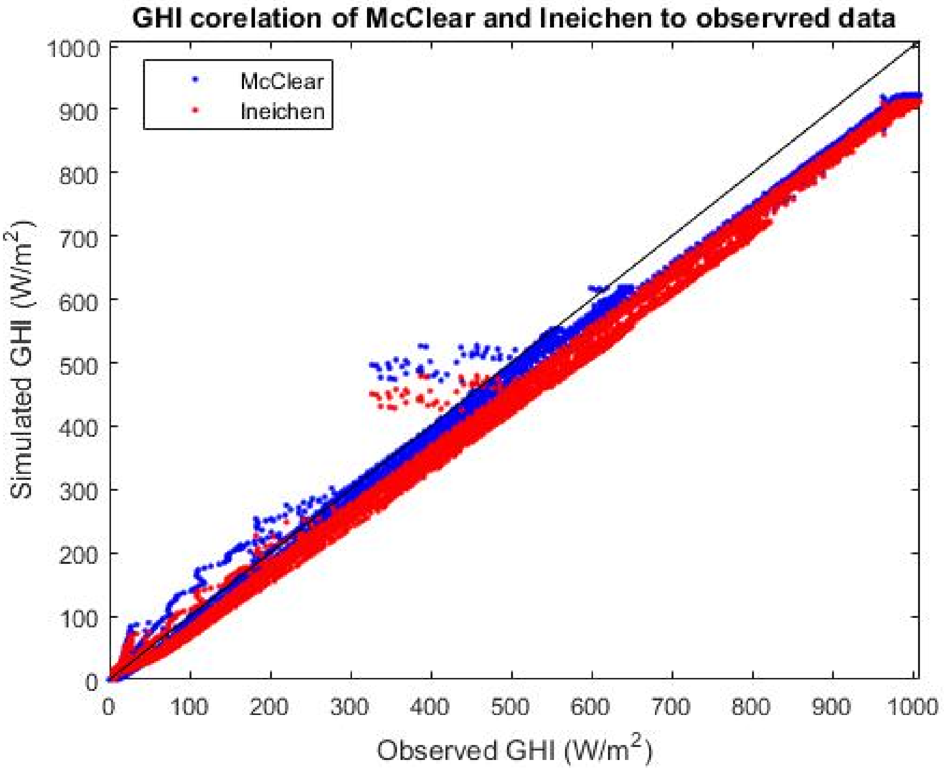
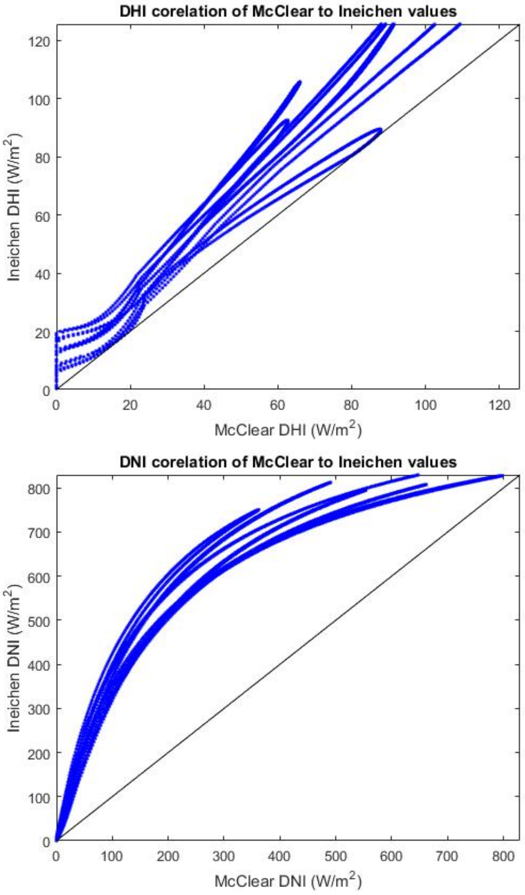


Figure 3.7: Correlation of GHI simulation of McClear and Ineichen with observations on selected days



has lower variation in terms of bias, this bias can be compensated with a scaling factor more easily than the bias of McClear which shows larger variation throughout the year. Note that, the outliers in this figure are representing cloudy times. Furthermore, this bias of Ineichen method is strongly related to the Turbidity factors that are neglected in our case by using the default values. It would be not surprising to see smaller bias if one uses Turbidity factors which are verified for the location of PV plant. The correlation of DNI and DHI values of both methods are illustrated in Figure 3.8. One can see that DNI is predicted much higher in Ineichen method and DHI is also simulated slightly higher than McClear values. This behavior is intensified during autumn for DNI, but it is not varying a lot for DHI. Since we do not have observation values of DHI and DNI, we cannot compare correlation of methods' results to the observed values in this figure. However, we can hypothesize than during a complete sunny day, if at a very short time (i.e. seconds) a cloud covers the sun completely, the direct irradiance (DHI) is almost zero and all the irradiation only comes from DHI source which is the scattered light in the sky. Therefore, we can use our GHI irradiance observations as DHI and compare it to the DHI simulated values for that particular time.

Figure 3.8: DNI and DHI correlation of McClear to Ineichen



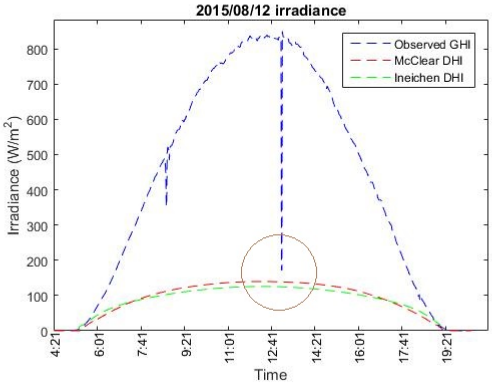
DHI corelation of McClear to Ineichen values

DNI corelation of McClear to Ineichen values

In Figure 3.9 this has been shown for a moment around 13:00 which sun is occluded by a small thick cloud and therefore, GHI is dropped rapidly to a value close to 150 which is very close to simulated DHI values from McClear and Ineichen methods. Also, there is no other cloud in the sky to influence the DHI component. Furthermore, the simulated DHI values are very close to each other at any time and resemble the shape of GHI relatively well during whole day. Thus, we can conclude that these models can predict DHI with good accuracy, and since DNI is a function of GHI and DHI according to Eq 1.1, DNI values calculated from McClear and Ineichen models are also accurate enough and for our application. This hypothesize has been verified by looking at many other data points where GHI observation gets very close to DHI simulation values and there is a cloud obstructing the direct sun light.

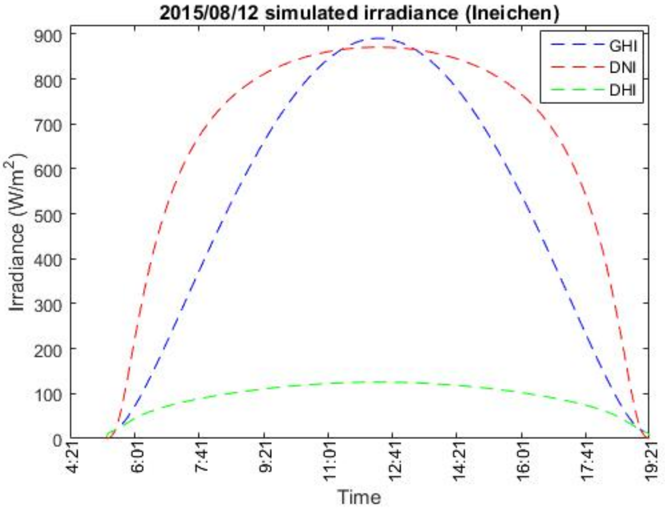
We know that DNI should be always larger than DHI during clear-sky con-

Figure 3.9: Comparison of DHI of McClear and Ineichen to observed GHI



dition. Looking at Figure 3.10, pattern of changes in the simulated values of DNI and DHI during a day confirms this condition too.

Figure 3.10: Variation of irradiance components during day



3.4.4 Choosing the clear-sky model

As we discuss in this chapter, McClear and Ineichen models both predict the clear-sky irradiance components relatively accurately,however the bias for Ineichen method is more robust and manageable than McClear bias. Furthermore, obtaining the McClear values requires downloading the irradiance files from their web service since there is not offline library for calculating them. All this considered together, we decided to use Ineichen as the clear-sky irradiance model for this study. The scaling factor for compensating the bias in Ineichen is set to 1.08 which is empirically calculated based on several clear day observations.

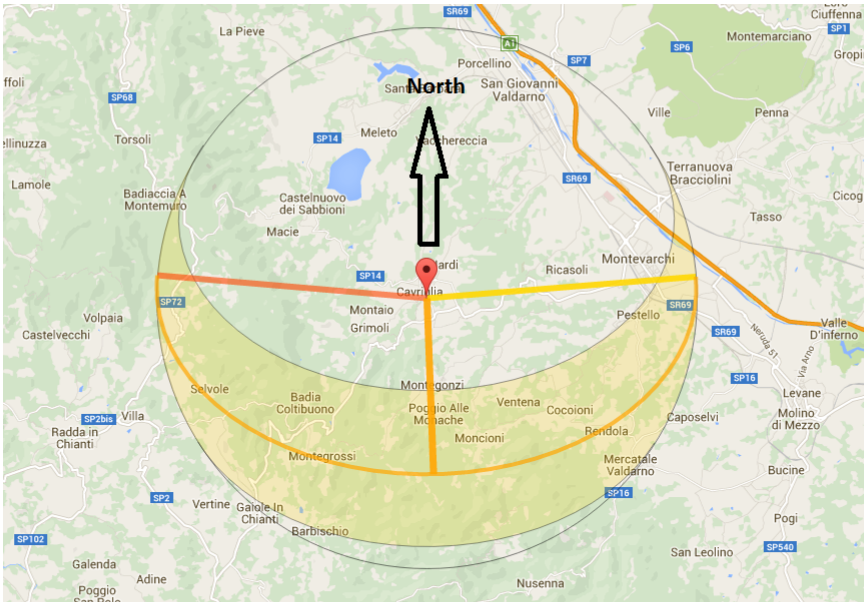
3.5 Estimating diffuse from pyranometers

After picking the suitable clear-sky model, we can predict the irradiance components for any given day in clear-sky conditions, but for determining the irradiation in cloudy conditions we need to first detect them and then find a relation between cloud states in the sky and the observed irradiance. Since GHI is composed of DHI and DNI, and because we want to get some hints from the images for determining GHI, the reasonable approach is to first estimate DNI and DHI values and then construct the GHI from them using sun zenith angle and Eq 1.1. Thus, for the learning algorithm we need DNI and DHI observations to relate them to image features. Ideally, we would have DHI and DNI observations separately along the GHI values from an advanced irradiance sensor[2], but in our experimental setup only GHI values are observed in horizontal and 40 degrees north. The idea behind putting one of the pyranometers tilted towards north is that based on sun path from an observant in that regions of the world, tilted surfaces toward the north do not get direct sun light during most of the day. That's why the PV plates are tilted towards the south to get more sunlight. The sun path at Cavriglia can be seen in Figure 3.11. There is a simple geometric rule that confirms this idea. If the angle between sun and the normal vector of a surface is greater than or equal to 90 degrees, sun rays which are coming straight from sun will not hit the surface. This condition will hold for many times during a day for the sensor with around 40 degrees tilt toward north.

Anyways, the sun path changes during the year, for example during the summer two edges of this path come higher towards north, and in winter they go lower towards south. This brings some direct sunlight to the tilted pyranometer in early morning and late evening during summer time. However, during winter and most of the autumn and spring, the tilted sensor is in complete shadow and does not receive direct sun irradiance (DNI). As a result, we can safely assume that during those times, the values recorded by this tilted sensor represent the diffuse irradiance component. This can be seen more clearly in Figure 3.12. In this picture, the dark blue curve represents irradiance observations of tilted sensor (sensor 1) during a sunny summer day -3th August. As one can see, around the morning and evening when the sun has a shine on the sensor, there is a bump in the irradiance.

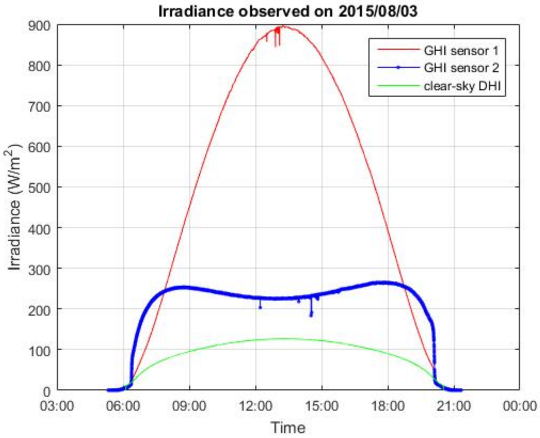
[2] This type of pyranometers such as Zonen CM11 have a dynamic shade-band along the sun path to make sure that the sensor is always in shadow, this recording only DHI component. Then DNI can be computed from GHI and DHI or by using another type of pyranometer such as Eppley NIP.

Figure 3.11: The sun path during a spring day at Cavriglia, Italy, location of ABB PV plant.



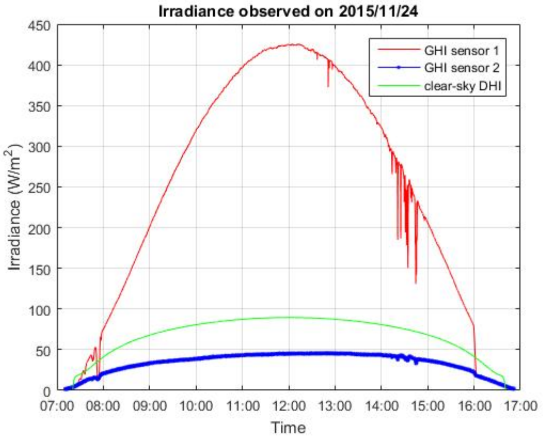
However, during rest of the day that the sensor goes into shadow, the irradiance is close to clear sky DHI. As we expect, during the winter the tilted sensor will be always shadowed, thus should not have any DNI part in its measurement during the whole day. Looking at Figure 3.13 confirms this idea, since there is no bump in the morning and evening records.

Figure 3.12: Irradiance comparison of horizontal sensor (1) and tilted sensor (2) for a summer day



Therefore, if we deduct the direct irradiance component from sensor observations, we can obtain diffuse component which is required for our learning algorithm. We previously have shown in 3.4.3 that our DHI and DNI values

Figure 3.13: Irradiance comparison of horizontal sensor (1) and tilted sensor (2) for a winter day



obtained from clear-sky model have enough accuracy for this application. But for calculating the effective direct irradiance on tilted sensor we need to find the angle between sun and surface of the sensor at any time, since this angle is different than sun zenith. According to Figure we can use sun azimuth and sun zenith angles to locate the sun vector. The normal vector of tilted sensor surface can also be defined based on its zenith and azimuth angles. Then using this linear algebra equation 3.1 we can find the angle between two vectors in 3D space.

$$\theta = Arctan2(\|a \otimes b\|, a \cdot b) \tag{3.1}$$

where Arctan2 is four quadrant arctangent of the elements.

However, for the horizontal sensor the effective direct irradiance can be simply computed as $DNI \times cos(zenith)$. We have shown in Figure 3.8 that DNI is almost zero(0) when the cloud occlude the sun completely. This case correspond to $sun\_flag = 1$ as we explained in 3.3. And if the sun is completely shining like a star, hence resulting in full DNI, it corresponds to $sun\_flag = 4$. For other sun states where the sun is partially obstructed, the DNI is not the same as clear sky DNI. We hypothesize that the DNI for sun-flag=2 and 3 is somewhere between 0 and clear-sky DNI. However, for the sake of simplicity we do not consider these states in this study. In other words, we only try to estimate the DHI for the conditions that either sun is completely occluded ($DNI = 0$) or it is completely visible and shining(i.e. $DNI = DNI_{clear\_sky}$. With this simplification we can compute the DHI for each irradiance sensor by re-writing Eq 1.1 as:

$$DHI = \begin{cases} GHI - DNI \times cos(angle), & \text{if } sun\_flag = 4 \\ GHI, & \text{if } sun\_flag = 1 \end{cases} \tag{3.2}$$

where GHI is the observed total irradiance by the sensor, $angle = \theta$ for tilted sensor and $angle = zenith_{sun}$ for the horizontal sensor. Note that our experiment dataset is pruned to only contain image samples from these two sun-flags. Furthermore, we do not consider images for early morning (i.e before 8) and very late evening (i.e. after 19) for our training, since the GHI in those moments is very low -less than 100- which makes them insignificant for power generation and not interesting for our application. Plus that the pattern of sun in the horizon widely varies and result in unusual errors in our algorithms. Since the exact tilt angle of sensor 2 and its azimuth to the north is not given for sure, we need to evaluate a range of possible values to find the angles that bring results with less error during sunny days compared to DHI_{clear_sky}. The optimal angles found to be **57.5** degrees for zenith and **-0.7** degrees for azimuth. In Figure 3.14, calculated DHI values for a sunny summer and a sunny winter day are shown.

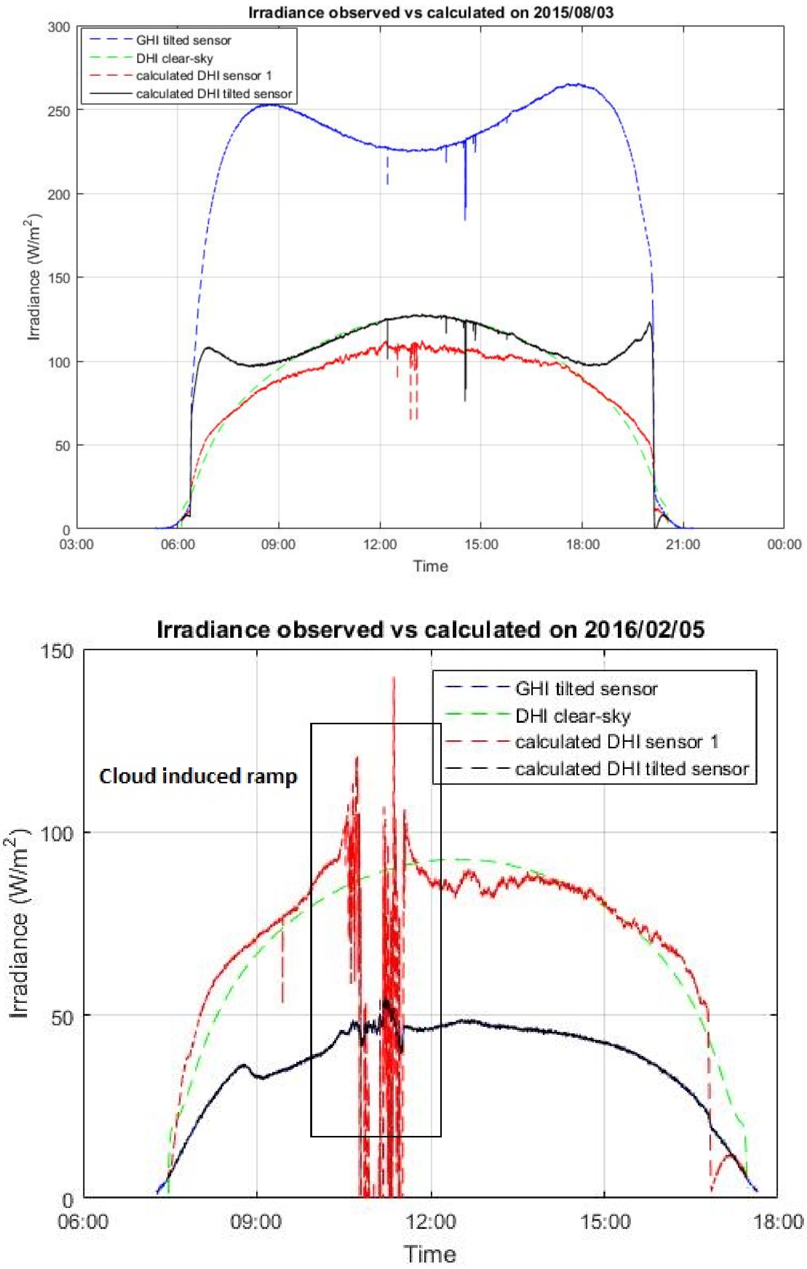
Interestingly, one can see two horn shape anomalies in the morning and evening on calculated DHI of a summer day. One hypothesize is that reflection of light from nearby mountains are causing this extra irradiance which is no accounted for in our model. Anyways, as we can see the DHI values of tilted sensor show more robustness compared to values of horizontal sensor. The reason is some complex effects of clouds on DNI which we are not considering so far. The tilted sensor is more immune to this DNI variation. Even though during the winter days, the calculated DHI for horizontal sensor is closer to clear-sky DHR, but the DHI of tilted sensor shows more robustness to cloud condition variations. Therefore, we decided to use DHI of titled sensor as actual DHI for both sensors on further calculations.

We also experimented reconstruction of observed irradiances for both sensors based on calculated DHI values (from tilted sensor), assumed DNI values and respective angles using Eq 3.3 .

$$\begin{bmatrix} GHI_1 \\ GHI_2 \end{bmatrix} = \begin{bmatrix} a_1 \times cos(zenith) & b_1 \\ a_2 \times max(0, cos(\theta)) & b_2 \end{bmatrix} \times \begin{bmatrix} DNI \\ DHI \end{bmatrix} \tag{3.3}$$

where GHI_1 is irradiance reconstructed for horizontal senor (1), GHI_2 is the tilted sensor irradiance, θ is the angle between sun and tilted sensor normal, a_1, a_2, b_1, b_2 are constant values needed to be tuned for the best fit in result. For reconstruction we evaluated different values in range of 5 to 60 to find zenith and azimuth of the tilted sensor which result in best fit. Also, for the constant parameters the following values worked best: $a_1 = 0.9, a_2 = 1, b_1 = 1.14, b_2 = 0.78$. The optimal angles found to be **56.5**

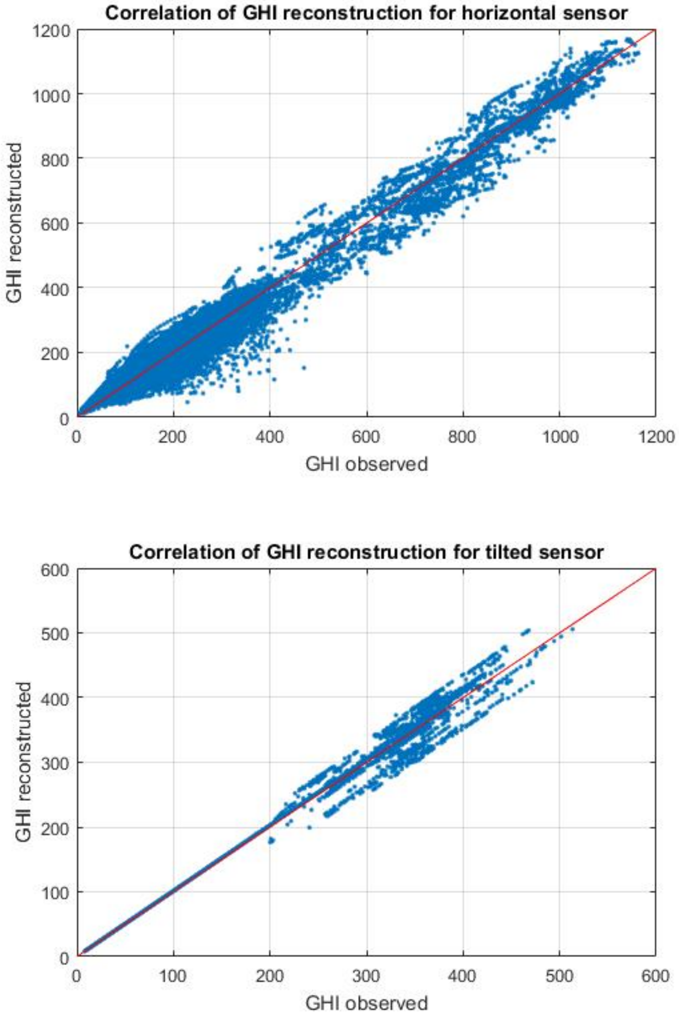
Figure 3.14: Calculated DHI



degrees for zenith and **11.5** degrees for azimuth which is different for the azimuth we used for obtaining the DHI values before (i.e. -0.7). However, we are only interested in reconstructing GHI for the horizontal sensor which does not depend on tilted sensor azimuth according to the aforementioned equation.

The correlation of reconstruction result to actual GHI values is shown in Figure 3.15 and proves that GHI values of horizontal (and tilted) sensor can be estimated using a simple DNI model and DHI of tilted sensor.

Figure 3.15: Calculated GHI based on DHI and DNI for both sensors



The hope was that by using the tuned version of Eq 3.3 and feeding it with GHI measurements of both irradiance sensors (GHI_1 and GHI_2) , we would be able to estimate DNI and DHI by multiplying GHI to inverse of coefficient matrix. In other word, we would be able to determine DHI and DNI for any situation including sun-flag=2 and sun-flag=3 which we excluded earlier for convenience. However, the results of this experiment was not satisfactory to hold our hypothesize. This could partially be due to the fact we are using DHI of tilted sensor instead of DNI of sensor 1 in this equation. Anyways, the result of DHI and DNI for sun-flag=1 and 4 is still valid and we should now try to estimate these values from images.

The idea is that as we limit the effect of clouds on DNI by assuming $DNI = 0$ or $DNI = DNI_{clear\_sky}$ in occlusion or not occlusion situations, the only way that clouds can affect GHI is through DHI variations. In next section we investigate characteristic of clouds in images to understand this effect better.

3.6 Key factors influencing DHI

As we have shown in Figure 3.10 all the irradiance components including DHI follow a bell-shape curve during a clear day and this curve varies throughout a year -lower in winter, higher in summer. We call these factors, **non-image** features as they are not obtained from images and are independent of cloud characteristics. We formulate these features as following:

- $DHI_{clearsky}$
- Zenith angle of the sun; between around 20 up to around 80 degrees

Even though $DHI_{clearsky}$ is dependent on zenith angle, but experiments show including both in the features list is beneficial.

Besides non-image features, we extract some visual features from sky images to help estimating DHI in not-clear situations. For image-based features there are several intuitions about what factors might affect DHI most. These includes cloud coverage, cloud type, shininess of sun, cloud color and etc. However, using cloud color features is not a good idea for our application, since they vary a lot even in very short-time ahead and predicting cloud color is not accurate enough. And since the aim of this study is to forecast irradiance for predicted cloud states in several minutes ahead, we restrict our image-based features to highly predictable features. This includes:

- sun-flag; as we explained in section 3.3, it is an integer number between 1 and 4. However, we only consider states 1 and 4.
- Semi-local cloud coverage; defined as the percentages of cloud pixels (i.e. image pixels which are classified as cloud) in four parts of the image separately, i.e. top right, top left, bottom right, bottom left. For each region, 100% indicates full cloud coverage and 0 means no cloud. More information is given in section 3.6.1.
- Cloud coverage around the sun; defined as the percentage of image pixels in a small circle around the sun which are classified as cloud. This feature is explained in more detail in section 3.6.1.

- Saturation factor; defined as the percentage of saturated cloud pixels (i.e. with very high brightness). For complete description of this feature refer to section 3.6.2.

3.6.1 Cloud coverage features

For designing a cloud feature we took several issues into consideration. Firstly, this feature should be not too sensitive to the position of clouds, since clouds are very dynamic and position sensitivity will lead to not consistent feature vectors for actually similar cases in terms of DHI. Secondly, the relative position of clouds should be taken into account in the feature, for example clouds which are close to sun are most probable to reflect the irradiation than clouds in other parts. As a middle-ground for this two conditions we initially designed a feature vector that is sensitive to clouds position but not too much. For that, the image is divided into four equal part by connecting the central point to middle of each side. Then, a feature vector with four elements is created consisting of cloud coverage percentage for each part. However, this feature does not put enough emphasis on cloud variations around the sun which is desired for reflection situations. Therefore, another feature is created for cloud coverage in a circle around the sun. Since this circle is small (40 pixels radius), position of clouds is not likely to cause considerable difference on DHI, thus we calculate the cloud percentage in this circle around the sun as a one value feature between 0 to 100. Note that pixels that are classified as neither cloud nor sky, are most probably sun. Therefore, these pixels are not included in total number of pixels when calculating the cloud coverage percentage. The intuition is that cloud coverage around the sun is more important that total cloud coverage for DHI, since those clouds usually reflect more sunlight to the ground. The areas for extracting cloud coverage features are illustrated in Figure 3.16.

3.6.2 Saturation factor

In some images that sun is occluded by a relatively thin cloud, we can see some pixels around position of sun, which are in fact part of the cloud but are very much illuminated by the sunlight which is passing through them. One example of this effect is shown in Figure 3.17.

Even though, the sun is not directly visible int these images, resulting in DNI=0, our DHI observations show a considerable improvement in such cases. Therefore, we hypothesize than measuring this saturated pixels can be valuable feature vector for DHI estimation. Depending on the cloud type, and sun position these saturated areas can be predicated in very close future. That makes such feature valid for our forecast application. For calculating this feature we visually inspected many samples and realized that in all of them there is a very hight brightness band around the saturated regions

Figure 3.16: Regions of interest for extracting cloud coverage features

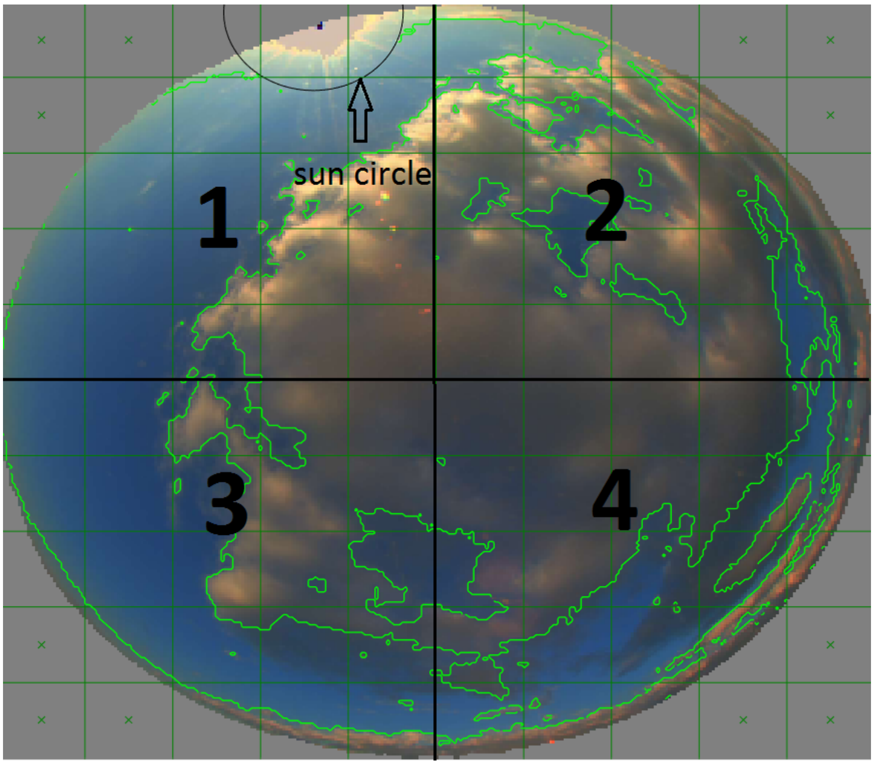
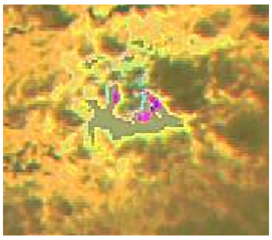
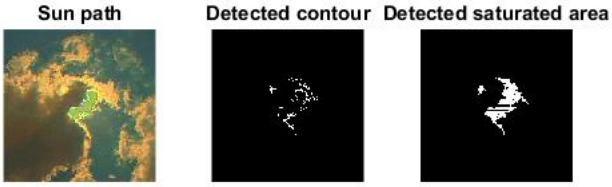


Figure 3.17: One example of saturated cloud pixels



which is even brighter than the saturated pixels. Based on this observation we designed the following algorithm for measuring saturation factor in any image. Firs, we crop an 120X120 pixel area around the sun as the only place saturation can occur. We convert this RGB patch into gray-scale and then find the pixels with intensity values more than 215. This threshold is been set empirically. These pixels are representing a contour for the saturated area. Since there are some discontinuity is this contour, we do the following approximation to find area inside. Every pixel in the patch is considered inside the contour, if that pixel or one of its side-neighbors (i.e. right or left) is bounded by at least 3 contour pixels in different directions (top,down,left,right). The results show that approximates saturated area is very close to actual area in the images. Finally, the saturation factor feature is calculated as the percentage of saturated pixels with respect to total patch pixels. This percentage is bounded to 60% to comply with visual observation and reduce effect of errors in the detection algorithm. One example of detected saturated area is depicted in Figure 3.18.

Figure 3.18: Detected saturated area in a patch around sun



Our final feature vector for each sky image consists of all the non-image and image-based features, resulting in a 9-element vector. Due to time limitation, we did not investigate cloud type or cloud texture pattern features.

3.7 Dataset

As it was mentioned in section 3.2, we have data records and images from August 2015 until present for every 8 seconds during the day. Since we created our feature vector dataset in February 2016, the time span of images are from August 2015 to February 2016. As we showed in Figure 3.14, HDI calculated values in the morning and evening are not accurate, therefore we skip these times in our final dataset. Furthermore, since estimating DNI for images with sun-flag 2 and 3 is too difficult and there is not ground-truth information for verifying results, we decided to restrict our data to only images with sun-flag 1 and 4, which corresponds to sun not visible and complete shining sun visible, respectively. This restriction reduces our usable data sample around 30%. We also prune our data set to remove data samples that are too close to each other both in time and cloud conditions. For that, we iterate through images of each day for sampling a data set, but change the sampling rate by amount of change in cloud coverage from last sampled image to current one. This means that if there is long steady sunny condition during a day, we only sample handful of them. The same goes for continues full cloud coverage times during a day. On the other hand, when the cloud condition is changing a lot, we take more data samples to diversity our dataset and include as many variation of cloud situation as we can. The final data set after all these pruning, includes around 44,000 data samples from many different cloud conditions and many days from summer to winter.

3.8 Learning the relation between image features and DHI

In previous sections we approximated DHI from the pyranometers and also created a feature vector representing an image. Now, we need to find a way to relate them to each other in order to be able estimate DHI values from a given feature vector. Since DHI values are real numbers, this is a regression problem. Therefore, for we used several regression methods for solving that to find the best performing regressor on this particular problem. Because the domain value of our feature elements is different, we normalize every feature to [0,1] range before feeding it in any regressor algorithm.

3.8.1 Linear Regression

We tried linear regression to determine if DHI can be written as a polynomial function of feature vectors lile in Eq 3.4.

$$y_i = \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \varepsilon_i \tag{3.4}$$

Where y_i is DHI and x_i is a feature vector. Also, square (x^2) and square root ($\sqrt(x)$) of features are added to the feature list to evaluate non-linear relation of features to target as well. To consider mutual relation of features, interaction of features (i.e. $x_i x_j$) were added to final feature list, resulting in 63 element feature vector. To prevent cluttering the features any further, we did not evaluate logarithmic or exponential of features.

3.8.2 K-nearest-neighbors (K-NN) Regression

Another approach that we tried was K-NN regression. This method aims to find unknown target value of a feature vector by calculating a weighted average of target values from K training feature vectors that are most closest (i.e. neighbors) to the query feature vector in the feature space. In other word, K-NN assumes that if two feature vector are very similar to each other based on a similarity measure, then their target should be similar too. And this is a valid assumption for our application. These neighbors of input vector are weighted by the inverse of their distance for averaging. The distance between feature vectors can be defined arbitrarily, but popular distance choices for continues variables are Euclidean, Manhattan and Minkowski. We experimented all three of these distances to find the one with least error. To reduce the noise in data, usually the number of neighbors (K) for averaging should be 10 or more. However, the best K can vary depending on the data set. Therefore, we have tried several values to tune this parameter.

$$d = \sum_{i=1}^{k} |x_i - y_i| \tag{3.5}$$

$$d = (\sum_{i=1}^{k} (|x_i - y_i|)^q)^{1/q} \tag{3.6}$$

3.8.3 Support Vector Machine Regression (SVR)

Even though Support Vector Machine[8] is best known for classification problem, but one can use a modified version of that for regression too with almost all the benefits of normal SVM. In SVR formulation two maximum margin hyperplanes are found between negative and positive errors between a polynomial function of input features and the target values. In the soft-margin SVM which we are using, the ε parameter helps to tolerate small errors and find better support vectors. Slack variables help to ensure existence of a convex solution as well as reducing the effect of noise in data. Kernels in SVR map the feature space to a higher dimensional space that might be more suitable for finding those separating hyperplanes. Since we are working with images and there is a lot of noise in features as well, a Radial Basis Function (RBF) kernel is used which can help smoothing these noise and find better hyperplanes. The gamma parameter (γ) in RBF kernel defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

3.8.4 Training and test datasets

For every learning algorithm we need to have a separate training and testing dataset. Since usually Linear Regression and K-NN need large training set size to avoid over-fitting and also smooth the noise in data, we dedicate 40% of our data samples for training set, and the rest for testing. This means the training dataset has around 18,000 data samples and test dataset has around 26,000 data samples. For the case of Support Vector Machine Regression, there is no need for such a big training size due to ε and slack variables as well as kernel non-linear mapping. However, for the sake of comparability of results, we use the same training and test sets for all three methods. The distribution of DHI (target) values in our dataset is not uniform. There are many more samples at low values of DHI than high values. Therefore, to make sure that there is enough sample from every DHI range in the training and test sets, we partition the DHI values into 5 ranges, [0,100], [100,200], [200,300], [300,400], [400,500]. Then in any range we choose $N/5$ of samples randomly for training where N is total number of training. If the number of samples in a range is less than $N/5$ we take 70% of the sample in that range as training and the rest as test. Using this sampling trick, we provide enough samples from every DHI range in order to be able to estimate values in that range later with regression. In next chapter, we show some results of this three regression method on test set after being trained with training data.

Chapter 4

Results and Discussion

In this chapter, we show results of applying the regression algorithms on our dataset for estimating DHI (Diffuse Horizontal Irradiance) from sky images. As the metric for comparison between methods, Root Mean Square Error(RMSE) is used. We will also discuss where the algorithms perform poorly by showing cases.

4.1 Feature Selection

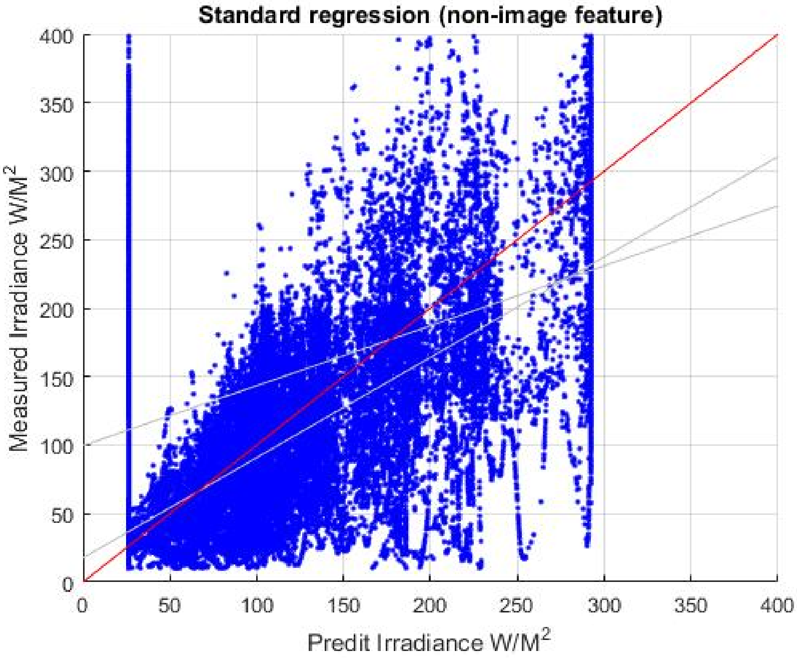
Since the correlation of each feature element to DHI is not strong enough, for we need to do a feature selection in order to select the most relevant features and discard the ones that are not helpful enough. For that purpose, we use a Forward-Selection algorithm for each method. This approach assumes an empty feature vector initially, and at every iteration evaluate the error of estimation after adding each one of the available feature elements to current feature set. The feature set with least error will be chosen and removed from available feature elements. However, the difference of last error and current error has to be more than a threshold for adding that feature elements. This helps to exclude random improvements caused by features and don't add them to final feature set. This threshold has been set empirically to 0.7 in RMSE -60% of inverse of chi-square cumulative distribution function for 1 degree of freedom.

To verify the advantage of using image features along non-image features in the feature vector, we evaluate these two cases separately by each regression method. The non-image feature set consists of [clear-sky-DHI, sun-zenith]. The full feature vector includes: [sun-state, saturation factor, cloud coverage of region 1, region 2, region 3, region 4, clear-sky-DHI, sun-zenith, cloud coverage of circumsolar area]. Also for diminishing the effect of randomness in training data selection, we repeat each experiment configuration three times and average the results.

4.2 Linear Regression

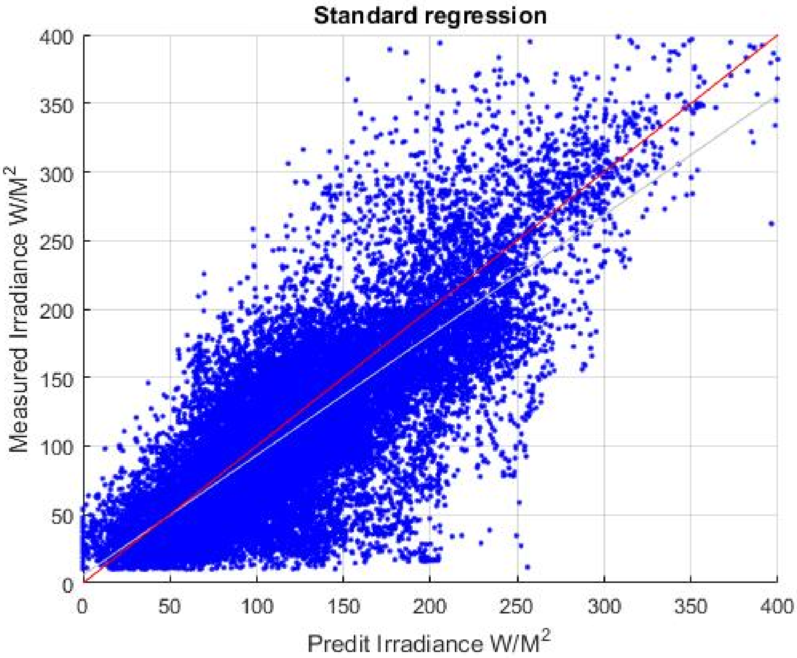
As we mentioned in 3.8.1, feature list for linear regression includes square and square root of base features too in order to account for non-linear relation of DHI to one of base features. First, we evaluated performance of non-image features for DHI estimation using linear regression. Feature selection algorithm suggests that the only important feature in non-image list is clear-sky DHI value. The RMSE in on test data set was 60.7 W/m^2 which is around 16% of range of DHI values, [30,400]. This indicates a poor performance for our application. The RMSE of the same configuration for training data was 64.4 which is even bigger than error on test data. However, normally we expect to see lower errors on the training set. This might suggest that the features are not representing intrinsic characteristics of data well. Figure 4.1 shows correlation between result of linear regression and target DHI values for non-image features.

Figure 4.1: Linear regression result using only non-image features



In next step, we included all the features (non-image and image-based) for regression training. The feature selection algorithm only chose 12 features out of 27 features (i.e. $9 + 9 + 9$ for base features, square and square root). The selected features are indexes of [1 2 4 5 7 11 14 17 20 21 22 23] in the feature vector. This indicates that there is some non-linearity in the relation of features to targets. The result of linear regression on this feature vector shows an RMSE of 44.7 W/m^2 on the test data set and 47.4 for training set. The result for only using image-based features is always worse than non-image features by a distance of around 5%. Thus, it is clear that there is a considerable improvement (26%) in regression performance when using both non-image and image features. The reason for performing slightly better on test data can be indicating that Linear regression can't model this dataset well or there is a bias in errors of training. We also have tried a training set two times bigger than test set, but the same pattern was observed. The correlation of estimated DHI to target values is illustrated in Figure 4.2.

Figure 4.2: Linear regression result using all the features

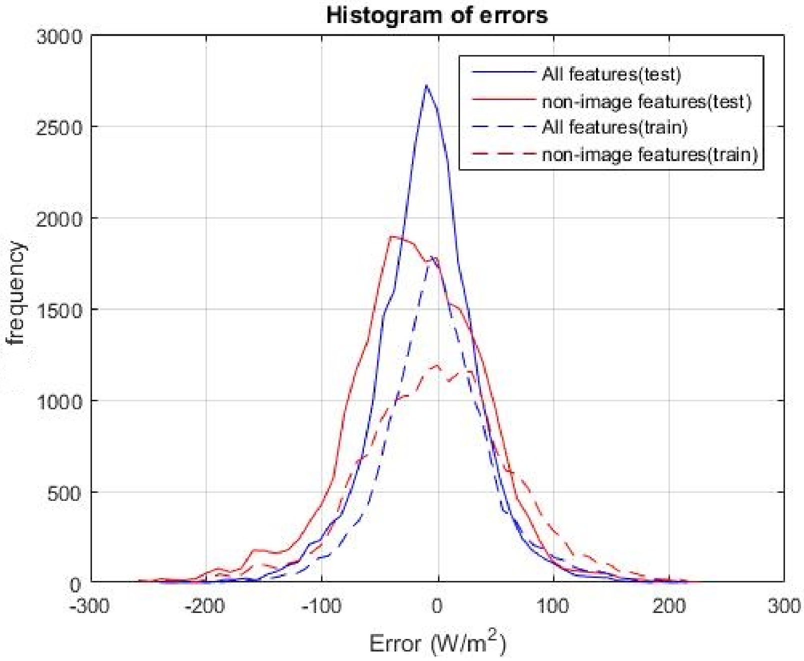


To investigate distribution of errors, we have plotted the histogram of training and test errors for both non-image and full feature set in Figure 4.3. One can see that error when using all the feature elements is clearly lower than when using only non-image features. Also, the training error is usually below the test error, but in the right edge of the plot the training error frequency exceeds the test error. This can indicate an underestimating bias in the training set which is due to relatively higher number of low-DHI samples in the data set.

4.3 K-NN

For K-NN method, the non-image and full feature experiments is repeated as well. After trying different values of neighbors (K), K=10 showed the least

Figure 4.3: Error histogram for linear regression



error and is been used for the rest of experiments. The feature selection on non-image features only selects clear-sky-DHI and exclude sun-zenith due to its negligible importance. The table 4.1 displays the errors obtained using K-NN regression on training ans test data. And the Figure 4.4 shows correlation of results for non-image feature set.

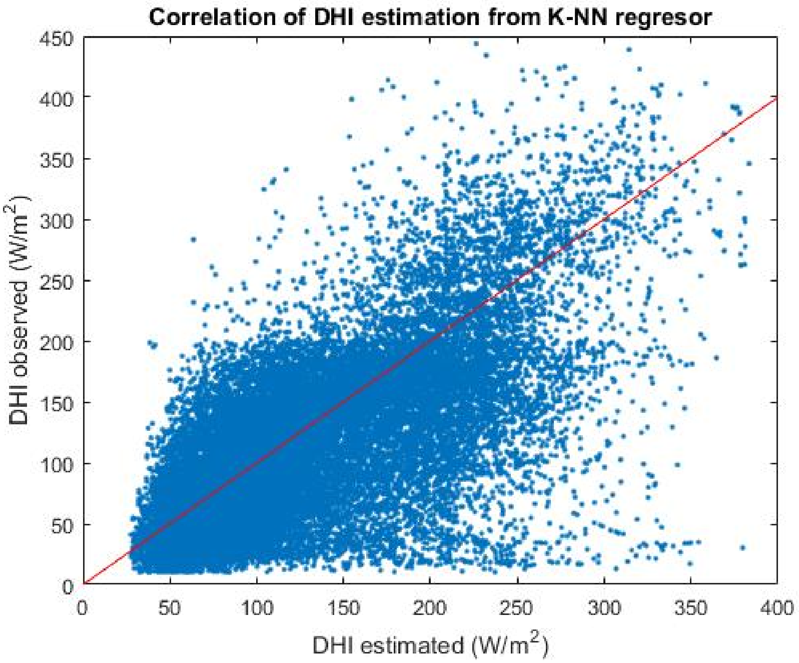
	non-image features	both feature types
Training set	58.2	35.7
Test set	57.2	**34.8**

Table 4.1: Regression errors of K-NN on non-image and full feature vectors

One can see that using both non-image and image-based features reduces RMSE by around 39% from only using non-image based features.

Running the feature selection on base features for K-NN regressor results in leaving out the saturation factor feature. This indicates its negligible correlation to DHI which was not obvious at the time of creating feature list. This low importance can be due to the fact that for many images where sun is completely visible, the saturation factor (which is only defined for cloud pixels around sun) is trivially zero. And these images correspond to high DHI values while having zero saturation. However, there are many cloudy images where sun is not visible but because of think clouds, the

Figure 4.4: K-NN estimation result using only non-image features



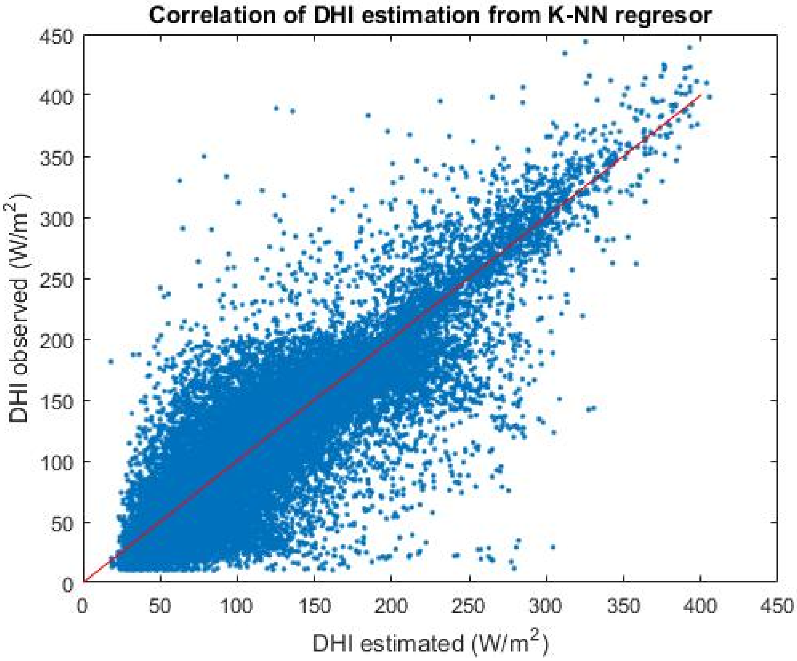
saturation factor is again zero and DHI value is low. Therefore, we can see that the relation of DHI to saturation factor is more complex than linear or simple non-linear methods. Thus, deriving a more sophisticated feature from saturation factor, sun-state and circumsolar cloud coverage could be more relevant to diffuse irradiance. The correlation of K-NN result to DHI while using the selected features is depicted in Figure 4.5 .

The error histogram of result for K-NN method is illustrated in Figure 4.6. This figure confirms that error when using both feature types is almost always below the error of non-image features alone. The training error is also below the test error except in the most right side of the plot where it goes above test error. The slightly higher training error in Table 4.1 can be related to this parts of the error histogram.

4.4 SVR

The performance of support vector regression is highly dependent on C, ε and γ parameters which define trade off between model complexity (i.e. number of support vectors) and regression error and control influence of support vectors. Therefore, it is very important to tune these parameters on our specific dataset through k-fold cross validation. We are using 5-fold validation based on our training sample size. Thus, we randomly divide our training set into 5 equal sized partitions. Then for each parameter combina-

Figure 4.5: K-NN estimation result using all the features



tion we use one of the folds for testing and the rest for training set. The final error of a parameter combination comes from averaging the errors for all the folds (5). After cross validation on possible ranges for C and ε and γ parameters, the best values are selected to be 250, 9 and 8 respectively. Besides RBF, linear and sigmoid kernels have been evaluated for this task, but RBF outperformed other kernel types and was selected for further experiments on SVR. The popular libsvm toolbox is used for training an SVM model and predicting DHI values. We have not applied feature selection directly on SVR due to its long runtime, however, we manually removed some feature elements such as saturation factor and circumsolar cloud coverage, and the error was decreased to 38.0. Therefore, we excluded them from features list for further experiments on SVR.

The result of SVR is summarized in the table 4.2. It shows an improvement of 38% when using both non-image and image-based features. This can also be seen in Figures 4.7 and 4.8 and which depicts correlation of SVR results to DHI values.

	non-image features	both feature types
Training set	62.0	38.4
Test set	62.7	**38.0**

Table 4.2: Regression errors of SVR on non-image and both feature types

Figure 4.6: Error histogram for K-NN estimation result

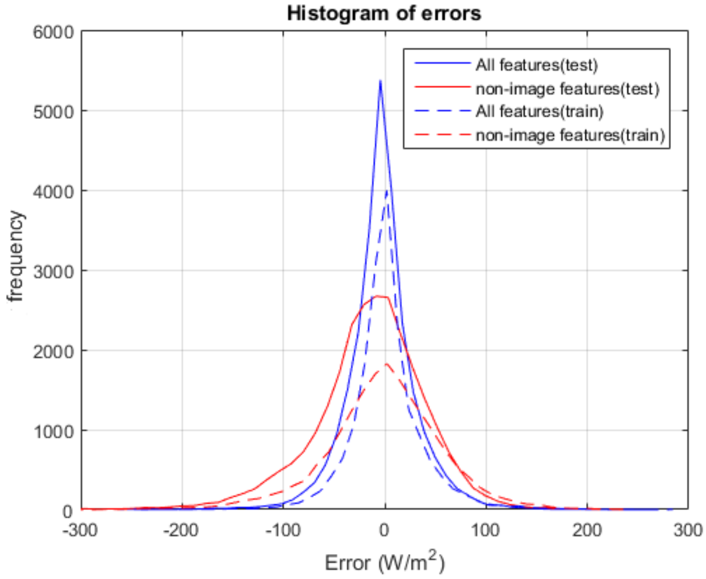
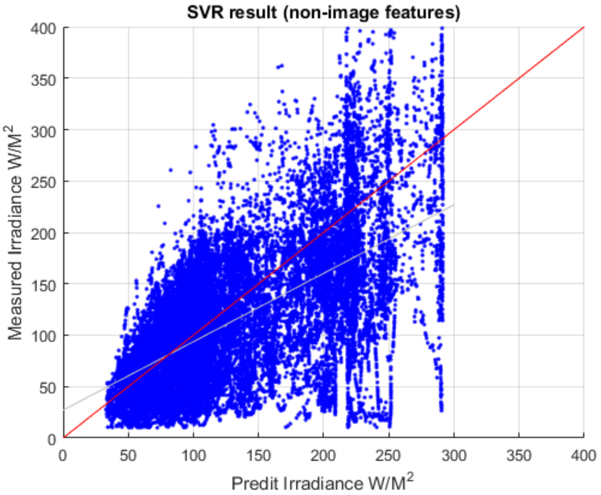


Figure 4.7: SVR result using only non-image features



The error histogram of error is plotted in Figure 4.9. This shows more clearly the improvement of using image-based features. It also proves that the learned model from SVR method is relying on key features that are showing consistent behavior in training and test sets.

Figure 4.8: SVR result using both feature types

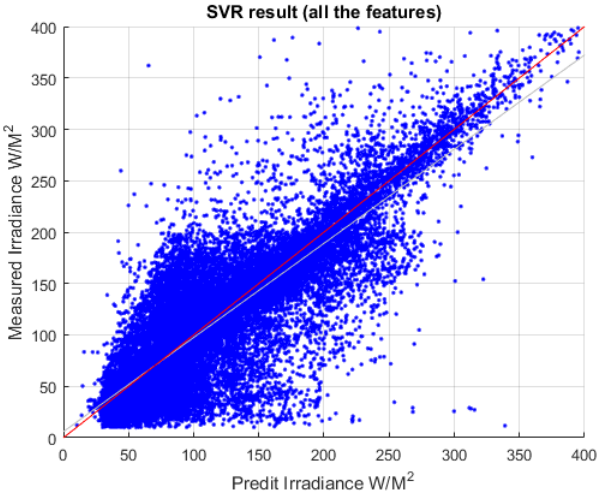
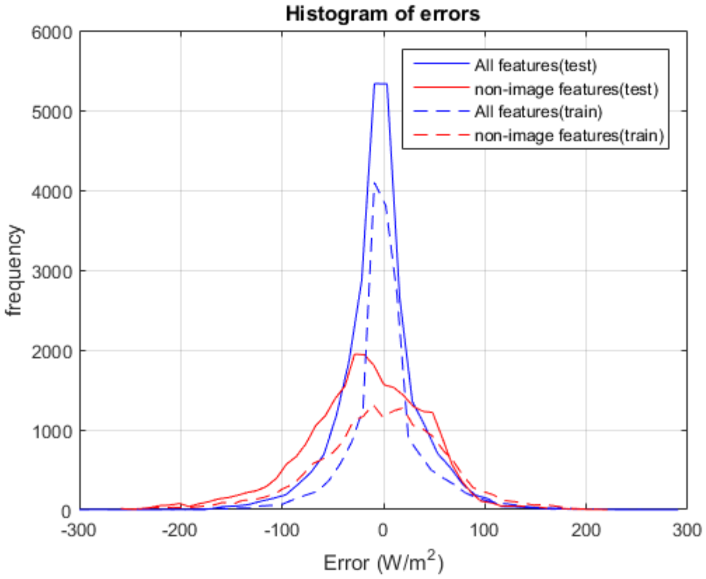


Figure 4.9: Error histogram for SVR result



4.5 Comparison

As we saw before, using image-based features along non-image features improves the regression accuracy considerably. Now, for choosing the best performing method among the three regression algorithms, we compare the RMSE of their result for the case where both feature types were used. As Table 4.3 clarifies, K-NN method outperforms linear regression by 20% and support vector regression by around 9% improvement.

However, bigger training set can reduce these errors. It's specifically more

	Linear Regression	K-NN	SVR
Test error	44.7	**34.8**	38.0

Table 4.3: Comparison of RMSE for all three regression methods using both feature types

beneficial for K-NN since it relies on finding similar cases in training data for every test instance. On the other hand, by using other kernel types or a better feature selection, we might be able to improve the performance of SVR, but it's not guaranteed. Therefore, we choose **K-nearest-neighbor** as the best method for DHI prediction for our application.

Chapter 5

Future Work

Chapter 6

Conclusion

Dummy text.

Bibliography

[1] Shperical model for position of sun. https://pvpmc.sandia.gov/modeling-steps/1-weather-design-inputs/sun-position/. Accessed: 2016-03-10.

[2] Shperical model for position of sun. http://www.soda-pro.com/web-services/radiation/cams-mcclear. Accessed: 2016-03-20.

[3] P. Blanc and L Wald. A Library for Computing the Relative Position of the Sun and the Earth. Technical report, 2011.

[4] J. Borkowski, Chai A.-T., Mo T., and Green A. E. O. Cloud effects on middle ultraviolet global radiation. 25(4):287–301, 1977.

[5] B. D Bourges. Yearly variations of the Linke turbidity factor. page 61–64, 1992.

[6] A. Cazorla. *Development of a Sky Imager for Cloud Classification and Aerosol Characterization*. PhD thesis, Universidad de Granada, Granada, Spain, 2010. PhD thesis.

[7] A. Cazorla, F. J. Olmo, and L. Alados-Arboledas. Development of a sky imager for cloud cover assessment. 25:29–39, 2008.

[8] O. Chapelle and Vapnik V. Model Selection for Support Vector Machines. 12, 1999.

[9] C. W. Chow, B. Urquhart, M. Lave, A. Dominguez, J. Kleissl, J. Shields, and B. Washom. Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. 85:2881–2893, 2011.

[10] M. Fontoynont, D. Dumortier, D. Heinnemann, A. Hammer, J. Olseth, A. Skarveit, P. Ineichen, C. Reise, J. Page, L. Roche, H. G. Beyer, and

L. andWald. Satellight: aWWWserver which provides high quality daylight and solar radiation data for Western and Central Europe. page 434–437, 1998.

[11] A. Hammer, D. Heinemann, E. Lorenz, and B. Lückehe. Shortterm forecasting of solar radiation: a statistical approach using satellite data. 67:139–150, 1999.

[12] P. Ineichen. Long Term Satellite Hourly, Daily and Monthly Global, Beam and Diffuse Irradiance Validation. 2013.

[13] R. H. Inman, H. T. C. Pedro, and C. F. M. Coimbra. Solar forecasting methods for renewable energy integration, Prog. Energ. 39:535–576, 2013.

[14] R. Johnson, W. Hering, and J. Shields. Automated Visibility and Cloud Cover Measurements with a Solid State Imaging System. 1989.

[15] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. 32:922–930, 1976.

[16] J. Kühnert, E. Lorenz, and D. Heinemann. Satellite-based irradiance and power forecasting for the German energy market. page 504, 2013.

[17] Sandia National Laboratories. PVLIB Toobox v1.2 for Matlab. 2012.

[18] M. Lefèvre, A. Oumbe, P. Blanc, B. Espinar, B. Gschwind, Z. Qu, L. Wald, M. Schroedter-Homscheidt, C. Hoyer-Klick, A. Arola, A. Benedetti, J. W. Kaiser, and J.-J. Morcrette. McClear: a new model estimating downwelling solar radiation at ground level in clear-sky conditions. 6:2403–2418, 2013.

[19] E. Lorenz and D. Heinemann. Prediction of solar irradiance and photovoltaic power. 1:239–292, 2012.

[20] E. Lorenz, D. Heinemann, and A Hammer. Short-term forecasting of solar radiation based on satellite data. page 841–848, 2004.

[21] F. J. Olmo, A. Cazorla, L. Alados-Arboledas, M. A. Lopez-Alvarez, J. Hernandez-Andres, and J. Romero. Retrieval of the optical depth using an all-sky CCD camera. 47:182–189, 2008.

[22] R. Perez, E. Lorenz, S. Pelland, M. Beauharnois, G. Van Knowe, K. Hemler Jr., D. Heinemann, J. Remund, S. C. Müller, W. Traunmüller, G. Steinmauer, D. Pozo, J. A. Ruiz-Arias, V. Lara-Fanego, L. Ramirez-Santigosa, M. Gaston-Romero, and L. M. Pomares. Comparison of numerical

weather prediction solar irradiance forecasts in the US, Canada and Europe. 94:305–326, 2013.

[23] G. Pfister, R. L. McKenzie, J. B. Liley, A. Thomas, B. W. Forgan, and C. N. Long. Cloud coverage based on all-sky imaging and its impact on surface solar irradiance. 42:1421–1434, 2003.

[24] I. Reda and A. Andreas. Solar position algorithm for solar radiation application. Technical report, 2003.

[25] G. Reikard. Predicting solar radiation at high resolutions: a comparison of time series forecasts. 83:342–349, 2009.

[26] J. Sabburg and J. Wong. Evaluation of a Ground-Based Sky Camera System for Use in Surface Irradiance Measurement. 16:752–759, 1998.

[27] S. Sayeef, S. Heslop, D. Cornforth, T. Moore, S. Percy, J. Ward, A. Berry, , and D. Rowe. Solar Intermittency: Australia's Clean Energy Challenge: Characterising the Effect of High Penetration Solar Intermittency on Australian Electricity Networks. 2012.

[28] D. Scaramuzza. OCamCalib: Omnidirectional Camera Calibration Toolbox for Matlab. 2014.

[29] T. Schmidt, J. Kalisch, and E Lorenz. Retrieving direct and diffuse radiation with the use of sky imager pictures. 17:12–17, 2015.

[30] T. Schmidt, J. Kalisch, E. Lorenz, and Heinemann D. Evaluating the spatio-temporal performance of sky-imager-based solar irradiance analysis and forecasts. 16:3399–3412, 2016.

[31] J. E. Shields, M. E. Karr, T. P. Tooman, D. H. Sowle, and S. T. Moore. The whole sky imager – a year of progress. page 23–27, 1998.

[32] R. Tapakis, A.G. Charalambides, M.D. Moldovan, and B.G. Burduhos. Cloudy sky irradiance model using sky images. 2015.

[33] R. Tapakis, A.G. Charalambides, M.D. Moldovan, and B.G. Burduhos. Effect of clouds on solar irradiance. Technical report, Universitatea Transilvania din Braşov, 2015.

[34] S. R. West, D. Rowe, S. Sayeef, and A. Berry. Short-term irradiance forecasting using skycams: motivation and development. 110:188–207, 2014.

[35] B. Wolff, E. Lorenz, and O. Kramer. Statistical learning for short-term photovoltaic power predictions. 2013.

[36] D. Yang, Z. Dong, T. Reindl, P. Jirutitijaroen, and W. M. Walsh. Solar irradiance forecasting using spatiotemporal empirical kriging and vector autoregressive models with parameter shrinkage. 103:550–562, 2014.