



---

# Final Year Project

---

## Development and Analysis of a Multi-Modal Environmental Data Acquisition System for Solar Forecasting

Ciarán Scanlon

---

Student ID: 21428034

---

A thesis submitted in part fulfilment of the degree of

**BSc. (Hons.) in Computer Science**

**Supervisor:** Doctor Soumyabrata Dev



UCD School of Computer Science

University College Dublin

April 27, 2025

---

# Table of Contents

---

<b>1</b>	<b>Introduction</b>	5
<b>2</b>	<b>Related Works and Ideas</b>	6
2.1	Introduction	6
2.2	Sky Imaging	6
2.3	Ground Based Sensors	9
2.4	Solar Forecasting	11
2.5	Related Ground Based Sky Imagers	14
<b>3</b>	<b>Hardware Overview</b>	16
3.1	Introduction	16
3.2	System Data Flow and Architecture Overview	16
3.3	Environmental Sensors	17
3.4	Imaging Hardware	19
3.5	Computational Hardware and Data Storage	20
<b>4</b>	<b>Enclosure Design</b>	22
4.1	Design Requirements	22
4.2	Junction Box Enclosure	23
4.3	Acrylic Domes for Camera Protection	24
4.4	Cable Gland Integration	24
4.5	Wooden Support and Mounting Frame	25
4.6	CAD Model and Dimensions	26
4.7	Discussion	27
<b>5</b>	<b>System Design, Deployment, and Monitoring</b>	28
5.1	Internal Layout	28
5.2	Component Interconnections	28
5.3	Power and Cable Management	30
5.4	Deployment	31

---

5.5	Platform and Orientation . . . . .	31
5.6	Remote Operation and Monitoring . . . . .	31
<b>6</b>	<b>Software and Data Handling . . . . .</b>	<b>32</b>
6.1	Overview . . . . .	32
6.2	Modular Script Design . . . . .	32
6.3	Image Capture . . . . .	33
6.4	Environmental Data Capture . . . . .	34
6.5	Data Logging . . . . .	36
6.6	Scheduling and Execution . . . . .	37
6.7	Results – Processing and Modelling . . . . .	37
<b>7</b>	<b>Data Insights and Predictive Modeling for Solar Irradiance Forecasting . . . . .</b>	<b>42</b>
7.1	Deployment Summary and Data Context . . . . .	42
7.2	Feature Correlation Analysis . . . . .	44
7.3	Image Histogram and Cloud Analysis . . . . .	47
7.4	LSTM Forecasting Model . . . . .	50
<b>8</b>	<b>Discussion, Future Work and Conclusion . . . . .</b>	<b>53</b>
8.1	Discussion . . . . .	53
8.2	Future Works: . . . . .	53
8.3	Conclusion . . . . .	54
<b>9</b>	<b>Acknowledgements . . . . .</b>	<b>56</b>
<b>A</b>	<b>Appendix . . . . .</b>	<b>60</b>
A.1	Code Repository Access . . . . .	60
A.2	Appendix B: Key Code Listings . . . . .	60
A.3	SolidWorks CAD Drawings . . . . .	65
A.4	Sample Data . . . . .	66
A.5	Tools Used . . . . .	70

---

# Abstract

---

This project presents the design, implementation and analysis of a multi-modal data acquisition system developed for the estimation and study of solar irradiance. The system integrates a range of environmental sensors including temperature, humidity and light sensors alongside dual imaging modules capable of capturing RGB (Red, green, blue) and NIR (Near-infrared) images at one minute intervals. The device is built on Raspberry Pi hardware and autonomously records synchronised environmental and visual data, creating a comprehensive dataset for solar analysis.

Subsequent data processing involved the extraction of numerical features from both the environmental sensors outputs and the image data. Comparative analysis was conducted to highlight the differences between RGB and NIR imaging, specifically in the context of solar forecasting and statistical analysis was performed on the numerical features to explore the relationships between each measured parameter and recorded solar irradiance values.

To further explore the predictive potential of the dataset, a Long short-Term Memory (LSTM) neural network was developed. The model was trained on historical sequences of multi-modal data to forecast future solar irradiance values.

Overall, this work demonstrates the feasibility and value of integrating environmental sensing with spectral imaging and deep learning techniques for solar irradiance estimation it also highlights the usefulness of infrared imaging for this, laying the groundwork for future applications for low-cost, real-time solar monitoring systems

---

# Chapter 1: Introduction

---

Solar energy stands at the forefront of national energy strategies, offering a clean, sustainable, and abundant alternative to fossil fuels. Its widespread deployment plays a pivotal role in reducing CO<sub>2</sub> emissions and contributing to a less carbon-intensive future, which is essential amid the ongoing global energy crisis [1]. However, the intermittent and unpredictable nature of solar power, due to daily and seasonal variations in sunlight and atmospheric conditions poses a major challenge to its reliable integration into modern power grids [2]. To overcome this, accurate and accessible solar forecasting solutions are essential. These tools help improve the reliability of solar energy systems and facilitate their seamless incorporation into the existing energy infrastructure [3].

Solar forecasting refers to the process of predicting the amount of solar energy that will be available at a specific location and time. Forecasting techniques are generally classified into three main categories: time-series statistical methods, physical models, and ensemble approaches [4]. Physical models, such as Numerical Weather Prediction (NWP), satellite imaging, and sky imagery, use atmospheric simulations and observational data to forecast solar irradiance, particularly effective for weather-based predictions [5]. Time-series statistical methods, including machine learning techniques like Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), utilize historical solar and meteorological data to learn patterns and relationships for predictive modelling. Ensemble methods combine both statistical and physical models to enhance forecast accuracy by leveraging the strengths of different approaches [4]. For this project, a machine learning-based statistical approach is employed due to its ability to model the non-linear and complex patterns in solar data.

Effective solar forecasting relies on the availability of accurate, high-resolution, and relevant data [6]. This data can be obtained through a variety of sources, including satellite imagery, sky-imagers, and ground-based environmental sensors. Sky imagers provide real-time, wide angle images of the sky which helps analyse cloud cover, cloud composition and cloud motion. Sky-imagers provide very useful data for short term predictions because of this [7]. Satellite imagery provides broader spatial coverage and is useful for identifying large-scale atmospheric patterns [8]. Ground sensors measure local conditions such as solar irradiance, temperature, humidity, and wind speed, all of which contribute valuable contextual data for forecasting models [9].

The aim of this project is to design and develop a data acquisition device capable of collecting local environmental sensor data and capturing both RGB and NIR sky images at regular intervals which will facilitate in the creation of accurate localised solar forecasting. The project ultimately demonstrates a low-cost, modular, and extensible approach to solar forecasting using real-time ground-level sensing and image analysis.

---

# Chapter 2: Related Works and Ideas

---

## 2.1 Introduction

This project aims to design and develop a novel, low-cost data acquisition system that captures both environmental sensor data and sky images in order to explore their usefulness in solar irradiance forecasting. While no identical system appears to have been implemented before, various individual components such as sky imaging, environmental sensing, and solar forecasting using machine learning are well documented in existing literature.

This section outlines previous work related to each of these components. First, the field of sky imaging is reviewed, focusing on its role in cloud detection, the integration of RGB (Red, green, blue) and NIR (Near-infrared) imaging, imaging techniques and technologies, and the challenges associated with sky-based image capture are discussed. Second, the importance and implementation of ground-based environmental sensors are explored, the kinds of sensors used in these devices and how they are powered and connected to. Third, solar forecasting methodologies are reviewed, with a particular emphasis on the use of machine learning models like LSTM for short-term predictions.

## 2.2 Sky Imaging

### 2.2.1 Cloud Detection

Clouds exert a substantial influence on solar energy generation by intermittently blocking sunlight and causing fluctuations in the amount of irradiance reaching solar panels [7]. Sky imaging plays a crucial role in detecting and analyzing these clouds. By capturing sky images, valuable insights into cloud coverage, motion patterns, and cloud-base height can be obtained, factors that are essential for improving the accuracy of solar forecasting models [7]. The movement of clouds is particularly significant, as their shifting positions directly affect the consistency of solar irradiance. As clouds drift across the sky, they can reduce the available solar energy in unpredictable ways. Sky imaging facilitates the monitoring of such dynamic cloud behaviour, thereby enhancing the forecasting of solar irradiance variations [10].



Figure 2.1: Image showing Cloud Detection Software

Figure 2.1 shows the output of a cloud detection system based on sky images and object detection algorithms, adapted from Wang et al. [11].

## 2.2.2 RGB And NIR

RGB and NIR imaging systems capture different segments of the electromagnetic spectrum, each providing unique information about the atmosphere and sky conditions.

RGB cameras capture the visible spectrum (approximately 400–700 nm), producing red, green, and blue values that mimic human vision as shown in Figure 2.2. These cameras are commonly used in sky imaging to identify cloud cover, cloud type, and sky clarity. The RGB values of pixels in a sky image correlate strongly with solar irradiance: blue sky pixels (e.g., RGB (0, 0, 255)) typically indicate high irradiance, while white clouds ((255, 255, 255)) and gray clouds ((150, 150, 150)) are associated with reduced irradiance [12].

In contrast, NIR imaging captures wavelengths just beyond the visible spectrum, typically from around 700 to 1000 nanometers Figure 2.3. NIR imaging provides several advantages over standard RGB imaging in the context of solar irradiance forecasting. NIR sky images significantly reduce saturation in the circumsolar region, meaning cloud data around the sun that could be washed out due to the brightness of the sun in RGB images can be maintained. Furthermore, NIR imaging enables improved detection of cloud layers and segmentation under complex sky conditions, as it is less affected by atmospheric scattering and haze as NIR light is less prone to scattering. NIR can also detect heat energy coming from clouds which can help indicate how thick they are and their altitude. These characteristics make NIR imaging useful especially for extracting physical cloud features. [13]



Figure 2.2: RGB Sky Image [12]

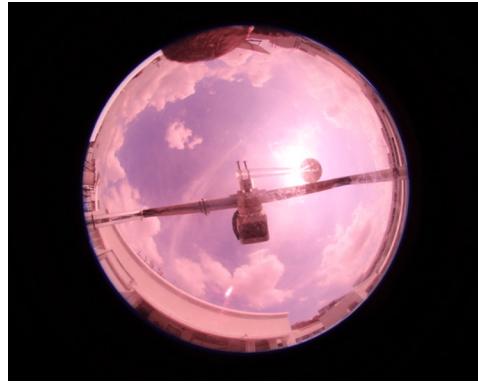


Figure 2.3: NIR Sky Image [7]

### 2.2.3 Lenses and Imaging Techniques

Ground-based sky imaging systems often utilize wide-angle lenses to capture expansive views of the sky. The Total Sky Imager (TSI), for instance, employs a hemispherical chrome-plated mirror that reflects the sky into an upward-facing CCD camera positioned above the mirror. Additionally, a sun-tracking shadow band is incorporated to shield the optical sensor from direct solar glare, ensuring consistent image quality [14].

Alternatively, All-Sky Imagers (ASIs) typically use fisheye lenses mounted on cameras housed within weatherproof enclosures. These fisheye lenses effectively capture the entire sky dome, although they introduce optical distortions that require correction during the image processing stage [10], other imagers incorporate wide-angle lenses that have less field of view but also less distortion.

Beyond lens selection, techniques such as High Dynamic Range (HDR) imaging are widely applied. HDR combines multiple exposures taken at different light intensities to produce images that capture a wider dynamic range. This method is particularly beneficial in sky imaging, where scenes often contain extreme contrasts between bright sunlight and darker cloud regions, ensuring that details across all brightness levels are preserved [15].

### 2.2.4 Challenges with Sky Imaging

The first challenge is the problem of cloud overlap, which arises from the inherent limitations of using 2D imaging to capture a 3D atmosphere. When multiple cloud layers are present, clouds at different altitudes can appear to overlap or merge in a single image, making it difficult to differentiate them or analyze their individual dynamics [15]. This perspective distortion limits the depth-related information that can be extracted from a single camera viewpoint.

The second major challenge is lighting variability. The interplay between the sun and clouds produces a wide range of light intensities, often exceeding the dynamic range of standard cameras. Without compensation, this can lead to overexposed regions near the sun and underexposed areas in shadowed regions, both of which result in a loss of detail. One approach to mitigate this issue is High Dynamic Range (HDR) imaging, which captures multiple exposures in rapid succession to better represent both bright and dark areas [15]. However, HDR processing requires additional hardware or synchronization and may not be suitable in fast-moving cloud conditions. Another potential solution would be to automatically change the image capturing settings based on the lighting conditions.

---

A third key challenge lies in the computational demands of processing sky images. Extracting meaningful features from high-resolution image sequences, especially when using deep learning models require significant computational power and storage resources [10]. Real-time applications in particular must balance speed with accuracy, which can be difficult to achieve on resource-constrained systems.

In addition to these core issues, practical challenges also arise in field deployments. One common problem is the accumulation of mist, condensation, or dust on the camera lens or housing. This can obscure parts of the image, reduce contrast, cause image glare, and introduce noise into the data. Regular maintenance and housing design become critical in such cases as well as other mitigating techniques. Furthermore, direct exposure to sunlight, especially when the sun is within the camera's field of view, can also cause lens flaring, which can degrade image quality and affects cloud segmentation performance.

Addressing these challenges requires a combination of hardware improvements, intelligent image processing techniques, and robust machine learning models capable of adapting to the diverse and dynamic nature of the sky.

## 2.3 Ground Based Sensors

### 2.3.1 Importance of Measuring Atmospheric Conditions

Atmospheric conditions play a critical role in both the calculation and prediction of solar energy potential. Key environmental factors such as cloud cover [16] and aerosol concentration [17] directly affect the amount of solar radiation that reaches the Earth's surface, and thus, the output of solar energy systems.

Ground-based sensors are essential for capturing localized atmospheric data. Unlike satellite data, which provides broader coverage but lower spatial resolution, ground sensors offer high-resolution, site-specific information. This includes measurements of irradiance, temperature, humidity, air pressure, and particulate matter. Such data is vital for improving the accuracy of solar forecasting models, particularly at short time scales.

By integrating real-time atmospheric data from ground sensors, forecasting systems can better adapt to rapid changes in local conditions. This is essential for optimizing local forecasting, minimising errors and improving accuracy. [18]

### 2.3.2 Types of Sensors Used

Accurate solar energy forecasting relies on a diverse array of atmospheric sensors that monitor environmental conditions. Key instruments include:

#### Whole Sky Imagers (WSI) and Ground-Based Sky Imagers (GSI)

WSIs and GSIs capture hemispherical images of the sky at regular intervals using fisheye lenses. These images provide high temporal and spatial resolution data on cloud cover and movement, which are critical for short-term solar irradiance forecasting. Unlike pyranometers, sky imagers offer detailed information about cloud dynamics, enabling more precise predictions of solar energy

---

availability [7].

## Pyranometers

Pyranometers are instruments that measure solar irradiance, specifically the total shortwave radiation from the hemisphere above the sensor. They are essential for assessing the amount of solar energy reaching a specific location. By converting solar radiation into an electrical signal, pyranometers provide real-time data on solar power potential [19].

## Ceilometers

Ceilometers are ground-based, active remote-sensing devices that measure the height of cloud bases using laser pulses. They provide accurate data on multiple cloud layers, even under limited visibility conditions, which is vital for understanding cloud impact on solar irradiance [ceilometer].

## Wind Speed Sensors

Wind speed sensors, or anemometers, measure the velocity of wind, which influences cloud movement and, consequently, solar irradiance levels. Accurate wind measurements are crucial for predicting changes in cloud movements and solar energy availability [20].

## Humidity Sensors

Humidity sensors monitor the amount of moisture in the air, which affects cloud formation, atmospheric transmittance, and solar radiation levels. Higher humidity can lead to greater scattering of light, reducing the amount of solar energy that reaches the surface. Studies have shown that relative humidity data can be used to estimate solar irradiance with high accuracy, making it a valuable input for solar forecasting models [21].

## Temperature Sensors

Temperature sensors measure the ambient air temperature, which influences cloud formation and atmospheric transmittance. Changes in temperature affect the amount of solar radiation that reaches the Earth's surface and are commonly used in models that estimate solar irradiance. When combined with humidity data, temperature has been shown to enable accurate predictions of hourly solar radiation [21].

## Additional Sensors

Other valuable sensors include:

- **Photometers:** Measure aerosol concentration.
- **Barometers:** Monitor atmospheric pressure.
- **Rain Gauges:** Quantify precipitation.

---

### 2.3.3 Power and Connectivity

Reliable power and data connectivity are crucial components in the deployment of ground-based sky imagers and environmental sensor systems, especially in long-term outdoor installations.

#### Powering Remote Sensing Systems

Powering such systems can be achieved through several means. In temporary or remote deployments, portable battery packs or power banks are often used for their ease of setup and mobility. For long-term or autonomous installations, solar panels paired with charge controllers and energy storage solutions are a widely adopted alternative, offering sustainable, off-grid power. This approach has been successfully demonstrated by Avlani et al. (2022), who developed a solar powered environmental sensor that can run continuously using a small amorphous silicon solar panel and energy saving techniques [22]. Where available, fixed infrastructure may also allow direct power supply, which simplifies deployment but limits flexibility in site selection like in WHARSIS [17].

#### Data Storage and Transmission Strategies

In terms of data transmission and storage, real-time cloud uploading via Ethernet or Wi-Fi is often ideal for remote environmental monitoring systems, allowing for live data access and centralized analysis. However, in many outdoor or remote settings, network connectivity may be intermittent or entirely unavailable. In such cases, local data storage becomes essential. Common approaches include writing data directly to onboard storage such as SD cards or to external USB drives [23], which offer high-speed data access and substantial capacity for high-frequency data logging tasks.

This offline-first architecture is well-suited for edge devices, enabling systems to function independently of network infrastructure. It is commonly adopted in sky imaging and environmental monitoring systems, such as the WAHRSIS platform, which relies on local data logging with periodic batch uploads or manual retrieval when internet access is not available [17]. These designs offer improved system reliability by eliminating dependence on real-time connectivity, reducing the risk of data loss from transmission failures.

---

## 2.4 Solar Forecasting

Effective solar forecasting relies on the integration of diverse data sources, including sky imagery and ground-based environmental sensors. Sky images, both in RGB and NIR spectra, have proven to be valuable in enhancing forecasting models by providing real-time visual data on cloud cover and atmospheric conditions. NIR imaging, in particular, improves cloud detection under variable lighting, making it especially useful for identifying features that influence irradiance but may be missed by other sensing methods [7].

Forecasting models that incorporate sky imagery can better anticipate rapid fluctuations in solar irradiance caused by dynamic cloud movements [7]. In the context of this project, such multi-modal data forms the basis for improving short-term solar forecasts. This section reviews key forecasting approaches and how imagery and environmental data can be integrated to enhance prediction accuracy.

---

## 2.4.1 Forecasting Methodologies

Solar forecasting techniques are generally categorized into three main approaches: statistical models, physical models, and artificial intelligence (AI)-based models [4]. The subsequent sections detail these forecasting methodologies.

## 2.4.2 Statistical Models

Statistical models form the foundation of many short-term solar forecasting techniques. These methods leverage historical data patterns to predict future values.

### Time Series Models

Time series analysis is one of the most widely used statistical approaches in solar forecasting. It involves analysing historical data collected at regular time intervals to uncover temporal trends, seasonality, and autocorrelations. Popular models include:

- **AutoRegressive (AR)**: Predicts current values based on a weighted sum of previous observations.
- **Moving Average (MA)**: Models the prediction as a function of past forecast errors.
- **ARMA/ARIMA**: Combines AR and MA components, with ARIMA including differencing for non-stationary data. [24]

Time series models are easy to implement and computationally efficient, making them suitable for baseline models and resource-limited environments. However, they have limitations in capturing non-linearities and external influencing factors like cloud movement [4].

## 2.4.3 Physical Models

Physical models simulate solar irradiance using radiative transfer models and NWP outputs. These models are robust for long-term forecasting and allow interpretability grounded in atmospheric science. However, they are highly sensitive to model resolution, data quality, and initialization conditions. Intra-hour predictions are challenging due to the coarse granularity of most NWPs [12]. As shown by Ramadhan et al., combining physical and machine learning models can yield improved results by leveraging both interpretability and flexibility [25].

## 2.4.4 Artificial Intelligence and Deep Learning

Artificial intelligence (AI) and deep learning techniques have revolutionized solar forecasting by enabling models to learn complex, nonlinear relationships from vast and diverse datasets. These models are good at handling the stochastic nature of solar irradiance, influenced by numerous interacting environmental factors.

## Sequential Models and Time Series Forecasting

Sequential models process input data as ordered sequences, making them especially effective for time-series data where past information is crucial for predicting future values. Unlike traditional statistical models, sequential models can learn hidden patterns in temporal sequences by memorizing data from prior time points making them useful in solar forecasting [26].

### Long Short-Term Memory (LSTM) Networks

A prominent sequential model in solar forecasting is the Long Short-Term Memory (LSTM) neural network, a variant of Recurrent Neural Networks (RNNs). LSTM networks are designed to capture long-term dependencies by mitigating the vanishing gradient problem common in standard RNNs. Each LSTM cell contains gates (input, output, and forget) that control the flow of information, allowing the model to selectively remember or discard information from past time steps [27].

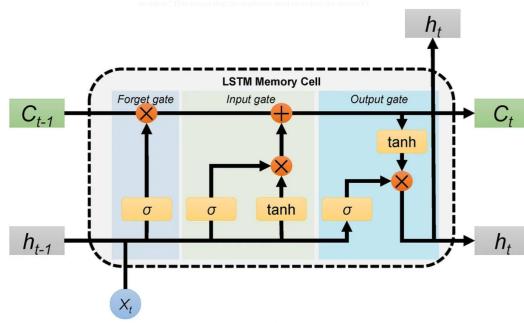


Figure 2.4: Long Short Term Memory Cell [28]

Long Short-Term Memory (LSTM) networks are highly effective for both intra-hour and day-ahead solar forecasting, owing to their ability to learn complex temporal dependencies in irradiance, temperature, humidity, and other meteorological variables. Their architecture enables them to mitigate the vanishing gradient problem common in standard RNNs, allowing them to retain relevant information across long sequences. Empirical studies have demonstrated that LSTM models consistently outperform traditional approaches such as ARIMA, feedforward neural networks (FFNNs), and standard RNNs in terms of forecasting accuracy and robustness [29]. LSTMs are particularly well-suited for challenging weather scenarios, such as cloudy and mixed sky conditions, where conventional models often fail [27]. Furthermore, LSTM architectures naturally support multivariate inputs, making them capable of incorporating other meteorological parameters such as wind speed, solar angle, and cloud cover—which can further enhance their predictive performance [30].

### Hybrid Architectures

Advanced architectures often combine LSTM with other deep learning components to enhance forecasting performance:

- **CNN-LSTM Hybrids:** These models utilize convolutional layers to extract spatial features, such as those found in meteorological grids or sky images and LSTM layers to capture their temporal evolution. This hybrid approach effectively models both spatial and temporal dependencies. For example, Zang et al. proposed a CNN-LSTM model that extracts spatial features from weather data and temporal features from historical Global Horizontal Irradiance (GHI) sequences, resulting in improved one-hour-ahead forecasting accuracy across multiple climate zones [31].

---

- **Encoder-Decoder LSTMs:** These are sequence-to-sequence (seq2seq) architectures where an encoder LSTM compresses the input time series into a fixed-length context vector, and a decoder LSTM uses this vector to generate multi-step forecasts. This setup allows for flexible prediction over varying time horizons. Ghimire et al. demonstrated that stacked encoder-decoder LSTM autoencoders can generate accurate daily GSR forecasts and quantify prediction uncertainty effectively, especially when combined with feature selection techniques [32]. Some encoder-decoder models also incorporate attention mechanisms, which dynamically weight time steps to further improve accuracy and interpretability.

**Encoder-Decoder LSTMs:** These models use one LSTM (encoder) to process the input time series and compress it into a summary like context vector. A second LSTM (decoder) then takes this summary to produce multi-step forecasts. This design makes it easier to predict over different time periods. Ghimire et al. showed that stacked encoder-decoder LSTM models can make accurate daily forecasts of Global Solar Radiation (GSR) and help estimate uncertainty in the predictions, especially when feature selection is used to choose important inputs [32].

## 2.4.5 Importance and Applications

As solar photovoltaic (PV) systems become more embedded in modern energy infrastructure, accurate forecasting plays a crucial role in ensuring grid reliability. Fluctuations in solar generation can lead to frequency instability and voltage deviations, especially in smart grids with high PV penetration [33]. Accurate short-term and day-ahead forecasts allow grid operators to anticipate these variations, enabling more efficient load balancing and reducing reliance on backup generation resources.

In addition, reliable forecasts improve the scheduling of battery storage systems and enhance the effectiveness of demand response strategies. By anticipating changes in solar output, grid managers can mitigate the risk of blackouts and improve the operational efficiency of renewable-integrated systems [34]. This capability is particularly critical during rapid ramp events, where even minor errors in forecasting can compromise system stability.

## 2.5 Related Ground Based Sky Imagers

### 2.5.1 Lamsky

The Low Cost and Miniature ground-based Sky Camera (LAMSkyScam) is a just that, a low-cost and miniature ground based sky-camera designed to capture high resolution images of the sky. The Lamsky's key design goals were its affordability and compactness, this was achieved by using 3D printed component and the use of store bought electronics. It used a Raspberry Pi HQ camera with a wide-angle lens. It emphasised easy installation which was achieved by making it pole mountable. It also incorporated cooling mechanisms and had local storage and backup capabilities to make it suitable for remote deployment. Overall it is a compact, cheap and cleverly designed system that is capable of taking high resolution images in both day and night and storing them on the device. [23]

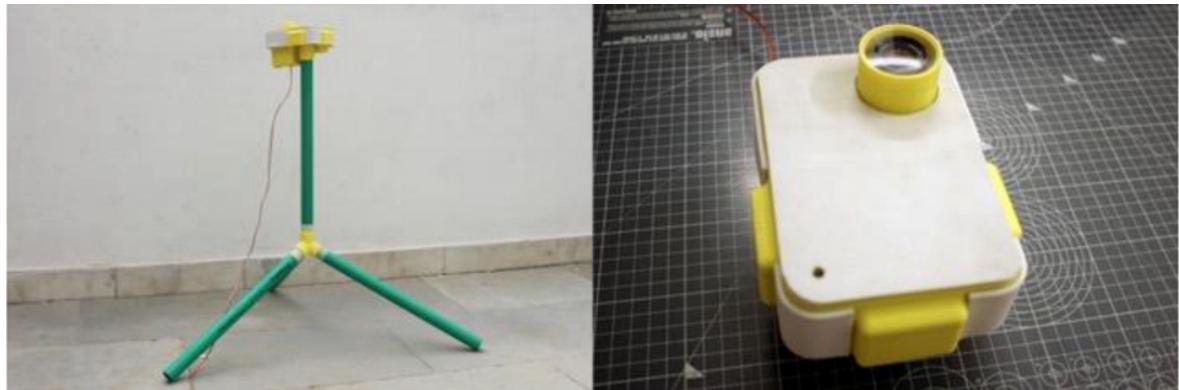


Figure 2.5: LAMSkyCam

### 2.5.2 WHARHIS

The Wide Angle High-Resolution Sky Imaging System (WAHRSIS) is a whole-sky imager developed with the aim of providing high-resolution and cost effective cloud imaging technologies. It uses a Canon EOS Rebel T3i camera paired with a fisheye lens which enable images that cover the entire hemisphere to be captured. It uses a piece of glass which allows near infrared imaging capabilities. This allows clearer images to be captured under hazy conditions. It also features a sun-blocking mechanism controlled by an Arduino, to reduce the glare and improve the images quality. This is also a compact device with some innovations to improve the quality of the images taken. [17]

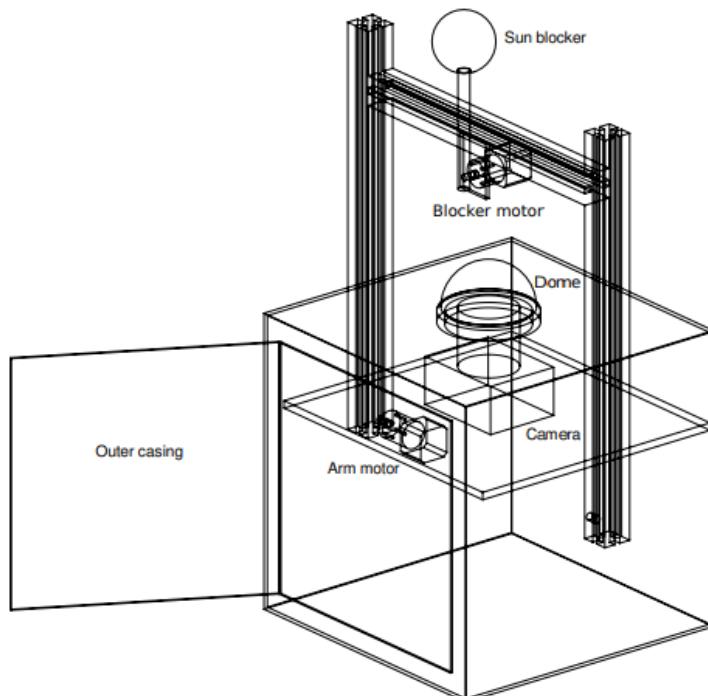


Figure 2.6: WHARSIS Design

# Chapter 3: Hardware Overview

## 3.1 Introduction

This chapter provides an overview of the sensing and computing hardware that form the core of the environmental data acquisition device. The system integrates multiple sensors and cameras with Raspberry Pi units for synchronised data logging.

## 3.2 System Data Flow and Architecture Overview

Category	Device	Purpose
Environmental Sensor	BH1750 (Lux Sensor)	Measure ambient light (lux)
Environmental Sensor	SHT40 (Temp & Humidity Sensor)	Measure temperature and humidity
Imaging	Waveshare IMX219-160 IR-Cut (NIR)	Capture near-infrared images
Imaging	Raspberry Pi HQ Camera (IMX477)	Capture visible-light (RGB) images
Data Processor	Raspberry Pi Model 4B (A)	Collect and timestamp data from NIR and lux sensor
Data Processor	Raspberry Pi Model 4B (B)	Collect and timestamp data from RGB and temp/humidity sensor
Storage	External USB Drive (2 TB)	Store NIR images and lux readings
Storage	External USB Drive (128 GB)	Store RGB images and environmental data

Table 3.1: Hardware Components: Category, Device, and Purpose

The block diagram in Figure 3.1 illustrates the architecture of the solar data acquisition system. The system comprises the following components split into environmental and imaging sensors:

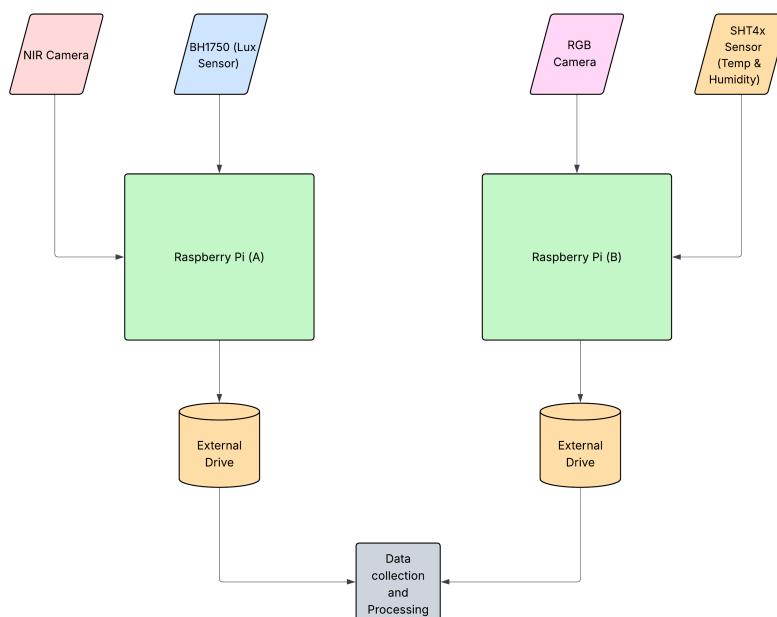


Figure 3.1: Data Flow Diagram for Data Acquisition Device

---

## Data Flow Description

### 1. Sensor and Camera Input:

- The BH1750 sensor transmits ambient light intensity (lux) to Raspberry Pi A.
- The SHT40 sensor provides temperature and humidity readings to Raspberry Pi B.
- The NIR and RGB cameras capture images on Raspberry Pi A and B, respectively.

### 2. Processing and Timestamping:

- Each Raspberry Pi runs a script to capture data once per minute.
- All readings and images are timestamped for synchronization.
- Numeric values are saved to a .csv file as well as the path of the capture images which are captured as .jpgs.

### 3. Storage:

- Data is stored on external drives to enhance reliability and preserve SD card lifespan.

### 4. Data Aggregation and Analysis:

- Data from both drives is periodically transferred for central processing.
- Time-aligned environmental and image data is analysed to extract features and support solar irradiance prediction.

### 3.2.1 Dual Raspberry Pi Setup

Figure 3.1 illustrates the dual Raspberry Pi setup employed in this project. Two separate Raspberry Pis were used, one handling the NIR camera and light sensor, and the other managing the RGB camera and temperature/humidity sensor. This approach was necessitated by the unavailability of a suitable adapter to connect and operate both cameras simultaneously on a single Raspberry Pi. In future iterations, the system may be upgraded to consolidate all sensors and cameras onto a single Raspberry Pi, should appropriate hardware become available, thereby simplifying the architecture and reducing potential points of failure.

## 3.3 Environmental Sensors

### 3.3.1 SHT40 (Temperature & Humidity Sensor)

The SHT40 is a fourth-generation digital temperature and humidity sensor designed for high-accuracy, low-power environmental monitoring. It offers a relative humidity accuracy of up to  $\pm 1.5\%$ RH (relative humidity) and a temperature accuracy of up to  $\pm 0.1^\circ\text{C}$ . The sensor can operate in a wide temperature range from  $-40^\circ\text{C}$  to  $125^\circ\text{C}$  and 0 %RH to 100 %RH, though it performs best within the recommended range of  $5^\circ\text{C}$  to  $60^\circ\text{C}$  and 20 %RH to 80 %RH. Communication is handled via the I<sup>2</sup>C interface and the sensor includes CRC checksums for data integrity. The SHT40 operates on a supply voltage of 1.08 V to 3.6 V. It features an internal heater with three power levels, enabling operation in high-humidity or condensing environments [35]. The particular sensor used also has a water proof housing making it suitable for deployment where rain is

---

present incessantly. Female header pins also had to be attached to the sensor so it could connect to the pi.

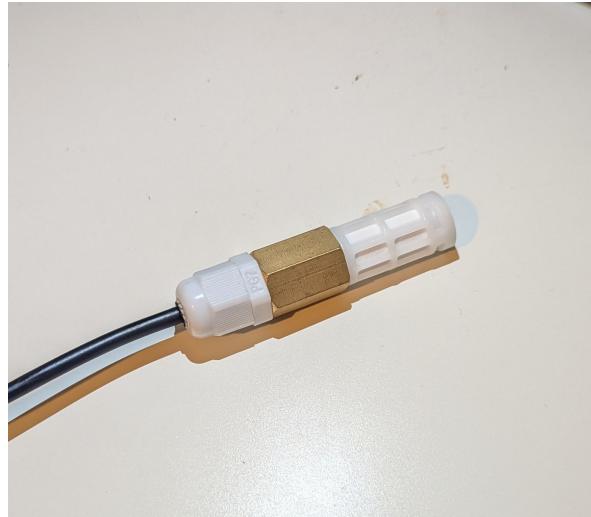


Figure 3.2: SHT40

### 3.3.2 BH1750 (Light Intensity Sensor)

The BH1750 is a digital ambient light sensor designed for direct lux measurement with a 16-bit output. The sensor can detect illuminance in a wide dynamic range from 1 to 65,535lux. It operates over a supply voltage range of 2.4V to 3.6V and communicates via the I<sup>2</sup>C interface. The sensor has three possible measurement modes: high resolution which can measure changes as small as 1lux, high-resolution 2 which can measure changes to 0.5lux and low-resolution which can measure changes as small as 4lux. The I<sup>2</sup>C slave address can be configured as either 0x23 or 0x5C depending on the state of the ADDR pin [36]. The sensor also arrived with the male header pins not attached so a solder put in place to fasten it.

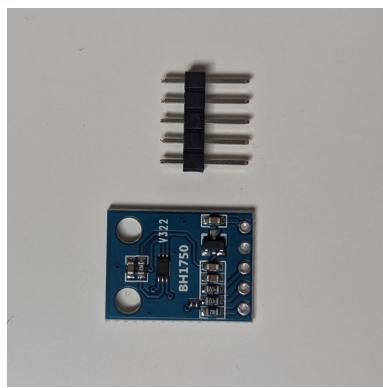


Figure 3.3: BH1750 before solder

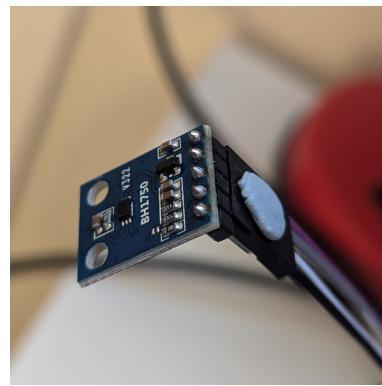


Figure 3.4: BH1750 after solder

---

## 3.4 Imaging Hardware

### 3.4.1 RGB Camera

The RGB imaging device used in this system is the Raspberry Pi High Quality (HQ) Camera, which features a 12.3-megapixel Sony IMX477 sensor. The camera is connected to the Raspberry Pi via the Camera Serial Interface (CSI) and is controlled using libcamera software. The IMX477 sensor provides high-resolution still images with a maximum output of  $4056 \times 3040$  pixels. A 6 mm 3MP HD CCTV lens is used providing a wide field of view while maintaining sharp image quality across the frame. The lens uses a standard CS-mount and is manually focused to achieve consistent image sharpness at the target distance. Images are captured at fixed 1-minute intervals, with each capture timestamped and saved locally to an external storage device.

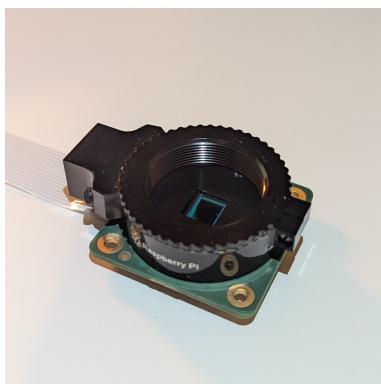


Figure 3.5: RGB Mount



Figure 3.6: RGB Lens

### 3.4.2 NIR Camera

The near-infrared (NIR) imaging system is based on the Waveshare IMX219-160 IR-Cut Camera, which utilizes an 8-megapixel Sony IMX219 sensor allowing it to capture images with a maximum output of  $3280 \times 2465$  pixels. This module originally includes an automatic IR-cut filter and side-mounted IR LEDs [37]; however, the IR lights have been removed and the cut filter disabled to allow continuous sensitivity to near-infrared wavelengths. The side-mounted IR lights provide infrared illumination for low light conditions but this was not needed as it is only useful for the immediate surroundings and not the sky. The cut filter allowed the camera to switch from NIR to RGB at specific light levels but this was not needed either. This keeps the camera shooting in NIR mode all the time. The camera is equipped with a wide-angle lens ( $160^\circ$  FOV). Captured NIR images are also timestamped and stored locally at fixed intervals.

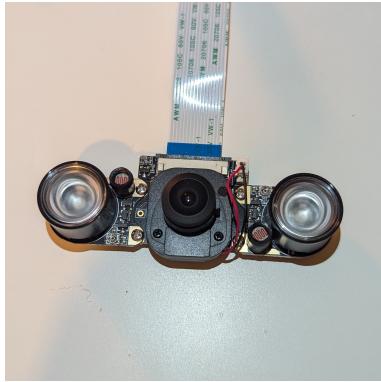


Figure 3.7: NIR Before



Figure 3.8: NIR After modifications

## 3.5 Computational Hardware and Data Storage

### 3.5.1 Raspberry Pi 4 Model B

The core of the system is a Raspberry Pi 4 Model B with 4 GB of RAM (see Figure 3.9), running Raspberry Pi OS (64-bit). The operating system was installed to a 64 GB microSD card using the official Raspberry Pi Imager tool, which enables straightforward flashing of OS images and initial configuration. The Pi 4B features a quad-core ARM Cortex-A72 processor clocked at 1.5 GHz, dual-band Wi-Fi, Bluetooth 5.0, Gigabit Ethernet, two USB 3.0 ports, and dual micro-HDMI outputs. It is responsible for sensor interfacing, image capture, data timestamping, and writing to external storage [38].

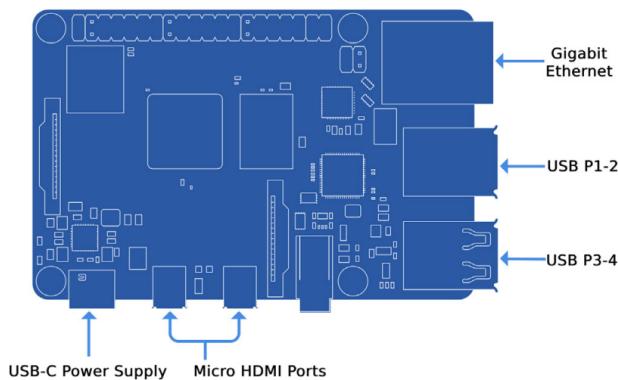


Figure 3.9: Raspberry Pi 4 Model B hardware layout

### 3.5.2 External Hard Drives

Data from the Raspberry Pi is logged to external USB storage devices formatted with the exFAT file system for cross-platform compatibility. Two drives are used: a 2 TB external hard drive and a 128 GB flash drive. These devices store environmental sensor readings and image data captured at fixed one-minute intervals. Each data point consists of a single line of comma-separated values (CSV) and two high-resolution JPEG images—one from the RGB camera and one from the NIR

---

camera. Each minute, the system writes approximately 2.5–5 MB of data, depending on image compression and environmental conditions. Over 24 hours, this totals roughly 3.5–7 GB of data so the current setup should be valid for at least 18 days.



Figure 3.10: USB Stick



Figure 3.11: USB Drive

### 3.5.3 Data Handling Responsibilities

- Pi A: Handles NIR image capture and light sensing (BH1750)
- Pi B: Handles RGB image capture and temperature/humidity data (SHT40)
- Each Pi independently timestamps and stores data to local storage

---

# Chapter 4: Enclosure Design

---

A custom weatherproof enclosure was developed to house the data acquisition components and protect them from environmental exposure during long-term outdoor deployment. This chapter details the mechanical design, materials selection, assembly process, and weatherproofing strategies used.



Figure 4.1: Full enclosure

## 4.1 Design Requirements

The enclosure was required to meet the following criteria:

- Resist water, dust, and insects
- Allow visibility for cameras while minimizing glare or and dew build up
- Provide stable mounting for Raspberry Pi boards and sensors
- Enable access for maintenance and data extraction
- Withstand varied weather conditions (rain, wind, temperature fluctuations)

## 4.2 Junction Box Enclosure

A commercial-grade plastic junction box [39] was used as the core of the enclosure. The chosen box is weatherproof rated to **IP56**, offering strong protection against dust, rain, and hose spray, making it well-suited for outdoor deployment. It features a durable, solid grey PVC plastic construction with ribbed internal surfaces to improve structural strength and provide elevation, allowing air circulation beneath mounted components and minimizing the risk of water pooling around electronics in case of ingress.

The enclosure has a hinged cover secured with four plastic screws. Two of these screws can be partially backed out to function as a makeshift hinge, allowing easier access for maintenance.



Figure 4.2: Junction box used as primary enclosure.

### Modifications made:

- Two circular holes were drilled in the lid to mount the RGB (Red,green,blue) and NIR (Near-infrared) cameras, one of which includes additional ventilation holes to promote airflow. See Fig (4.3)
- Three holes were drilled in the bottom of the box to fit cable glands (Fig 4.4):
  - Two for supplying power to each Raspberry Pi.
  - One for routing the SHT40 sensor cable to the exterior.



Figure 4.3: Lid holes

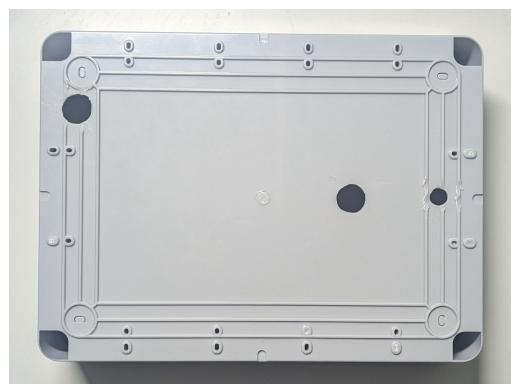


Figure 4.4: Bottom holes

## 4.3 Acrylic Domes for Camera Protection

Clear acrylic half-sphere domes were used to protect the RGB and NIR cameras while preserving optical clarity and environmental resistance. Two domes of different sizes were mounted over precision-cut circular openings in the junction box lid. These domes were selected for their high transparency (94%), low distortion factor (0.16%), and excellent resistance to UV exposure and aging.

- **80 mm inner diameter dome** — Outer diameter: approx. 99 mm, height: 45 mm, center hole spacing: 62 mm / 88 mm [40]
- **100 mm inner diameter dome** — Outer diameter: approx. 119 mm, height: 55 mm, center hole spacing: 77 mm / 109 mm [41]
- Both domes are UV-resistant, optically clear, and designed for minimal image distortion

Each dome was sealed to the enclosure using a clear, outdoor-grade silicone sealant to ensure weatherproofing and prevent moisture ingress.

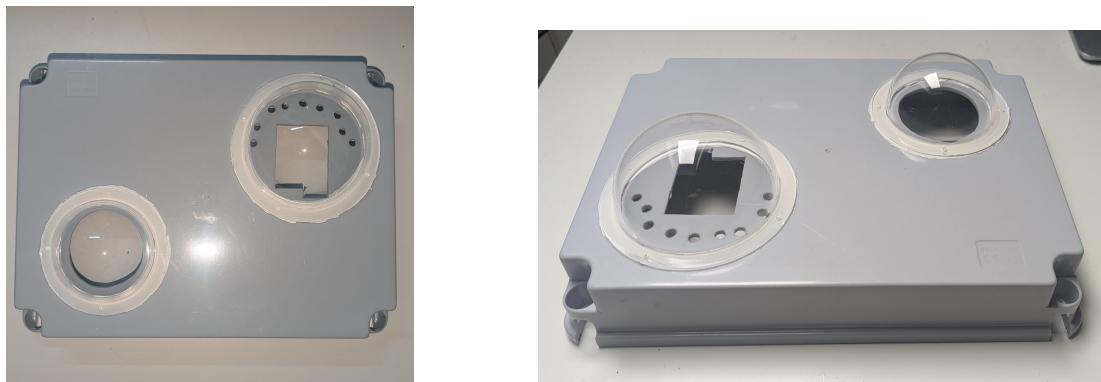


Figure 4.5: Acrylic domes mounted over camera openings from two different angles.

## 4.4 Cable Gland Integration

To maintain the IP rating of the enclosure while enabling cable access, IP68-rated cable glands were installed in the junction box. These glands provided reliable strain relief and watertight seals for all external wiring, ensuring safe operation under outdoor conditions. The glands used were made from high-quality nylon and resistant to UV exposure [42].

- **2 × M20 cable glands** were used to route power supply cables to the Raspberry Pi units.
- **1 × M12 cable gland** was used to pass the SHT40 sensor cable to the external environment.

The adjustable and multi-threaded design of these glands allowed for quick installation and a secure fit with the enclosure bottom contributing to the system's long-term durability in challenging weather conditions.



Figure 4.6: Lid holes



Figure 4.7: Bottom holes

## 4.5 Wooden Support and Mounting Frame

To keep costs low and provide a stable internal layout, wooden plates and risers were cut to size and mounted inside the box. These provided support for, Raspberry Pi mounting and access keeping them off the box floor. More wooden risers supplied the cameras with a direct line of sight through domes and being close enough to prevent glare.

The full support frame is shown in Figure 4.8.

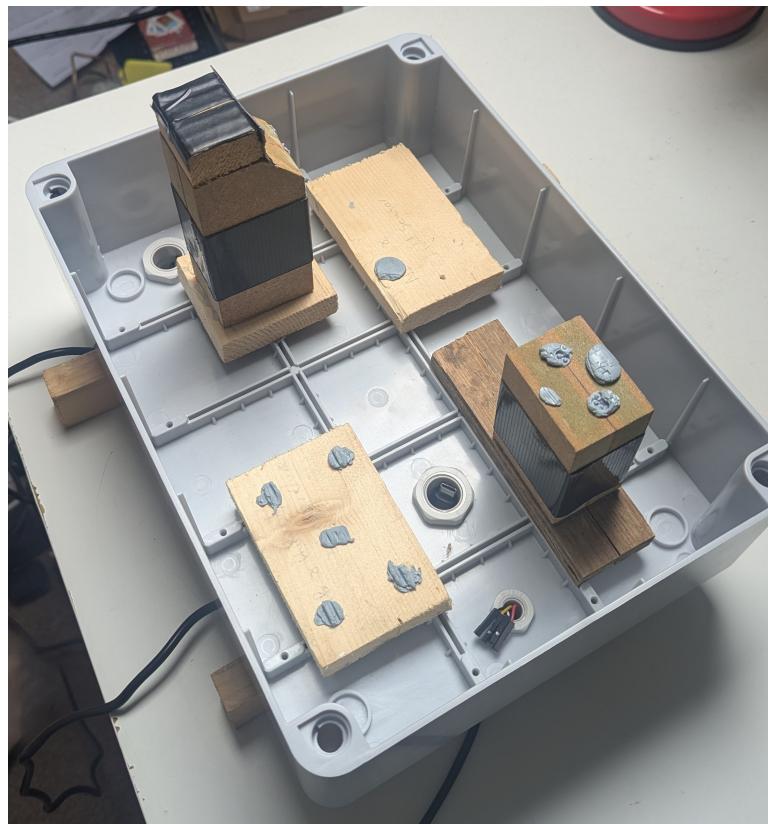


Figure 4.8: Wooden mounting system installed inside enclosure.

## 4.6 CAD Model and Dimensions

A CAD model of the final layout was developed to assist with component positioning and dome placement and for displaying purposes. The CAD was created using Solidworks [A.3].

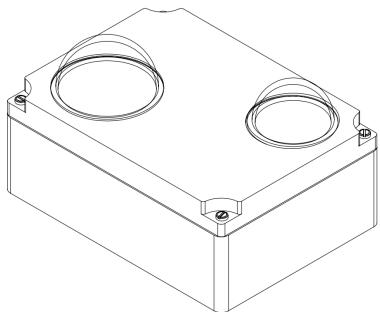


Figure 4.9: Isometric View

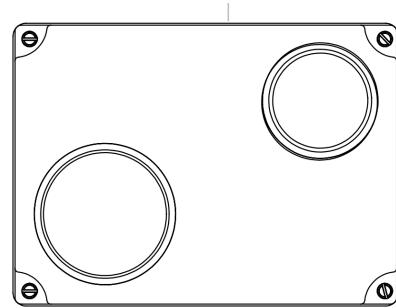


Figure 4.10: Top View

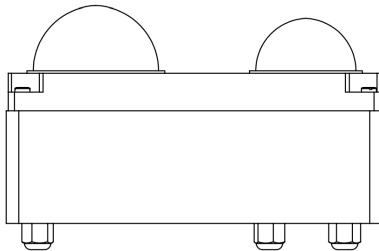


Figure 4.11: Side View

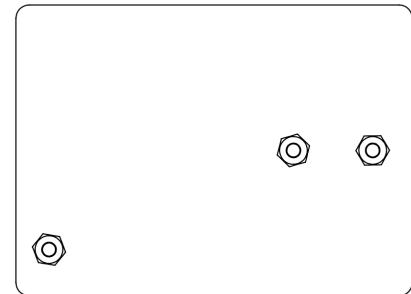


Figure 4.12: Bottom View

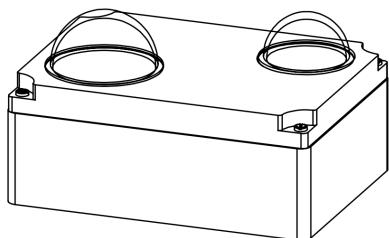


Figure 4.13: Dimetric View Angle

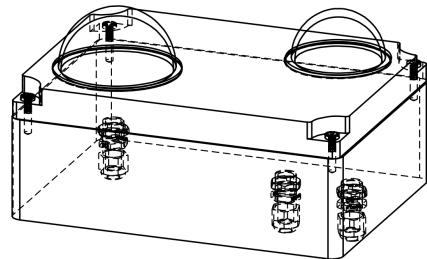


Figure 4.14: Dimetric View (with hidden detail)

Figure 4.15: CAD model of the enclosure showing multiple perspectives: isometric, orthographic, and alternative angles, including seethrough rendering for internal visualization.

---

## 4.7 Discussion

During testing, the enclosure remained dry through moderate rain and daily temperature variation, indicating good seal integrity. Condensation and dust ingress were minimal. Future improvements may include:

- Internal temperature/humidity sensor for monitoring the internal state of the device
- Fans for additional ventilation for deployments in hotter climates
- 3D-printed internal brackets to replace the wooden ones
- Heating coil for inside camera domes to prevent dew

---

# Chapter 5: System Design, Deployment, and Monitoring

---

This chapter describes the physical and electrical layout of the data acquisition system. It covers internal component placement, cable routing and power delivery. The system was designed with modularity, durability, and weather resistance in mind to ensure reliable long-term operation in outdoor environments. It also describes the field deployment process, platform setup, and orientation strategies designed to maximize data quality and ensure stability under environmental conditions. Finally, the remote monitoring and operational procedures are explained, detailing how the system was managed, controlled, and maintained during data collection.

## 5.1 Internal Layout

All components were arranged compactly inside a weatherproof junction box. To improve airflow and simplify cable routing, two custom wooden risers were installed to mount the Raspberry Pi boards above the base level. This elevated design also helps protect the electronics in the unlikely event of water ingress.

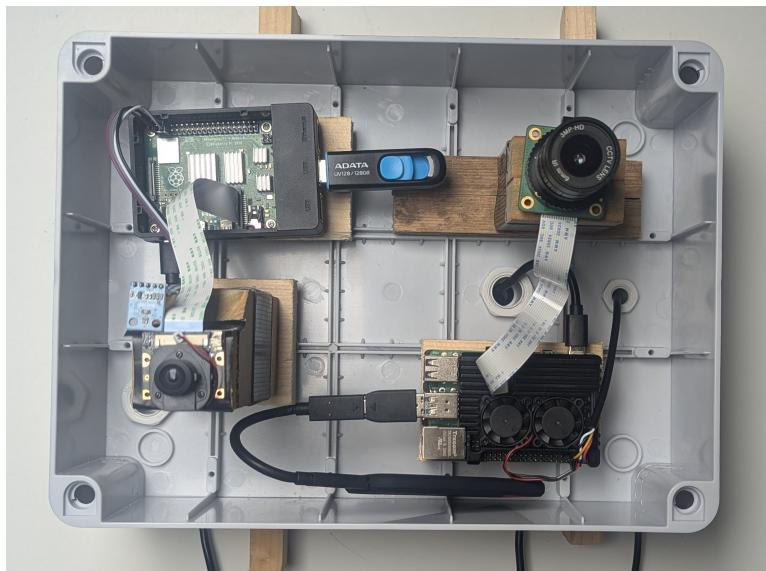


Figure 5.1: Interior view of the system enclosure showing mounted Raspberry Pis and cable routing.

## 5.2 Component Interconnections

Each Raspberry Pi is wired to its own dedicated camera and I2C sensor. The following figures illustrate the component connections, including power, I2C communication lines, and camera

---

interface.

## NIR Raspberry Pi Wiring

Figure 5.2 shows the wiring for the Raspberry Pi responsible for capturing near-infrared (NIR) images and measuring ambient light levels.

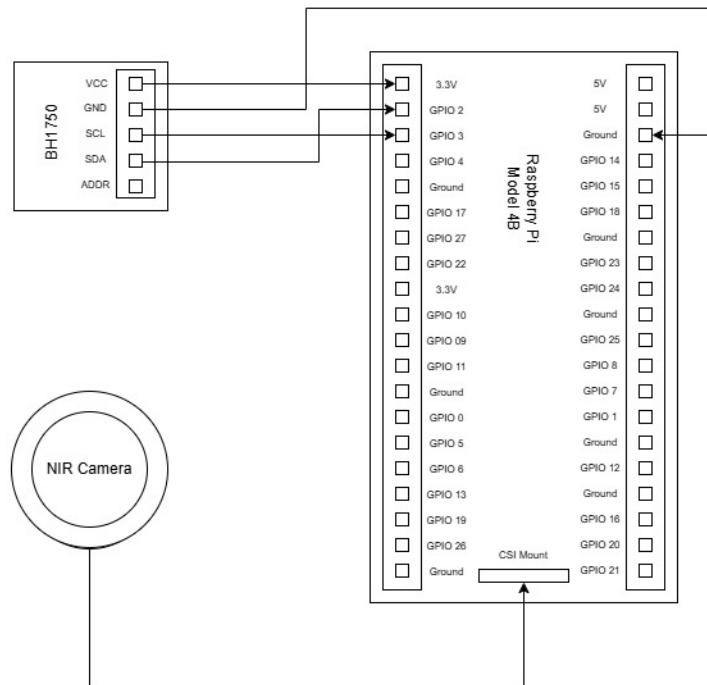


Figure 5.2: NIR system schematic: BH1750 sensor and NIR camera wiring to Raspberry Pi.

In this setup:

- The BH1750 is connected via the I<sup>2</sup>C bus using GPIO 2 (SDA) and GPIO 3 (SCL). It measures the ambient light intensity in lux.
- The sensor is powered from the 3.3V pin and grounded using a GPIO ground pin.
- The NIR camera is connected to the CSI camera port of the Raspberry Pi using a ribbon cable. No additional GPIO connections are required for the camera.

## RGB Raspberry Pi Wiring

Figure 5.3 presents the wiring for the Raspberry Pi that captures RGB images and records environmental temperature and humidity.

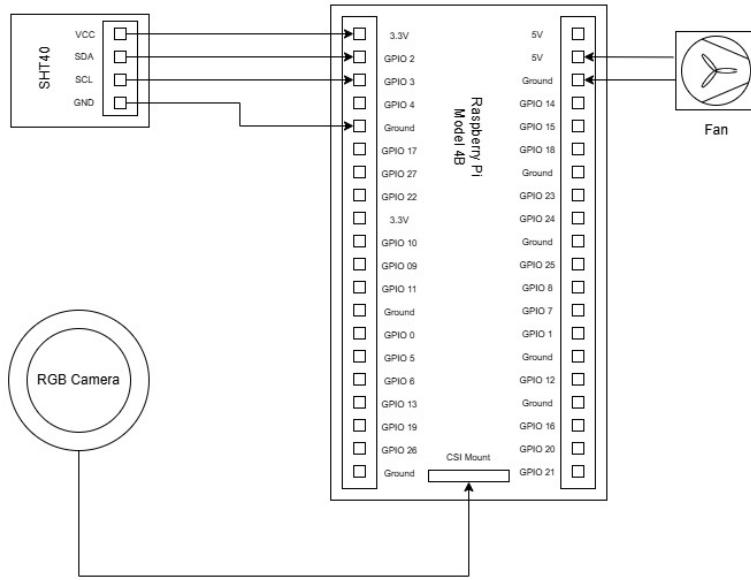


Figure 5.3: RGB system schematic: SHT4x sensor, RGB camera, and fan wiring to Raspberry Pi.

In this configuration:

- The SHT40 sensor is used to measure temperature and humidity, and it also communicates over I<sup>2</sup>C using GPIO 2 (SDA) and GPIO 3 (SCL).
- It receives power from the 3.3V pin and shares a ground connection.
- The RGB camera connects via the CSI port, similar to the NIR camera setup.
- A 5V fan is powered directly from the 5V GPIO rail and grounded to help with airflow and cooling within the enclosure.

Both Raspberry Pis are connected to: USB-based SSDs for local data storage and independent 5V USB-C power supplies routed through IP68-rated cable glands to maintain waterproof integrity.

## 5.3 Power and Cable Management

Both Raspberry Pi units are powered by dedicated 5V USB-C power cables, selected for their:

- **Extended length (2 meters):** Allows flexible mounting.
- **Durable jacket:** Offers protection against abrasion and UV exposure.
- **Weatherproofing:** Cables and plug is one device and not separate components which would offer more water protection as there is less points of entry.

## 5.4 Deployment

The environmental data acquisition system was deployed outdoors on a metal platform to ensure unobstructed exposure to sunlight and far from trees and other object that could cast shadows. The enclosure, which housed the cameras and sensors, was secured in place using elastic bungee cords tensioned diagonally across the top. This setup ensured mechanical stability in the face of wind while allowing for quick access if needed.



Figure 5.4: Chords Securing the Device

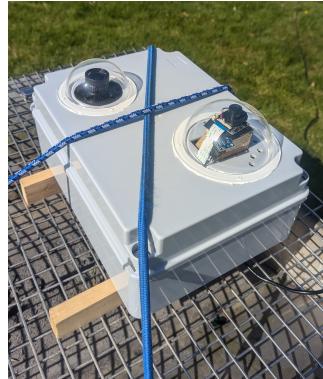


Figure 5.5: Device in position

## 5.5 Platform and Orientation

The enclosure was elevated slightly above the platform using wooden spacers. The orientation was chosen to provide an unobstructed view of the sky, maximizing light capture by the sensors and cameras. It was ensure the platform was level using a levelling tool and adjusting the wooden spacers as needed.

## 5.6 Remote Operation and Monitoring

The system was powered via a stable DC supply and interfaced using a local Wi-Fi connection. An extension lead was ran to the platform where the raspberry pi plugs were attached and this was enclosed using plastic to ensure waterproofing. All scripts were run and monitored remotely using SSH (Secure Shell) access from a host computer. The primary data logging script was launched using the "no hang up" utility to ensure persistent execution [A.2.9]:

```
nohup python3 coordinated_script.py > output_logs.txt 2>&1 &
```

System health and runtime logs were written to output\_logs.txt, allowing real-time inspection of temperature, humidity, lux values, and image capture success. Periodic checks were performed remotely to validate correct operation, monitor storage space, and reboot the system if necessary.

# Chapter 6: Software and Data Handling

---

This chapter details the software architecture responsible for controlling data capture, managing storage, and preparing the data for analysis.

## 6.1 Overview

The system consists of two Raspberry Pi units, each responsible for data acquisition from a different modality (RGB or NIR). All software is implemented in Python and follows a modular design to support sensor interfacing, image capture, logging, and fault-tolerant execution.

Each device runs a coordinated script that invokes functions from dedicated modules for capturing environmental data, acquiring images, and writing to a shared CSV log. Data is saved to external SSDs mounted via USB.

## 6.2 Modular Script Design

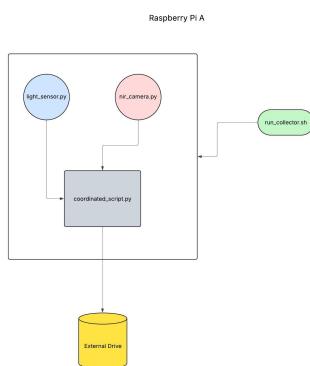


Figure 6.1: NIR Software Flow

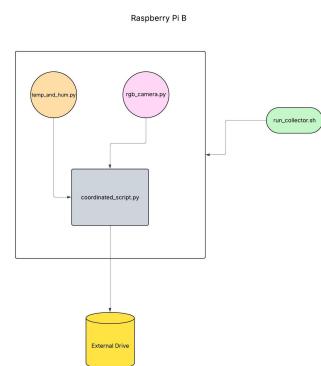


Figure 6.2: RGB Software Flow

The architecture is separated into four key modules:

- `rgb_camera.py` and `nir_camera.py` – control image capture via `libcamera` using sensor-aware exposure settings.
- `temp_and_hum.py` – reads temperature and humidity values from the Adafruit SHT4x sensor.
- `light_sensor.py` – reads illuminance (lux) using a BH1750 light sensor and estimates solar irradiance.
- `coordinated_script.py` – the main execution script, which synchronizes all actions per minute, performs error handling, and logs output.

---

## 6.3 Image Capture

Images are acquired using the `libcamera` suite, a powerful Linux-based camera stack specifically developed for the Raspberry Pi. The `libcamera` library provides a modern camera stack and enables full control over image acquisition parameters through command-line utilities such as `libcamera-still` and `libcamera-jpeg`. This project used:

- **Gain:** Increases image brightness by amplifying the signal after it is captured. Useful in low-light conditions.
- **Shutter speed:** Controls exposure duration. Longer exposure helps in dark scenes; shorter helps prevent overexposure in bright light.
- **AWB (Auto White Balance):** Balances color tones based on ambient lighting. A custom mode with fixed awbgains was used for NIR imaging.
- **Saturation:** Used for colour enhancement or suppression.

These options provided the flexibility required for capturing consistent images under a wide range of lighting conditions.

### 6.3.1 NIR Camera Settings

The Near-Infrared camera captures images using `libcamera-still` with exposure settings dynamically adjusted based on ambient light levels. The logic is implemented in `nir_camera.py` and uses the lux value from a BH1750 sensor to choose between three capture modes [A.2.2] :

- **Night Mode (lux < 10):**

```
libcamera-still --shutter 6000000 --gain 32
```

This configuration uses a 6-second shutter and high gain to capture usable images in very low light.

- **Sunny Mode (lux < 15000):**

```
libcamera-still --shutter 500 --gain 1 --awb custom --awbgains 1.2,1.2
```

Fast shutter and low gain prevent overexposure during bright daylight.

- **Normal Mode (lux >= 15000):**

```
libcamera-still --awb custom --awbgains 1.2,1.2
```

A moderate exposure mode with a fixed white balance suited for consistent NIR image characteristics. The custom gains were found through trial and error.



Figure 6.3: Without AWB



Figure 6.4: With AWB

This design allows the system to adapt automatically to real-world lighting conditions without user intervention.

### 6.3.2 RGB Camera Settings

The RGB camera uses `libcamera-jpeg`, which is a simplified utility for capturing JPEGs. The script `rgb_camera.py` calls [A.2.3]:

```
libcamera-jpeg -o [output_filename]
```

The RGB images were captured using default automatic settings, as the color fidelity and brightness provided by `libcamera`'s default exposure and white balance were sufficient under typical daylight conditions. This minimal configuration helps reduce system complexity while maintaining high-quality outputs suitable for downstream image processing.

If lighting conditions change significantly in the future, parameters such as `-shutter`, `-gain`, or `-awb` could be easily tuned.

## 6.4 Environmental Data Capture

Environmental data is collected using two I<sup>2</sup>C-based sensors: the SHT40 for temperature and humidity, and the BH1750 for ambient light intensity. Both sensors operate on the I<sup>2</sup>C bus and are polled once per minute in coordination with the image capture routine.

### 6.4.1 I<sup>2</sup>C

The Inter-Integrated Circuit (I<sup>2</sup>C) protocol is a widely used two-wire serial communication interface, ideal for short-distance communication between microcontrollers and peripherals such as sensors. It uses two lines, SDA (Serial Data Line) used for sending and receiving data and SCL (Serial Clock Line) used to provide the timing signal.

On the Raspberry Pi, the default I<sup>2</sup>C pins are:

- 
- GPIO2 (SDA) → Sensor SDA
  - GPIO3 (SCL) → Sensor SCL

Sensors are powered via the 3.3V or 5V GPIO header, with a common ground connection. Both sensors use the I<sup>2</sup>C protocol but at different levels of abstraction, with the SHT40 accessed through a high-level driver that handles communication and data parsing internally [A.2.7], while the BH1750 requires low-level I<sup>2</sup>C commands, manual timing control, and explicit handling of raw measurement data [A.2.6]

## 6.4.2 SHT40

The SHT40 is a digital temperature and humidity sensor from Sensirion. It offers high accuracy and stability, making it suitable for outdoor environmental monitoring. The sensor communicates via I<sup>2</sup>C and is polled using the Adafruit CircuitPython library.

Each call to `read_temp_and_humidity()` returns a pair of floating-point values representing the ambient temperature (in °C) and relative humidity (in %) [A.2.7].

## 6.4.3 BH1750

The BH1750 is a digital ambient light sensor with a 16-bit output over I<sup>2</sup>C. The BH1750 supports multiple measurement modes as discussed (link to hardware overview)

In this project, One-Time High-Resolution Mode was selected via the instruction opcode 0x20, ensuring power-efficient operation and automatic return to power-down mode after each measurement.

### MTreg: Measurement Time Register

MTreg is a configurable register in the BH1750 sensor that controls the sensor's integration time, directly affecting its sensitivity and measurement range, higher values increase sensitivity for low-light conditions, while lower values reduce exposure to handle bright light. The default MTreg value is 69, and it controls both integration time and sensor sensitivity:

MTreg is configured using two I<sup>2</sup>C commands to set the upper 3 bits and lower 5 bits separately. The code dynamically selects MTreg based on an initial trial measurement [A.2.4].

### Lux Calculation and Sensitivity Ratio

Once a measurement is taken, the BH1750 returns a 16-bit value (high byte and low byte) which must be converted into a lux value. The datasheet [36] specifies a typical conversion formula:

$$\text{Lux} = \frac{\text{Raw Count}}{1.2} \times \left( \frac{69}{\text{MTreg}} \right)$$

This formula adjusts for the MTreg value, normalizing the sensitivity back to a standard lux scale [A.2.5].

---

#### 6.4.4 Solar Irradiance Estimation

To support downstream solar forecasting, lux measurements are converted to approximate solar irradiance values using the empirical ratio [43]:

$$\text{Irradiance (W/m}^2\text{)} = \frac{1}{120} \times \text{Lux}$$

This allows the BH1750 to serve as a low-cost, indirect irradiance sensor for analysis [A.2.8]

## 6.5 Data Logging

Environmental and visual data are logged locally on a mounted external SSD to ensure persistence and portability. Each Raspberry Pi device maintains its own structured directory, where sensor readings and image captures are saved in a consistent and synchronized format.

All data is saved in a structured format on the mounted SSD:

```
/media/[device]/Project/  
    photos_rgb/  
    photos_nir/  
    data.csv
```

Each image is timestamped at the time of capture and saved to its respective subdirectory, `photos_rgb/` or `photos_nir/`. The corresponding filename is recorded in a central CSV file to enable easy traceability and data-image pairing during analysis.

Sensor values are polled once per minute, alongside image acquisition, and appended as a new row in `data.csv`. See [A.4.1] and [A.4.2] for NIR and RGB system example data.

Images are saved with filenames formatted as:

`YYYY-MM-DD_HH-MM_[rgb|nir].jpg`

This convention ensures both uniqueness and chronological sortability, enabling automated parsing during post-processing.

The main coordination script checks if the target directory exists and creates it if needed. It also verifies that the storage device is correctly mounted before proceeding with data logging to avoid writing to non-persistent memory. Errors during file writing or sensor polling are caught using exception handling, and logs are recorded to a system log file for debugging and reliability tracking.

---

## 6.6 Scheduling and Execution

The data acquisition scripts are designed to run continuously in an infinite loop, with internal scheduling logic that ensures data is captured precisely once per minute [A.2.1]. The script uses the system clock to check the current time and triggers data capture when the seconds field reaches zero. The timing loop waits briefly between checks to avoid busy waiting and unnecessary CPU usage.

Startup is managed using a shell script that launches the Python logger in the background while redirecting standard output and errors to a log file for monitoring [A.2.9]:

```
nohup python3 coordinated_script.py > output_logs.txt 2>&1 &
```

This approach ensures that the script remains active even if the terminal session is closed, which is essential for long-term unattended operation and because SSH was the main method of access to the device. Logs written to `output_logs.txt` include timestamps and status messages to assist in system monitoring and debugging.

## 6.7 Results – Processing and Modelling

### Dataset Structure

Two datasets, `train_with_cam_info.csv` and `test_with_cam_info.csv` were used for training and evaluation [A.4.3]. Each contained timestamped entries of:

- Temperature and humidity (from the SHT40)
- Lux and derived solar irradiance (from the BH1750)
- Mean red channel value (from RGB images)
- Histogram-derived brightness levels (from NIR images)

The data were structured into input-output windows with a look-back period of 30 time steps (i.e., 30 minutes), enabling the model to learn temporal trends.

### Normalization with Min-Max Scaling

Prior to modelling, all features were normalized using a Min-Max scaling transformation to ensure that each feature contributed equally during training and to improve model convergence. This scaling mapped all input features to the range [0, 1].

The Min-Max scaling formula is:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

where:

- 
- $x$  is the original feature value,
  - $x_{\min}, x_{\max}$  are the minimum and maximum values of that feature in the training set,
  - $x'$  is the normalized value.

This transformation preserved the original data distribution while ensuring that all features operated within the same numeric scale, facilitating efficient model training [A.2.10].

### 6.7.1 LSTM Model Architecture

#### Network Structure

The forecasting model was implemented using TensorFlow and Keras, adopting a Bidirectional LSTM architecture to capture both past and future dependencies in the time series data [A.2.11]. The model structure is as follows:

- **Input layer:** Accepts multivariate time series inputs, with the shape defined by the window length and number of features.
- **Bidirectional LSTM layer:** Consists of 512 units, designed to capture temporal dependencies in both forward and backward directions.
- **Dropout layer:** A dropout rate of 0.5 was applied to prevent overfitting by randomly deactivating neurons during training.
- **Dense hidden layer:** Contains 8 units with a ReLU activation function to introduce non-linearity and enable the model to learn complex patterns.
- **Dense output layer:** A single neuron with a linear activation function to produce the final irradiance prediction.

#### Activation Function: ReLU

The Rectified Linear Unit (ReLU) activation function was used in the dense hidden layer to introduce non-linearity:

$$\text{ReLU}(x) = \max(0, x)$$

### 6.7.2 Training Configuration

#### Optimizer: Adam

The network was compiled using the **Adam optimizer** with a learning rate of 0.001. Adam is particularly effective for time-series models like this one, as it adapts learning rates for individual parameters and utilizes running averages of both the gradients and their squared values to adjust updates. This allows for more stable and efficient convergence, especially in non-stationary environments such as solar irradiance forecasting.

---

The Adam optimizer combines the advantages of two other methods **AdaGrad**, which adapts learning rates for each parameter, and **RMSProp**, which adjusts learning rates based on recent gradient magnitudes. The parameter update rule for Adam is given by:

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

where:

- $\theta_t$  is the updated parameter at time step  $t$
- $\eta$  is the learning rate
- $\hat{m}_t$  and  $\hat{v}_t$  are the bias-corrected estimates of the first and second moments of the gradients
- $\epsilon$  is a small constant to prevent division by zero

## Training Hyperparameters

The LSTM model was trained with the following hyperparameters, designed to balance convergence speed and generalization:

- **Learning rate:** 0.001
- **Batch size:** 1024
- **Epochs:** 500 (with early stopping applied to halt training when necessary)
- **Loss function:** Mean Squared Error (MSE)
- **Evaluation metric:** MSE during training, with Root Mean Squared Error (RMSE) calculated on unscaled test data for interpretability.

**Window Size and Lead Time:** A **window size** of 30 time steps was used (`MULTIV_TRAINING_WINDOW`), providing the model with sequences of past environmental data and image-derived features. Additionally, a **lead time** of 10 time steps was incorporated, meaning the model forecasts solar irradiance 10 minutes ahead rather than the immediate next point. This design choice aligns with the requirements of solar forecasting, where anticipating conditions further into the future is essential for grid and energy management.

## Callbacks

To ensure robust training and mitigate overfitting, two key callbacks were implemented:

- **EarlyStopping:** Monitored the validation loss with a patience of 5 epochs. If no improvement in validation loss was observed for five consecutive epochs, training was halted, and the best-performing model weights were restored.
- **ModelCheckpoint:** Saved the best model based on the lowest validation loss throughout training to ensure that the optimal configuration was preserved for deployment. The final model was stored at `models/second_deployment.keras`.

This configuration provided a stable training environment, ensured convergence without overfitting, and preserved the most effective model weights for accurate solar irradiance forecasting.

### 6.7.3 Model Evaluation

**Mean Squared Error (MSE) and Root Mean Squared Error (RMSE)** The model's performance was evaluated using two common regression metrics: Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

**Mean Squared Error** measures the average squared difference between predicted and actual values:

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

**Root Mean Squared Error** is the square root of MSE:

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

where:

- $y_i$  is the actual value
- $\hat{y}_i$  is the predicted value
- $n$  is the total number of predictions

**MSE** is useful for penalizing large errors more heavily due to squaring, which is helpful when outliers are important. **RMSE**, being in the same unit as the target variable, offers a more interpretable measure of average prediction error.

In this study, both metrics were used to assess the model's precision in predicting solar irradiance, with RMSE being the primary score reported for interpretability.

### 6.7.4 Image Feature Extracting

Image-derived features were extracted from both Near-Infrared (NIR) and RGB images to enhance the solar irradiance forecasting model. For each data point, the associated NIR and RGB images were processed using the OpenCV library in Python.

**NIR Image Features:**

- **NIR Brightness:** The mean grayscale intensity of the NIR image, representing the average infrared reflectivity.
- **NIR Brightness Standard Deviation:** The spread of grayscale pixel intensities, indicating contrast within the NIR image.
- **NIR Red Channel Mean:** The average intensity of the red channel (which carries the NIR signal).

---

### RGB Image Features:

- **RGB Brightness:** The mean grayscale intensity of the RGB image, representing overall visible light brightness.
- **RGB Brightness Standard Deviation:** The variation in grayscale pixel intensities, indicating contrast in visible light.
- **RGB Red, Green, and Blue Channel Means:** The average intensities of each respective channel, reflecting visible light color distribution.

These features capture key information about cloud cover, brightness, and atmospheric conditions, improving the model's ability to predict solar irradiance fluctuations due to dynamic weather patterns [A.2.13].

---

# Chapter 7: Data Insights and Predictive Modeling for Solar Irradiance Forecasting

---

This section presents the outcomes of data analysis and modeling, focusing on correlations between solar irradiance and both image-derived and environmental features, the performance of a predictive LSTM model, and an exploration of how image characteristics vary under different lighting conditions.

## 7.1 Deployment Summary and Data Context

The data acquisition system was deployed outdoors in Longford, Ireland over a two-week period. During this time, the system collected synchronized environmental sensor data (including temperature, humidity, and light intensity) alongside RGB (Red, Green, Blue) and NIR (Near-Infrared) sky images. The RGB camera captured images at a resolution of  $4056 \times 3040$  pixels, while the NIR camera recorded at  $3280 \times 2464$  pixels. Images and sensor readings were collected at one-minute intervals, ensuring fine-grained temporal alignment between environmental conditions and sky imagery.

In total, the system gathered 9947 RGB [A.1] and 8473 NIR images [A.2], with 9953 corresponding lux and solar irradiance readings from the NIR system and 8480 corresponding temperature and humidity readings from the RGB system. This means that in the data file there was 9953 rows of readings and image paths [A.4] and 8480 rows of readings and image paths [A.3] from the RGB system. However, due to the iterative nature of the deployment, which included multiple hardware adjustments, enclosure refinements, and script debugging—the final dataset consists of four complete days of continuous, high-quality data.

The data collection period coincided with an unseasonal heatwave in March, resulting in predominantly clear-sky conditions and consistently high levels of solar irradiance. This climatic consistency, limited the variability in the dataset, particularly in terms of capturing diverse cloud formations, overcast skies, or transitional weather patterns. This constraint inherently affects the generalizability of some findings, especially in scenarios involving dynamic or complex atmospheric conditions.

Nevertheless, this dataset serves as a proof of concept for the data acquisition system and demonstrates its potential for generating meaningful insights through the integration of sky imaging and environmental sensing. Whose use is highlighted in the subsequent analysis sections.

Future work will involve stationing the imager for extended periods, allowing for the collection of a richer and more diverse dataset that captures a broader range of weather conditions and sky states. This will support more comprehensive model training and evaluation, ultimately improving the robustness and accuracy of solar irradiance forecasts, especially under varied atmospheric scenarios.

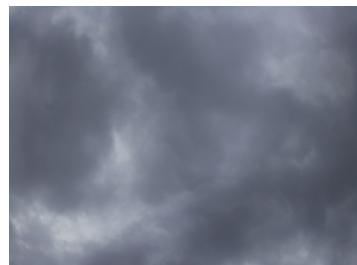
The current results, despite their limitations, lay the groundwork for ongoing sky imaging and environmental data collection, with the potential to scale up into a long-term, real-time solar

---

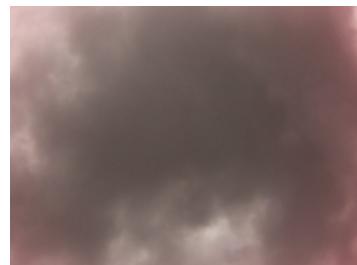
monitoring and forecasting system that leverages both spectral imaging and environmental sensing system.



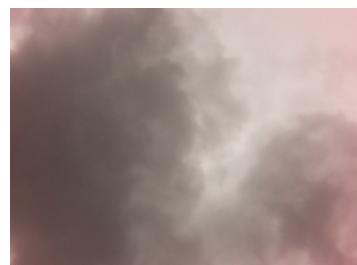
RGB and aligned NIR sky images at 13:32.



RGB and aligned NIR sky images at 13:34.



RGB and aligned NIR sky images at 13:37.



RGB and aligned NIR sky images at 13:39.

Figure 7.1

Example pairs of RGB and aligned NIR sky images are presented in [Figure 7.1](#), illustrating the visual differences between the two spectral bands under various sky conditions. Each row shows the RGB image (left) and the corresponding aligned NIR image (right).

## 7.2 Feature Correlation Analysis

To understand the relationships between the collected variables and solar irradiance, a Pearson correlation matrix was generated from the collected data.

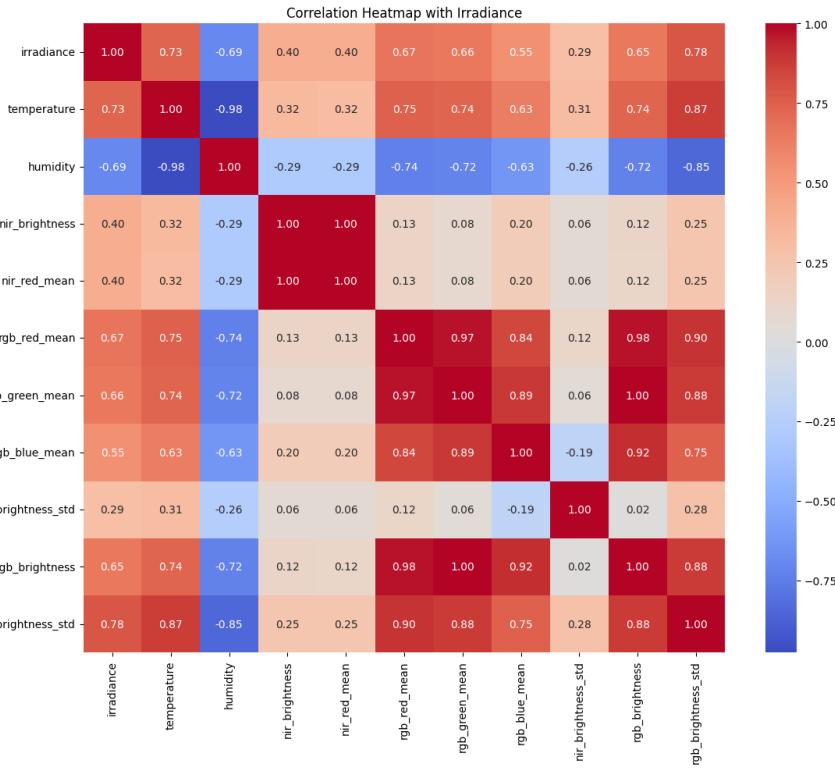


Figure 7.2: Correlation matrix showing relationships between features and solar irradiance

Several key patterns emerge from the correlation heatmap:

- **Temperature** shows strong positive correlations with irradiance. This is consistent with the physical expectation air temperature rise under high solar irradiance.
- **RGB-derived features** performed well in predicting irradiance, even under persistently clear, blue skies. This effectiveness stems from the RGB camera's ability to capture subtle variations in sky color and brightness, which are influenced by the sun's position and atmospheric interactions throughout the day. I expect this could be the camera picking up on Rayleigh scattering, where shorter wavelengths of light (blue) scatter more than longer wavelengths (red) as sunlight passes through the atmosphere. This scattering causes the sky to appear blue during midday and shifts toward warmer hues (reds and oranges) at sunrise and sunset when the sun's rays travel through more atmosphere.

These colour shifts, even without cloud cover, are captured by the RGB sensor as variations in red, green, and blue channel intensities. For example, the red channel mean increases during sunrise and sunset due to the reddish tint in the sky, while brightness and colour balance fluctuate throughout the day with solar elevation. This makes RGB-derived features highly sensitive to the irradiance cycle, providing strong predictive power despite the lack of clouds.

RGB red mean ( $r = 0.67$ ) and RGB brightness standard deviation ( $r = 0.78$ ) are among the strongest predictors.

- 
- **Humidity** exhibits a moderate negative correlation with irradiance. This aligns with typical atmospheric behavior, where higher humidity levels coincide with reduced incoming solar radiation.
  - In contrast, **NIR-derived features** such as `nir_brightness` and `nir_red_mean` exhibit lower correlations with irradiance ( $r \approx 0.40$ ), suggesting limited predictive power in the current dataset.

### 7.2.1 Impact of Atmospheric Conditions on NIR Effectiveness

During model development, a version of the irradiance predictor was trained using only features extracted from NIR images. This NIR-only model consistently underperformed compared to the RGB-only model. A likely explanation for this outcome is the lack of atmospheric variation during data collection: the dataset was gathered during an unusual period of consistently sunny, cloud-free conditions.

Under these clear-sky conditions, NIR images tend to be relatively uniform in intensity. NIR reflectance from the clear atmosphere is minimal and lacks the contrast and visual cues present in the RGB spectrum. In contrast, RGB images captured during the same conditions contain more dynamic information—such as brightness variations, color balance, and shadow contrasts, which are more directly tied to irradiance levels, as reflected in the heatmap correlations.

Additionally, the grayscale conversion of NIR images may have introduced redundancy between features like `nir_brightness` and `nir_red_mean`, reducing the diversity of input signals available to the model. The use of saturation could have also impacted the images gathered, older code used `—saturation 0.0` but this may have suppressed important image qualities like colour causing the redundancy of the images.[\[A.2.14\]](#).

It is anticipated that NIR features would demonstrate stronger predictive value under different atmospheric conditions, such as cloudy or overcast skies, where NIR reflectance interacts more significantly with cloud layers and airborne particles. Future deployments across varied weather patterns would help confirm this hypothesis and potentially enhance the model's robustness across diverse conditions.

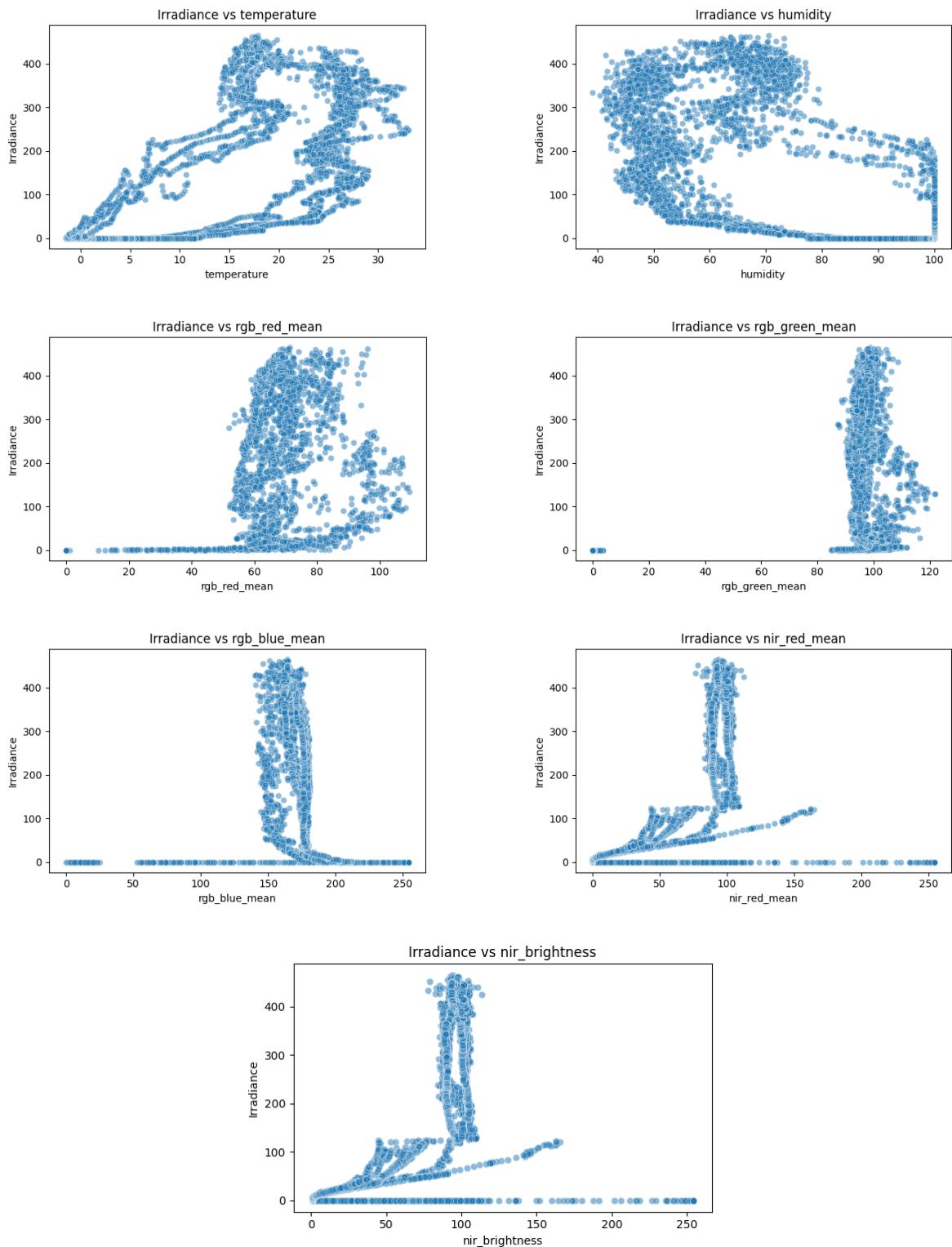


Figure 7.3: Scatter plots showing relationships between solar irradiance and various environmental and image-derived features

The scatter plots in Figure 7.3 illustrate the relationships between solar irradiance and each of the collected features. Notable observations include:

- **Temperature** displays a positive relationship with irradiance, consistent with the expectation that clear, sunny days yield both higher irradiance and temperatures.

- 
- **Humidity** is inversely correlated, with higher humidity typically aligning with lower irradiance, likely due to cloud formation or atmospheric moisture.
  - **RGB image features**, especially the red channel mean, show strong clustering with irradiance, validating their role as proxies for ambient light.
  - **NIR features** provide useful variance under low-light conditions, complementing the RGB-derived data.

These observations highlight the usefulness of both sensor and image-derived features contributing to meaningful information for irradiance modeling and serve as justification for their inclusion in the predictive LSTM model.

### 7.3 Image Histogram and Cloud Analysis

To analyze cloud characteristics for solar forecasting, both RGB and NIR images were captured and processed. Figure 7.4 shows a side-by-side comparison of an RGB image and its corresponding NIR image.



Figure 7.4: Left: RGB Image. Middle: NIR (All Colour). Right: NIR Image (Red Channel)

The pixel intensity distributions for the red, green, and blue channels of the RGB image, along with the NIR red channel, are shown in Figure 7.5. The x-axis represents pixel intensity values (0–255), and the y-axis represents the frequency of pixels with those intensities.

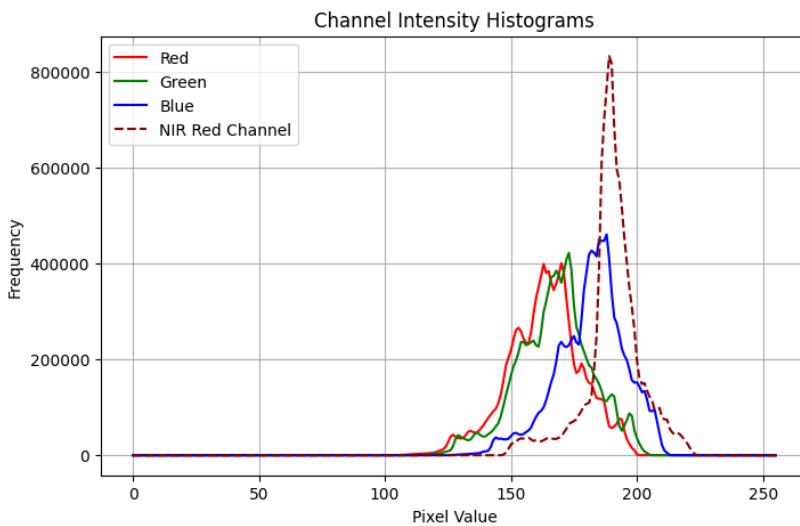


Figure 7.5: Channel intensity histograms for RGB and NIR images.

**Interpretation:** The NIR red channel histogram is narrower and peaks at a higher intensity (190–200) compared to the RGB channels, which peak around 160–180. This indicates that clouds reflect significantly more infrared radiation than visible light. The higher NIR intensity suggests thicker or higher-altitude clouds, which are more reflective in the infrared spectrum due to greater water or ice content. This characteristic makes NIR imaging particularly useful for identifying cloud coverage and type and this graph shows the drastic difference between the data that can be ascertained from NIR images compared to RGB.

### 7.3.1 Cloud Segmentation Using Otsu Thresholding

To quantify cloud coverage, Otsu's thresholding method was applied to the RGB red channel, RGB grayscale image, and NIR red channel. This technique automatically determines an optimal intensity threshold to separate foreground (clouds) from background (sky). The results are displayed in Figure 7.6.

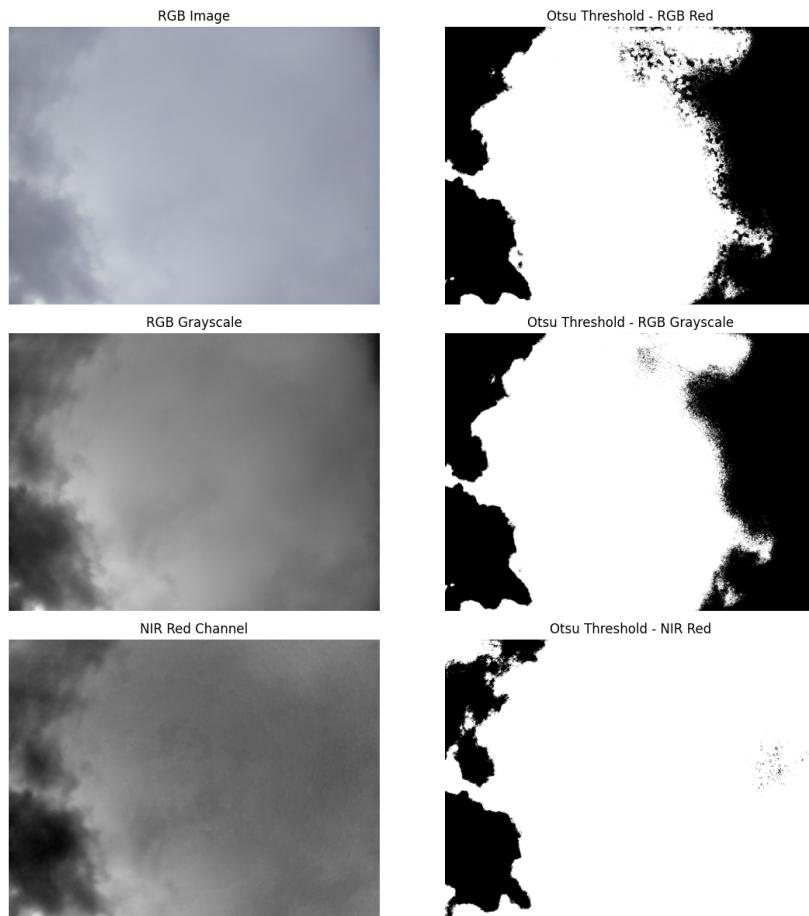


Figure 7.6: Cloud segmentation using Otsu thresholding for RGB and NIR images.

**Interpretation:** The NIR red channel provides a much clearer separation between clouds and sky compared to the RGB channels. This is because clouds reflect more NIR radiation, while the clear sky absorbs it, making segmentation more distinct. As a result, NIR-based thresholding offers a more reliable method for cloud detection, which is essential for accurate solar irradiance forecasting.

### 7.3.2 Comparative Pixel Intensity Distribution

A direct comparison of pixel intensity distributions between the RGB red channel and the NIR red channel is shown in Figure 7.7.

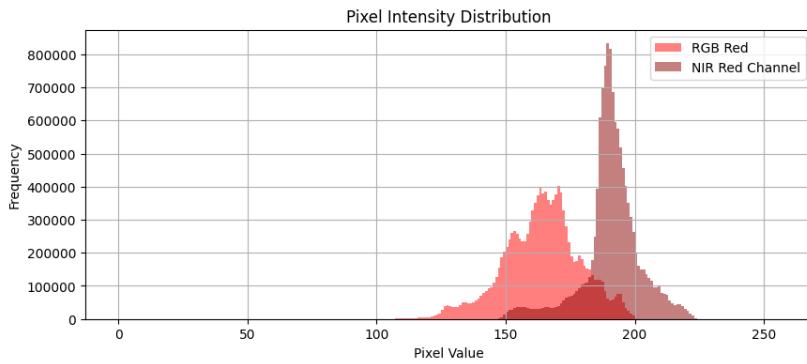


Figure 7.7: Pixel intensity distribution for RGB red channel and NIR red channel.

**Interpretation:** The NIR histogram demonstrates a sharper, higher-intensity peak, reinforcing that NIR captures more consistent cloud reflectance across the scene. In contrast, the RGB red channel shows a broader distribution due to variability in visible light scattering and absorption.

### 7.3.3 Applications for Solar Forecasting

The image analysis demonstrates that NIR and RGB imagery together offer a more comprehensive view for solar irradiance forecasting. NIR imaging excels at cloud detection and segmentation due to its strong response to cloud reflectance, ensuring reliable identification of cloud coverage and thickness. In contrast, RGB imagery captures visible atmospheric conditions, such as sky colour and thin clouds, which also affect irradiance but may not appear in NIR.

By combining NIR's consistency in cloud detection with RGB's sensitivity to atmospheric variability, a forecasting model could gain a broader understanding of sky conditions and provide a strong foundation for handling dynamic environments, improving the model's ability to adapt to changing light and cloud conditions in future deployments

## 7.4 LSTM Forecasting Model

The primary model evaluated for solar irradiance prediction incorporates both environmental sensor data (e.g., temperature, humidity) and features extracted from NIR and RGB images. This combination allows the model to better account for atmospheric conditions such as cloud coverage.

### 7.4.1 Performance Metrics

The model performance was assessed using the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE):

Metric	Value (Unscaled)
Mean Squared Error (MSE)	2.80
Root Mean Squared Error (RMSE)	1.67

Table 7.1: Performance metrics for the model with image features.

## 7.4.2 Graphical Analysis of Model Performance

### Actual vs. Predicted Values:

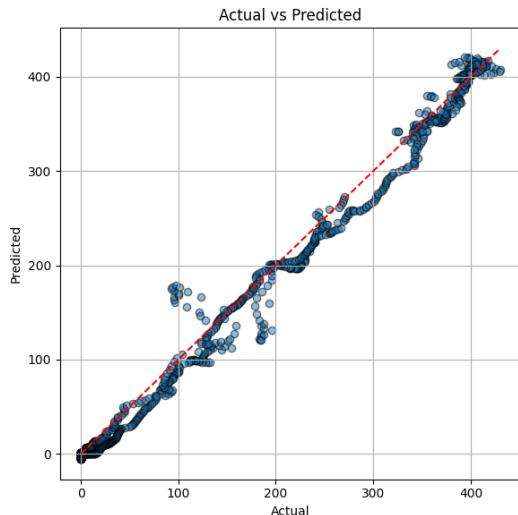


Figure 7.8: Model with image features: Actual vs. predicted values.

The scatter plot in Figure 7.8 shows a strong alignment between predicted and actual irradiance values. Most points cluster closely around the 1:1 red dashed line, indicating accurate predictions across the range of irradiance levels. Minor deviations occur at mid-range values, but overall, the model demonstrates reliable performance.

### Error Distribution:

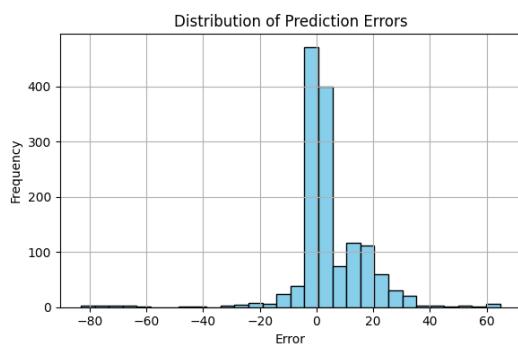


Figure 7.9: Model with image features: Distribution of prediction errors.

Figure 7.9 presents the distribution of prediction errors. The histogram is centered around zero, indicating minimal bias in the model's predictions. The majority of errors fall within a narrow range, demonstrating that the model produces small deviations between predicted and actual irradiance. The relatively symmetrical distribution suggests consistent performance without significant over- or underestimation.

### Predicted vs. Actual Over Time:

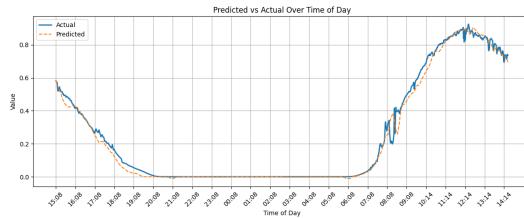


Figure 7.10: Model with image features: Predicted vs. actual irradiance over time.

Figure 7.10 illustrates the model's ability to track irradiance over time. The model captures the general trend and magnitude of solar irradiance throughout the day, including key transitions such as sunrise, peak irradiance, and sunset. Although some slight lag occurs during rapid changes in the morning, the model accurately follows the overall pattern.

### Residuals Over Time:

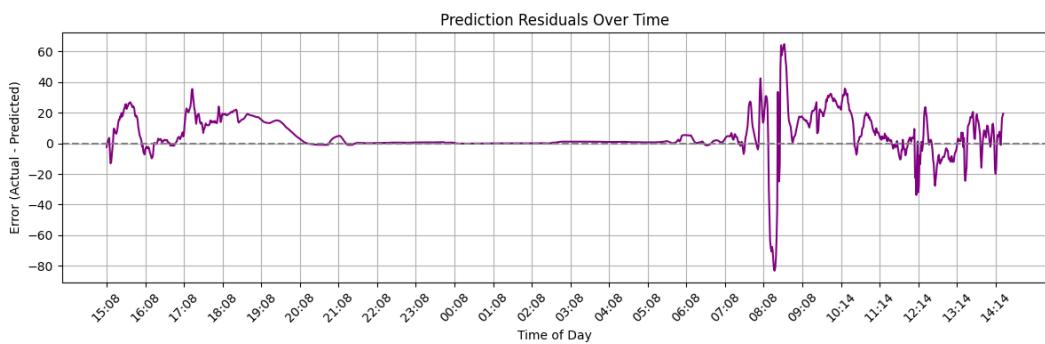


Figure 7.11: Model with image features: Prediction residuals over time.

The residuals over time in Figure 7.11 provide insight into how prediction errors fluctuate during the day. Residuals remain close to zero for most periods, with minor deviations occurring during sunrise and mid-morning. These deviations could correspond to the accumulation of dew or the rapid change in irradiation at dawn and dusk. Overall, the model maintains stable residuals.

### 7.4.3 Comparison to Model Without Image Features

To show the benefit of including image-derived features, a second model was trained using only environmental sensor data. This model achieved a worse RMSE of 1.76 and MSE of 3.09, as shown in Table 7.2.

Model	MSE (Unscaled)	RMSE (Unscaled)
With Image Features	2.80	1.67
Without Image Features	3.09	1.76

Table 7.2: Comparison of models with and without image features.

**Conclusion:** The inclusion of image-derived features from NIR and RGB images improves the model's ability to predict solar irradiance. This enhancement is reflected in lower error metrics, tighter error distributions, and improved tracking of irradiance changes throughout the day, particularly during periods of cloud-induced variability.

---

# Chapter 8: Discussion, Future Work and Conclusion

---

## 8.1 Discussion

The image analysis conducted throughout this project demonstrated the critical role that Near-Infrared (NIR) and RGB play in understanding atmospheric conditions relevant to solar forecasting. The analysis of pixel intensity distributions and cloud segmentation techniques highlighted that NIR imagery can provide sharper, more stable indicators of cloud presence, thanks to its sensitivity to infrared reflectance. Conversely, RGB imagery captured subtler atmospheric variations, such as sky colour offering complementary insights that NIR alone could not provide. Together, these imaging modalities delivered a comprehensive view of sky conditions, extending beyond the capabilities of traditional environmental sensors.

By integrating image-derived features like brightness, contrast, and colour channel intensities with environmental sensor data (e.g., temperature, humidity, and light intensity), the solar irradiance forecasting model was able to more effectively interpret dynamic sky conditions. This enriched dataset led to a clear improvement in predictive performance, with the model achieving lower RMSE values compared to versions using only sensor data whose accuracy was already good showing great potential for multi-modal systems.

The hardware enclosure functioned effectively in protecting the imaging devices and sensors from environmental exposure, ensuring reliable data collection throughout the deployment period. However, occasional lens fogging and dew accumulation, particularly in the early morning, were observed and impacted image clarity. These challenges highlight the need for further hardware refinements, such as incorporating a heating coil or hydrophobic lens coatings to mitigate moisture-related issues.

Despite these positive outcomes, the limited duration of data collection and geographic scope of the deployment restricted the model's exposure to a wider range of seasonal and weather conditions. This limits the generalizability of the model across different climates and timeframes.

Future work will focus on addressing these limitations, extending the deployment period, improving hardware resilience, and exploring autonomous power solutions and remote data transfer to enable longer-term, off-grid operation.

## 8.2 Future Works:

Several improvements are proposed to enhance the system's reliability and scalability:

- **Extended Field Deployment:** Deploy the system over longer durations to collect more diverse seasonal data, improving the model's generalization across varying weather conditions.
- **Dew Prevention:** Integrate a small heating coil within the enclosure to prevent dew accu-

---

mulation on camera lenses, ensuring consistent image quality in humid or cold environments.

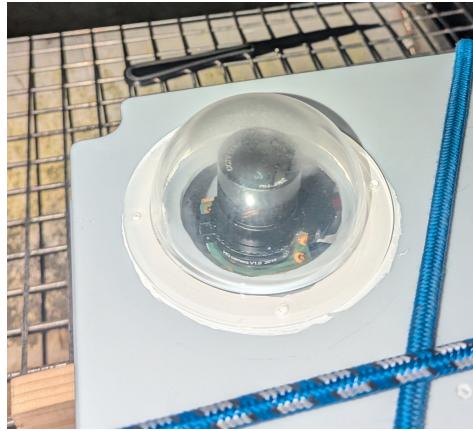


Figure 8.1: Image showing dew build up on lens

- **Solar-Powered Operation:** Explore powering the system using solar panels and battery storage to enable off-grid, autonomous operation, reducing the reliance on external power sources.
- **Remote Data Transfer:** Implement wireless data transfer protocols to enable remote data collection and system monitoring without manual intervention.
- **Consolidated Hardware Platform:** Redesign the system architecture to operate both the RGB and NIR cameras on a single Raspberry Pi device, reducing hardware complexity, power consumption, and system footprint.
- **Synchronized Dual-Camera Capture:** Implement simultaneous image acquisition from two dedicated Waveshare NIR-cut cameras—one with the IR-cut filter permanently engaged for RGB capture, and the other with the filter permanently removed for continuous NIR imaging. This configuration would mean less aligning would need to be done as both cameras would use the same field of view.
- **Advanced Image Processing Pipeline:** Integrate an enhanced image processing pipeline for both NIR and RGB images. This will include operations such as histogram equalization for contrast enhancement, noise reduction using Gaussian blurring, cloud segmentation through thresholding or edge detection, and dynamic white balance adjustment. These techniques aim to improve the quality of the extracted features, ensuring more reliable data for model training.

These enhancements aim to develop a fully autonomous, resilient system capable of long-term deployment in various environments for reliable solar forecasting.

## 8.3 Conclusion

This project successfully developed and implemented an integrated solar irradiance forecasting system, combining both hardware and software components. The system included a custom-designed enclosure housing environmental sensors and imaging devices capable of capturing both Near-Infrared (NIR) and RGB images alongside atmospheric measurements such as temperature, humidity, and light intensity.

---

The forecasting model demonstrated improved accuracy through the inclusion of image-derived features, such as brightness, contrast, and individual colour channel intensities, which provided valuable insights into cloud coverage and atmospheric conditions. This enhancement underscores the effectiveness of combining traditional environmental sensing with computer vision techniques for weather-related forecasting.

Overall, the project highlights the potential and viability of low-cost, scalable multi-modal solutions in solar forecasting, contributing to more efficient renewable energy management and supporting the broader transition toward sustainable energy systems.

---

## Chapter 9: Acknowledgements

---

I would like to express my gratitude to my supervisor, Dr Soumyabrata, for the opportunity to work with him in the creation of this project.

I am also deeply thankful to Ifran Rahman Nijhum whose assistance and advice greatly contributed to the development of both the hardware and software aspects of this project. His experience, knowledge and availability throughout were invaluable to me and I am extremely grateful.

I would like to thank my family for their support. A special thanks to my friend Bert and to my housemates in Beech Hill for providing a supportive and enjoyable environment throughout the course of this project.

This project would not have been possible without the combined support of all these individuals.

---

# Bibliography

---

1. Kishore, T. S., Kumar, P. U. & Ippili, V. Review of global sustainable solar energy policies: Significance and impact. *Innovation and Green Development* (2025).
2. Wu, C., Zhang, X.-P. & Sterling, M. Solar power generation intermittency and aggregation. *Scientific Reports* (2022).
3. Barhmi, K., Heynen, C., Golroodbari, S. & van Sark, W. A Review of Solar Forecasting Techniques and the Role of Artificial Intelligence. *Solar* (2024).
4. Sobri, S., Koohi-Kamali, S. & Abd Rahim, N. Solar photovoltaic generation forecasting methods: A review. *Energy Conversion and Management* (2018).
5. Inman, R. H., Pedro, H. T. & Coimbra, C. F. Solar forecasting methods for renewable energy integration. *Progress in Energy and Combustion Science* (2013).
6. Attya, M., Abo-Seida, O. M., Abdulkader, H. M. & Mohammed, A. M. Advanced solar radiation prediction using combined satellite imagery and tabular data processing. *Scientific Reports* (2025).
7. Dev, S., Savoy, F. M., Lee, Y. H. & Winkler, S. Estimating solar irradiance using sky imagers. *Atmospheric Measurement Techniques* **12** (2019).
8. Paletta, Q. et al. Advances in solar forecasting: Computer vision with deep learning. *Advances in Applied Energy* (2023).
9. Blazakis, K. et al. Towards Automated Model Selection for Wind Speed and Solar Irradiance Forecasting. *Sensors* (2024).
10. Lin, F., Zhang, Y. & Wang, J. Recent advances in intra-hour solar forecasting: A review of ground-based sky image methods. *International Journal of Forecasting* **39** (2023).
11. Wang, D., Tan, X., Wu, C., Zhang, H. & Song, Y. *Cloud detection using sky images and object detection algorithms* in 2023 International Conference on Renewable Energy and Power Engineering (REPE) (2023).
12. Wang, F. et al. A minutely solar irradiance forecasting method based on real-time sky image-irradiance mapping model. *Energy Conversion and Management* (2020).
13. Terrén-Serrano, G. & Martínez-Ramón, M. Deep learning for intra-hour solar forecasting with fusion of features extracted from infrared sky images. *Information Fusion* (2023).
14. Nie, Y. et al. Open-source sky image datasets for solar forecasting with deep learning: A comprehensive survey. *Renewable and Sustainable Energy Reviews* **189** (2024).
15. Urquhart, B. et al. Development of a sky imaging system for short-term solar power forecasting. *Atmospheric Measurement Techniques* **8** (2015).
16. Park, S. et al. Prediction of Solar Irradiance and Photovoltaic Solar Energy Product Based on Cloud Coverage Estimation Using Machine Learning Methods. *Atmosphere* **12** (2021).
17. Dev, S., Savoy, F. M., Lee, Y. H. & Winkler, S. WAHRSIS: A low-cost, high-resolution whole sky imager with near-infrared capabilities in *Proceedings of IS&T/SPIE Infrared Imaging Systems* (SPIE, 2017).
18. Bamisile, O., Acen, C., Cai, D., Huang, Q. & Stoffell, I. The environmental factors affecting solar photovoltaic output. *Renewable and Sustainable Energy Reviews* (2025).
19. Myers, D. R., Stoffel, T. L., Reda, I., Wilcox, S. M. & Andreas, A. M. Recent Progress in Reducing the Uncertainty in and Improving Pyranometer Calibrations. *Journal of Solar Energy Engineering* **124** (2002).

- 
20. Blazakis, K., Katsigiannis, Y. & Stavrakakis, G. One-Day-Ahead Solar Irradiation and Wind-speed Forecasting with Multi-Channel Convolutional Neural Networks. *Energies* (2022).
21. Al Riza, D. F., Gilani, S. I. u. H. & Aris, M. S. Hourly solar radiation estimation using ambient temperature and relative humidity data. *International Journal of Environmental Science and Development* (2011).
22. Avlani, S., Seo, D.-H., Chatterjee, B. & Sen, S. EICO: Energy-Harvesting Long-Range Environmental Sensor Nodes With Energy-Information Dynamic Co-Optimization. *IEEE Internet of Things Journal* (2022).
23. Jain, M. et al. LAMSkyCam: A low-cost and miniature ground-based sky camera. *HardwareX* **12** (2022).
24. Atique, S. et al. Forecasting of total daily solar energy generation using ARIMA: A case study (2019).
25. Ramadhan, J. M., Ahmed, D. M. & Hassan, M. M. Comparison of physical and machine learning models for solar irradiance and photovoltaic power forecasting. *International Journal of Electrical and Computer Engineering* (2021).
26. Ahmed, D. M., Hassan, M. M. & Mstafa, R. J. A Review on Deep Sequential Models for Forecasting Time Series Data. *Applied Computational Intelligence and Soft Computing* (2022).
27. Yu, R., Lin, C. & Yang, Q. An LSTM short-term solar irradiance forecasting under complicated weather conditions. *Energy* (2019).
28. Fouzan, M. *Understanding LSTM, GRU, and RNN Architectures* 2023.
29. Srivastava, S. & Lessmann, S. A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. *Solar Energy* (2018).
30. Michael, A. D., Selvakumar, S., Rani, S. & Kirthiga, M. Short-term solar irradiance forecasting using stacked LSTM networks and multivariate data. *Renewable Energy* (2022).
31. Zang, H. et al. Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations. *Renewable Energy*.
32. Ghimire, S. et al. Stacked LSTM Sequence-to-Sequence Autoencoder with Feature Selection for Daily Solar Radiation Prediction: A Review and New Modeling Results. *Energies* (2022).
33. Wang, F. et al. A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework. *Energy Conversion and Management* (2020).
34. Wang, J., Zhang, N., Wang, Y., Zhao, J. & Wang, C. Day-ahead photovoltaic power production forecasting methodology using weather sequence similarity. *Applied Energy* (2020).
35. AG, S. *SHT40: 4th Generation, High-Accuracy, Ultra-Low-Power, 16-bit Relative Humidity and Temperature Sensor Platform* [https://www.sensirion.com/file/datasheet/SHT4x\\_Datasheet.pdf](https://www.sensirion.com/file/datasheet/SHT4x_Datasheet.pdf). Datasheet, Version 1. Oct. 2020.
36. Semiconductor, R. *BH1750FVI: Digital 16-bit Serial Output Type Ambient Light Sensor IC for I2C Bus Interface* <https://www.mouser.com/datasheet/2/348/bh1750fvi-e-186247.pdf>. Version Rev. G. Datasheet. Nov. 2014.
37. Waveshare Electronics. *IMX219-160 IR-CUT Camera – Product Wiki* [https://www.waveshare.com/wiki/IMX219-160\\_IR-CUT\\_Camera](https://www.waveshare.com/wiki/IMX219-160_IR-CUT_Camera). Accessed: 19 April 2025. 2025.
38. Raspberry Pi Ltd. *Raspberry Pi 4 Model B Product Brief* <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>. Published by Raspberry Pi Ltd. Feb. 2025.
39. Tayg. *Tayg 605 Screw Top Plastic Junction Box, IP65 Rated, 160x120x80mm* Accessed: 2025-04-20. <https://www.amazon.co.uk/dp/B00U6BUE4U>.
40. Generic. *Clear Acrylic Dome for Security Camera, Weatherproof Housing Cover* <https://www.amazon.co.uk/dp/B08N4VV99F>. Accessed: 2025-04-20.

- 
41. Generic. *Clear Acrylic Dome Cover, 100mm Inner Diameter, Weatherproof, Transparent* [https://www.amazon.co.uk/dp/B08N4NCN25?ref=ppx\\_yo2ov\\_dt\\_b\\_fed\\_asin\\_title&th=1](https://www.amazon.co.uk/dp/B08N4NCN25?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1). Accessed: 2025-04-20.
  42. MELONBOY. *MELONBOY 42 Pcs Cable Glands Set, M12 M16 M20 M25 Nylon Water-proof Glands, IP68 Rated* Accessed: 2025-04-20. [https://www.amazon.co.uk/gp/aw/d/B0CYLPM7T7/?\\_encoding=UTF8&pd\\_rd\\_plhdr=t&aaxitk=f20c05cfa6221f1182bceace885402b9&hsa\\_cr\\_id=0&qid=1745155337&sr=1-1-e0fa1fdd-d857-4087-adda-5bd576b25987&ref\\_=sbx\\_be\\_s\\_sparkle\\_lsi4d\\_asin\\_0\\_title&pd\\_rd\\_w=DGWlu&content\\_id=amzn1.sym.25f7c301-a223-4ff8-91c9-accfeab9fda8%3Aamzn1.sym.25f7c301-a223-4ff8-91c9-accfeab9fda8&pf\\_rd\\_p=25f7c301-a223-4ff8-91c9-accfeab9fda8&pf\\_rd\\_r=VHB5EM6E5PKBYX2QB110&pd\\_rd\\_wg=174B2&pd\\_rd\\_r=83c62d93-422b-4016-baf1-acffcd0db934&th=1](https://www.amazon.co.uk/gp/aw/d/B0CYLPM7T7/?_encoding=UTF8&pd_rd_plhdr=t&aaxitk=f20c05cfa6221f1182bceace885402b9&hsa_cr_id=0&qid=1745155337&sr=1-1-e0fa1fdd-d857-4087-adda-5bd576b25987&ref_=sbx_be_s_sparkle_lsi4d_asin_0_title&pd_rd_w=DGWlu&content_id=amzn1.sym.25f7c301-a223-4ff8-91c9-accfeab9fda8%3Aamzn1.sym.25f7c301-a223-4ff8-91c9-accfeab9fda8&pf_rd_p=25f7c301-a223-4ff8-91c9-accfeab9fda8&pf_rd_r=VHB5EM6E5PKBYX2QB110&pd_rd_wg=174B2&pd_rd_r=83c62d93-422b-4016-baf1-acffcd0db934&th=1).
  43. Michael, P. R., Johnston, D. E. & Moreno, W. A Conversion Guide: Solar Irradiance and Lux Illuminance. *Journal of Measurements in Engineering* (2020).

---

# Appendix A: Appendix

---

## A.1 Code Repository Access

All code used in this project is available at the following GitLab repository:

[https://csgitlab.ucd.ie/cscan\\_9/final-year-project.git](https://csgitlab.ucd.ie/cscan_9/final-year-project.git)

## A.2 Appendix B: Key Code Listings

### A.2.1 Coordinated Logging Script

The following loop synchronizes data capture and logging every minute on the RGB Pi.

```
1 def main():
2     try:
3         while True:
4             now = datetime.now()
5             if now.second == 0:
6                 try:
7                     timestamp = now.strftime("%Y-%m-%d_%H-%M-%S")
8                     temp, humidity = read_temp_and_humidity()
9                     image_filename = f"{timestamp}_rgb.jpg"
10                    capture_image(image_filename, photos_dir)
11                    write_to_csv(timestamp, temp, humidity, f"/
12                                photos_rgb/{image_filename}")
13                    time.sleep(1)
14                except Exception as e:
15                    print(f"Error during capture: {e}")
16                else:
17                    time.sleep(0.5)
18            except KeyboardInterrupt:
19                print("Interrupted by user")
```

Listing A.1: Main Logging Loop

### A.2.2 Adaptive Exposure Control in NIR Camera

This logic switches between night, sunny, and normal modes based on lux levels.

---

```

1 def capture_normal_image(image_filename, photos_dir):
2     filename = os.path.join(photos_dir, image_filename)
3     os.system(f"libcamera-still-o{filename}--awb-custom--awbgains"
4               "1.2,1.2")
5
6 def capture_sunny_image(image_filename, photos_dir):
7     filename = os.path.join(photos_dir, image_filename)
8     os.system(f"libcamera-still--shutter500--gain1--awb-custom--"
9               "awbgains1.2,1.2-o{filename}")
10
11 def capture_night_image(image_filename, photos_dir):
12     filename = os.path.join(photos_dir, image_filename)
13     os.system(f"libcamera-still--shutter600000--gain32-o{"
14               "filename}")

```

Listing A.2: Adaptive Exposure Functions

### A.2.3 RGB Camera Capture Function

This function captures RGB images using libcamera-jpeg.

```

1 def capture_image(image_filename, photos_dir):
2     filename = os.path.join(photos_dir, image_filename)
3     os.system(f"libcamera-jpeg-o{filename}")

```

Listing A.3: RGB Image Capture Function

### A.2.4 BH1750 Sensor Mtreg Setter

Function to set Mtreg automatically based on the lux

```

1 def auto_select_mtreg():
2     trial_mtreg = 69
3     set_sensitivity(trial_mtreg)
4     trial_lux = read_lux(mtreg=trial_mtreg)
5     if trial_lux < 5:
6         mtreg = 150
7     elif trial_lux < 500:
8         mtreg = 120
9     elif trial_lux < 20000:
10        mtreg = 69
11    else:
12        mtreg = 32
13    return mtreg

```

Listing A.4: Auto MTReg Selection

## A.2.5 BH1750 Sensor Read Lux

```
1 def read_lux(mtreg):
2     bus.write_byte(BH1750_ADDR, ONE_TIME_HIGH_RES_MODE)
3     time.sleep(0.180 * (mtreg / 69.0))
4     data = bus.read_i2c_block_data(BH1750_ADDR, ONE_TIME_HIGH_RES_MODE
5         , 2)
6     count = (data[0] << 8) | data[1]
7     ratio = 1 / (1.2 * (mtreg / 69.0))
8     lux = count * ratio
9     return lux
```

Listing A.5: Lux Reading

## A.2.6 Light Sensor Script Granularity

These lines show the granular control of the bh1750 sensor.

```
1 # I2C setup for BH1750
2 bus = smbus.SMBus(1)
3 BH1750_ADDR = 0x23
4
5 # BH1750 Commands
6 POWER_ON = 0x01
7 RESET = 0x07
8 ONE_TIME_HIGH_RES_MODE = 0x20
```

Listing A.6: BH1750 Set-Up

```
1 def set_sensitivity(mtreg):
2     bus.write_byte(BH1750_ADDR, POWER_ON) # Power on the sensor
3     bus.write_byte(BH1750_ADDR, 0x40 | (mtreg >> 5)) # Set MTreg lower
4         3 bits
5     bus.write_byte(BH1750_ADDR, 0x60 | (mtreg & 0x1F)) # Set MTreg
6         upper bits does XOR with 0x1F = 00011111
7     bus.write_byte(BH1750_ADDR, RESET) # Reset the sensor
```

Listing A.7: Set sensitivity function showing necessity of bus writing

```
1 def read_lux(mtreg):
2     bus.write_byte(BH1750_ADDR, ONE_TIME_HIGH_RES_MODE)
3     time.sleep(0.180 * (mtreg / 69.0)) # Wait for the sensor to take a
4         reading
5     data = bus.read_i2c_block_data(BH1750_ADDR, ONE_TIME_HIGH_RES_MODE
6         , 2)
7     ...
8
```

Listing A.8: Further lines exercising low-level functionality

## A.2.7 Temperature and Humidity Script

---

```

1 import board
2 import adafruit_sht4x
3
4 # Initialize I2C and sensor
5 i2c = board.I2C()
6 sensor = adafruit_sht4x.SHT4x(i2c)
7
8 def read_temp_and_humidity():
9     temperature, humidity = sensor.measurements
10    return temperature, humidity

```

Listing A.9: Temperature and Humidity Script

## A.2.8 Solar Irradiance Calculation

```

1 def computeSolarIrradiance(lux):
2     return (1/120) * lux

```

Listing A.10: Solar Irradiance calculation

## A.2.9 Starter Script

```

1#!/bin/bash
2
3echo "Starting data logger..."
4nohup python3 "coordinated_script_cptsc.py" > "output_logs.txt" 2>&1 &
5
6echo "Script started in background. Logging to: output_logs.txt"
7echo "Use 'tail -f output_logs.txt' to view logs."

```

Listing A.11: Starter Script

## A.2.10 Scaling Code

```

1 scaler = MinMaxScaler()
2 train = pd.DataFrame(scaler.fit_transform(train), columns=train.columns)
3 test = pd.DataFrame(scaler.transform(test), columns=test.columns)

```

Listing A.12: Code used for scaling

## A.2.11 Model Creation

```

1 model = Sequential()
2 model.add(InputLayer((MULTIV_TRAINING_WINDOW, NUMBER_OF_FEATURES)))

```

---

```

3 model.add(LSTM(model_lstm_param))
4 model.add(Dropout(0.5))
5 model.add(Dense(model_dense_params, 'relu'))
6 model.add(Dense(1, 'linear'))

```

Listing A.13: Model Creation

### A.2.12 Model Compilation showing Adam usage

```

1 model.compile(
2     loss=MeanSquaredError(),
3     optimizer=Adam(learning_rate=LEARNING_RATE),
4     metrics=[MeanSquaredError()]
5 )

```

Listing A.14: Model Compilation

### A.2.13 Image Feature Extraction

```

1 for _, row in tqdm(df.iterrows(), total=len(df), desc="Processing images"):
2     # load nir images
3     nir_color = cv2.imread(row['nir_image_path'])
4     if nir_color is not None:
5         nir_gray = cv2.cvtColor(nir_color, cv2.COLOR_BGR2GRAY)
6         nir_color_rgb = cv2.cvtColor(nir_color, cv2.COLOR_BGR2RGB)
7         nir_r, _, _ = cv2.split(nir_color_rgb)
8
9         # NIR features
10        nir_brightness.append(np.mean(nir_gray))
11        nir_brightness_std.append(np.std(nir_gray))
12        nir_red_mean.append(np.mean(nir_r))
13    else:
14        print(f"Image not found: {row['nir_image_path']}")
15        nir_brightness.append(np.nan)
16        nir_brightness_std.append(np.nan)
17        nir_red_mean.append(np.nan)
18
19    # also handle the rgb images
20    rgb_img = cv2.imread(row['rgb_image_path'])
21    if rgb_img is not None:
22        rgb_img = cv2.cvtColor(rgb_img, cv2.COLOR_BGR2RGB)
23        rgb_gray = cv2.cvtColor(rgb_img, cv2.COLOR_RGB2GRAY)
24        r, g, b = cv2.split(rgb_img)
25
26        # RGB features
27        rgb_brightness.append(np.mean(rgb_gray))
28        rgb_brightness_std.append(np.std(rgb_gray))

```

---

```

29     rgb_red_mean.append(np.mean(r))
30     rgb_green_mean.append(np.mean(g))
31     rgb_blue_mean.append(np.mean(b))
32
33 else:
34     print(f"Image not found: {row['rgb_image_path']}")
35     rgb_brightness.append(np.nan)
36     rgb_brightness_std.append(np.nan)
37     rgb_red_mean.append(np.nan)
38     rgb_green_mean.append(np.nan)
39     rgb_blue_mean.append(np.nan)

```

Listing A.15: Image Feature Extraction

### A.2.14 Change in NIR capture code

This section shows the change in NIR code capture

```

1 libcamera-still --shutter 500 --gain 1 --saturation 0.0 -o {filename}

```

Listing A.16: Old libcamera commands

```

1 libcamera-still --shutter 500 --gain 1 --awb custom --awbgains 1.2,1.2
      -o {filename}

```

Listing A.17: New libcamera commands

## A.3 SolidWorks CAD Drawings

The complete set of SolidWorks parts, assembly, and technical drawings used in this project are available at the GitLab repository:

[https://csgitlab.ucd.ie/cscan\\_9/final-year-project/-/tree/main/CAD\\_Parts\\_and\\_Drawings](https://csgitlab.ucd.ie/cscan_9/final-year-project/-/tree/main/CAD_Parts_and_Drawings)

These include:

- Junction Box Part.
- 2 Domes.
- Cable gland parts
- Screw parts
- Assembly of the box which drawings are done from
- Dimensioned technical drawings in both SolidWorks and PDF formats.

---

## A.4 Sample Data

### A.4.1 Data From NIR System

	A	B	C	D	E	F	G
388	2025-04-06_06-08-00	0.4166667	0.003472	/photos_nir/2025-04-06_06-08-00_nir.jpg			
389	2025-04-06_06-09-00	0.8333333	0.006944	/photos_nir/2025-04-06_06-09-00_nir.jpg			
390	2025-04-06_06-10-00	0.8333333	0.006944	/photos_nir/2025-04-06_06-10-00_nir.jpg			
391	2025-04-06_06-11-00	0.8333333	0.006944	/photos_nir/2025-04-06_06-11-00_nir.jpg			
392	2025-04-06_06-12-00	1.25	0.010417	/photos_nir/2025-04-06_06-12-00_nir.jpg			
393	2025-04-06_06-13-00	1.6666667	0.013889	/photos_nir/2025-04-06_06-13-00_nir.jpg			
394	2025-04-06_06-14-00	1.6666667	0.013889	/photos_nir/2025-04-06_06-14-00_nir.jpg			
395	2025-04-06_06-15-00	2.5	0.020833	/photos_nir/2025-04-06_06-15-00_nir.jpg			
396	2025-04-06_06-16-00	2.9166667	0.024306	/photos_nir/2025-04-06_06-16-00_nir.jpg			
397	2025-04-06_06-17-00	3.3333333	0.027778	/photos_nir/2025-04-06_06-17-00_nir.jpg			
398	2025-04-06_06-18-00	4.1666667	0.034722	/photos_nir/2025-04-06_06-18-00_nir.jpg			
399	2025-04-06_06-19-00	5	0.041667	/photos_nir/2025-04-06_06-19-00_nir.jpg			
400	2025-04-06_06-20-00	5.8333333	0.048611	/photos_nir/2025-04-06_06-20-00_nir.jpg			
401	2025-04-06_06-21-00	6.6666667	0.055556	/photos_nir/2025-04-06_06-21-00_nir.jpg			
402	2025-04-06_06-22-00	8.3333333	0.069444	/photos_nir/2025-04-06_06-22-00_nir.jpg			
403	2025-04-06_06-23-00	10	0.083333	/photos_nir/2025-04-06_06-23-00_nir.jpg			
404	2025-04-06_06-24-00	11.666667	0.097222	/photos_nir/2025-04-06_06-24-00_nir.jpg			
405	2025-04-06_06-25-00	14.166667	0.118056	/photos_nir/2025-04-06_06-25-00_nir.jpg			
406	2025-04-06_06-26-00	16.666667	0.138889	/photos_nir/2025-04-06_06-26-00_nir.jpg			
407	2025-04-06_06-27-00	20	0.1666667	/photos_nir/2025-04-06_06-27-00_nir.jpg			
408	2025-04-06_06-28-00	23.333333	0.194444	/photos_nir/2025-04-06_06-28-00_nir.jpg			
409	2025-04-06_06-29-00	27.5	0.229167	/photos_nir/2025-04-06_06-29-00_nir.jpg			
410	2025-04-06_06-30-00	32.5	0.270833	/photos_nir/2025-04-06_06-30-00_nir.jpg			
411	2025-04-06_06-31-00	37.916667	0.315972	/photos_nir/2025-04-06_06-31-00_nir.jpg			



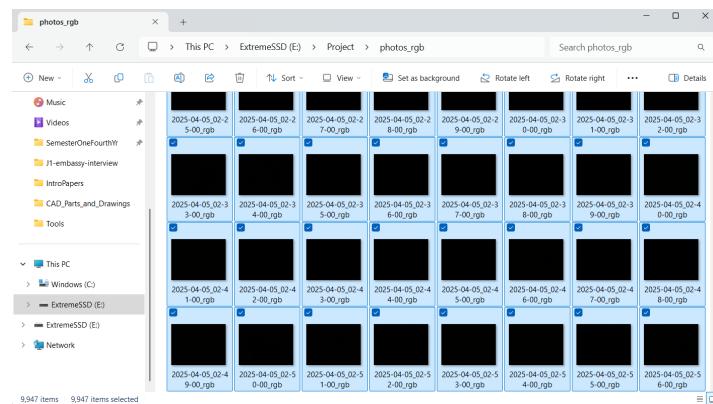


Figure A.1: Total RGB images captured by the system.

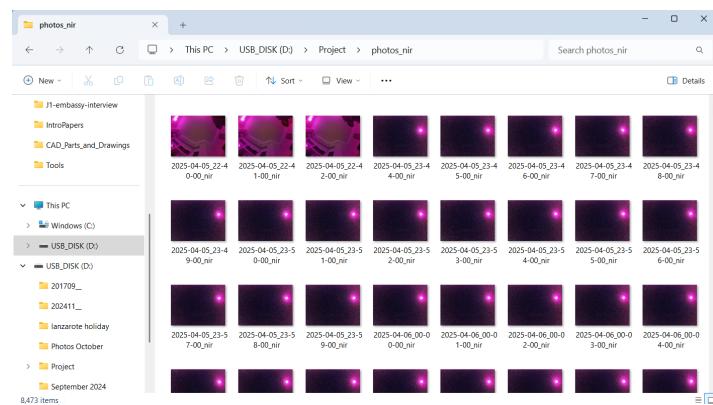


Figure A.2: Total NIR images captured by the system.

A	B	C	D	E	F	G	H
8460	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-17-00_nir.jpg			
8461	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-18-00_nir.jpg			
8462	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-19-00_nir.jpg			
8463	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-20-00_nir.jpg			
8464	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-21-00_nir.jpg			
8465	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-22-00_nir.jpg			
8466	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-23-00_nir.jpg			
8467	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-24-00_nir.jpg			
8468	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-25-00_nir.jpg			
8469	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-26-00_nir.jpg			
8470	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-27-00_nir.jpg			
8471	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-28-00_nir.jpg			
8472	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-29-00_nir.jpg			
8473	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-30-00_nir.jpg			
8474	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-31-00_nir.jpg			
8475	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-32-00_nir.jpg			
8476	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-33-00_nir.jpg			
8477	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-34-00_nir.jpg			
8478	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-35-00_nir.jpg			
8479	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-36-00_nir.jpg			
8480	2025-04-1	0.3833333	0.003194	/photos_nir/2025-04-15_21-37-00_nir.jpg			
8481							

Figure A.3: Number of rows in the RGB data file (image paths and corresponding sensor readings).

A	B	C	D	E	F	G	H	I
9942	s			/photos_rgb/2025-04-15_21-27-00_rgb.jpg				
9943	2025-04-1	11.851300	63.130998	/photos_rgb/2025-04-15_21-28-00_rgb.jpg				
9944	2025-04-1	11.845960	61.137788	/photos_rgb/2025-04-15_21-29-00_rgb.jpg				
9945	2025-04-1	11.704432	61.271305	/photos_rgb/2025-04-15_21-30-00_rgb.jpg				
9946	2025-04-1	11.659037	63.344625	/photos_rgb/2025-04-15_21-31-00_rgb.jpg				
9947	2025-04-1	11.605630	66.243839	/photos_rgb/2025-04-15_21-32-00_rgb.jpg				
9948	2025-04-1	11.685740	65.412222	/photos_rgb/2025-04-15_21-33-00_rgb.jpg				
9949	2025-04-1	11.797894	66.713054	/photos_rgb/2025-04-15_21-34-00_rgb.jpg				
9950	2025-04-1	11.899366	61.864499	/photos_rgb/2025-04-15_21-35-00_rgb.jpg				
9951	2025-04-1	11.875333	61.500190	/photos_rgb/2025-04-15_21-36-00_rgb.jpg				
9952	2025-04-1	11.709773	59.57183	/photos_rgb/2025-04-15_21-37-00_rgb.jpg				
9953	2025-04-1	11.568245	61.240787	/photos_rgb/2025-04-15_21-38-00_rgb.jpg				
9954								
9955								
9956								
9957								
9958								
9959								
9960								
9961								
9962								

Figure A.4: Number of rows in the NIR data file (image paths and corresponding sensor readings).

---

## A.5 Tools Used

This is a list of tools that were used during the hardware construction of the project.

### A.5.1 Soldering Kit



### A.5.2 Wire Crimping and Dupont Connector Kit





#### A.5.3 Electric Drill



#### A.5.4 Hot Glue Gun



---

### A.5.5 Screw Driver Set

