Computer Vision on Edge Devices for the Short Term Prediction of Cloud Cover

Jordan Perr-Sauer, Kristin Munch, and Robert White

National Renewable Energy Laboratory

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-2C00-77999
December 2024

Computer Vision on Edge Devices for the Short Term Prediction of Cloud Cover

Jordan Perr-Sauer, Kristin Munch, and Robert White

National Renewable Energy Laboratory

Suggested Citation
Perr-Sauer, Jordan, Kristin Munch, and Robert White. 2024. *Computer Vision on Edge Devices for the Short Term Prediction of Cloud Cover*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-2C00-77999. https://www.nrel.gov/docs/fy25osti/77999.pdf.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-2C00-77999
December 2024

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

Acknowledgments

The project was funded under a SCEA Seed LDRD Grant.

Many people at NREL provided assistance during this project. I will specifically call out: David Rager for helping with Amazon Web Services. Steven Robbins, Keith Layton, and Caleb Phillips for help designing the enclosure. Afshin Adreas for being my point of contact at the Mesa Top facility, and for helping fine tune the sky imager enclosure. Karen Kuoppala, Jake Gedvilas, and Mark Kutchenreiter at the SRRL helped perform the facilities review. Jose Pizano and Michael McCausey at the machine shop for building such a beautiful enclosure. Shane Parsely, Steve Menken, Kieran Rhysling, Ann Craig, Connie Overly and Ken Hyatt for performing a cyber-security review. Nick Wunder for the cellular modem. Lindsay Proell and Mike LeComte for help setting up the cellular modem with wireless service. Heath Garrison and Phil Clark for providing an environmental health and safety review of the sky imager. Tami Sandberg at the NREL library for her assistance in the literature review.

Executive Summary

Edge Computing and Internet of Things (IoT) are vital pieces of today's technological landscape. Here, we describe CloudCV, a low-cost IoT sky imager programmed and managed using Amazon Web Services (AWS) GreenGrass. The sky imager was deployed for three months in 2019 at the NREL Mesa Top facility, where it collected a 10-second resolution sky image and irradiance dataset. We demonstrate remote reprogramming of this device to load software that predicts sun shading events through an optical flow-based linear advection method, which is a baseline algorithm that can be used to benchmark algorithmic improvements in future work. Some future directions for research in cloud cover prediction using sky imaging are enumerated.

The outcomes of this work are:

- The CloudCV sky imager was designed and built at the National Renewable Energy Laboratory (NREL). It was deployed at NREL's Solar Radiation Research Laboratory (SRRL) Mesa Top facility for three months in 2019.

- A demonstration of programming and managing the device using the AWS GreenGrass IoT platform. This technology could be useful in future field experiments and facilitate scientific data collection.

- A dataset of sky images and irradiance measurements was collected at a 10-second time resolution and 1920x1080 pixel image resolution. Ten seconds is a significantly shorter time interval than other sky imager datasets. A subset of this dataset will be published and made available through NREL's Data Catalog.

- A preliminary analysis is described using this dataset to perform irradiance forecasting using optical flow and linear advection. This method was able to achieve a Critical Success Index (CSI) score of 0.6 for sun shading events at the 1-minute time horizon for a hand-picked, ideal day, after manual tuning.

- The software, including sky imager code and the preliminary analysis code, will be published and made available through Github.

Table of Contents

Introduction

This technical report summarizes the work performed in FY2019 for the SCEA Seed LDRD "Computer Vision on Edge Devices for the Short-Term Prediction of Cloud Cover." The report is divided into four sections. Section 1 provides a high-level overview and background for the work. Section 2 provides a comparison of the various Internet of Things (IoT) platforms and devices which were considered for this project. Section 3 describes the process of configuring and evaluating an in-house sky imager on NREL's campus. Section 4 describes a use case of this device to rapidly deploy algorithms to the edge for short term prediction of cloud cover using computer vision.

1.1 Definitions of Edge Computing, The Cloud, and IoT

Edge computing, the cloud, and IoT are closely related terms that are easy to confuse and may not be well defined. Here, we define this terminology as it should be understood in the context of this document.

Edge computing is a paradigm in systems architecture to distribute data storage and computational burden to many small devices which are physically located close to the system which it is measuring or controlling. This is an inherently decentralized paradigm. Edge computing can describe computer systems that operate without internet connectivity, or which may face limited connectivity to a central server. These systems are expected to make decisions based on inputs gathered from local sensors, without the benefit of a centralized server or centralized decision-making capability.

The cloud is a paradigm of centralization of computing and data storage to large, often commercial, datacenters. In this computing paradigm, centralized computing power is made available in a flexible (on-demand) manner, such that the resources can be shared between multiple tenants and reconfigured for each job. The ability to shift compute infrastructure to commercial datacenters, which are highly reconfigurable, has revolutionized information technology (IT) throughout many areas of business, academia, and government. Communication with resources in the cloud requires an internet connection.

Internet of things (IoT) refers to the connection of edge computing devices to the internet, and more specifically to centralized infrastructure running on a cloud platform. The manifestation of this "IoT revolution" is seen in the marketplace as various "Smart" devices which the user can control and manage through an online portal or smartphone app. Examples of such devices include smart watches, smart speakers, smart thermostats, smart plugs, and smart lightbulbs.

Commercial providers of cloud services, such as Amazon, Microsoft, and Google, have all invested in developer-friendly frameworks to enable IoT use cases, with security, monitoring, and management capabilities built into the platforms. Additionally, these IoT frameworks can be used with a broad array of devices for many different edge computing use cases. IoT is an important concept to the industries in which NREL operates, with edge computing having many potential applications to research projects at NREL. Ideally, NREL can leverage the IoT frameworks from commercial cloud providers to accelerate this research. In the following section, potential use cases of IoT which are relevant to NREL's core missions will be described.

1.2 Uses for an Edge Computing Platform at NREL

In this Seed LDRD, we are evaluating a commercial, cloud-based IoT platform for its ability to manage an edge computing device for research purposes. To our knowledge the use of a commercial IoT solution like AWS GreenGrass to enable the programming of scientific devices is novel at NREL. Potential benefits of using Cloud based IoT solution include ease of programming, ability to reprogram the devices from a central workstation, and the abstraction of security and identity management concerns to the cloud platform provider.

1.2.1 Control Algorithm Validation and Deployment

One application of edge computing devices to NREL's mission statement is in the development of smart controllers for distributed energy resources on the energy grid. There may be an opportunity to demonstrate large scale deployment of physical controllers on a campus such as NREL's to quickly deploy and test advanced control algorithms on test systems.

1.2.2 Data Collection and Streaming

The development of new sensor technologies which generate data at high bitrates demands every larger network throughput. Pre-processing of sensor output, such as high-resolution imaging devices, is common use for edge computing devices in research. Edge computing devices allow researchers to deploy image processing and other machine learning algorithms to extract the most relevant features from high throughput sensors, transmitting only the relevant information back to the laboratory. If an internet connection is available, interesting data may be continuously streamed back to the laboratory instead of being manually retrieved from physical data loggers.

1.2.3 Smart Devices

The current niche for IoT in the marketplace is with so-called "smart devices," which we define as edge computing devices which have some form of centralized management often coming from infrastructure configured with a commercial cloud provider. Therefore, any projects that aim to create or improve upon smart devices could benefit from an IoT platform supported by NREL.

1.2.4 Hardware-In-The-Loop

Simulations of physical processes on HPC can provide insights into the dynamic systems that drive them. These simulations, however, are often too computationally burdensome to offer value for real-time operations in the field, and they can be difficult and expensive to run and validate due to required computing power. Validation of techniques and algorithms discovered through simulation may only be possible by using edge computing devices that are located physically close to the experimental system. This is sometimes called a "cyber-physical system", and such controllers may be tested using "hardware-in-the-loop" simulations. Such systems enable real-time control of experiments with micro-second response time, as opposed to the typical delays that are associated with communication over the internet. Moving experiments out of the lab and into the field could benefit from edge computing devices managed more through an IoT platform.

1.3 Edge Computing Platforms at Other Labs

A paper from Argonne National Laboratory details the creation of Waggle: An Open Sensor Platform for Edge Computing Devices. (Beckman et al., 2017) This platform was used to deploy air quality sensors in Chicago and a promising open-source implementation of an IoT platform. In this case, the authors developed their own open-source IoT platform. Waggle appears to offer similar features to those offered by commercial cloud providers.

A technical report from Sandia National Laboratory demonstrates the usage of Raspberry Pi devices to test assumptions of network latency for different topologies of networks. (Selorm et al., 2017) The idea is to use these devices to study the behavior of controllable, distributed energy resources in a realistic network environment. As far as we can tell, the researchers programmed their devices manually, without an IoT platform.

2 Comparison of IoT Platforms and Devices

To perform the work in this Seed LDRD, several edge computing devices were purchased and configured. The choice of hardware and software platform may have outsized impacts down the road, so it is important to do a thorough review of the various offerings in the market before committing to one platform. The three components necessary for this work were: The microcontroller, the camera, and an IoT platform. In this section, we compare several different products and services for our sky imager.

2.1 Microcontrollers

The microcontroller provides the central processing unit (CPU), memory, and I/O capabilities. Special considerations for microcontrollers used in edge computing devices include the power requirements, and the environmental operating parameters such as the minimum and maximum operating temperature for devices which will operate outdoors, and in direct sunlight.

Table 1: Features and specifications for four of the edge computing microcontrollers considered for this work.

Device Name	Raspberry Pi 3 Model B+	Arduino Uno Wi-Fi Rev2	Intel Compute Stick	NVIDIA Jetson Nano
Price	$30	$50	$120	$99
CPU	ARM 1.26GHz Quad Core Broadcom	ARM 16MHz Atmel ATMega	x86-64 1.44 GHz Intel Atom	ARM 1.54 GHz A57
GPU	Dual Core VideoCore IV® Multimedia Co-Processor	None	Intel Integrated Graphics (OpenCV)	128 Core NVIDIA Maxwell (CUDA, OpenCV, cuDNN)
OS	Raspbian (Linux)	None	Windows 10, Linux	Linux for Tegra
I/O	USB2, 40 Pin GPIO, CSI,	USB, 20 pin GPIO	USB, Bluetooth	USB3, CSI-2, PCI-e
RAM	1GB	6144 Bytes	2GB	4GB
Disk	16GB + Micro SD	48KB	32GB + MicroSD	Micro SD
Display	HDMI, DSI	None	HDMI	HDMI

Device Name	Raspberry Pi 3 Model B+	Arduino Uno Wi-Fi Rev2	Intel Compute Stick	NVIDIA Jetson Nano
Network	802.11a/b/g/n 100 Base Ethernet	WiFi	802.11 a/b/g/n/ac	Gigabit Ethernet
Power	USB 2.1 amp	7-12V	5V	5/10W

There are many different microcontrollers on the market. These devices come with community support and a large ecosystem of software packages that are pre-compiled for these devices. The ease of development for these devices is attractive for a scientific environment like NREL, where development speed may be more important than per-unit cost or various performance benchmarks. The devices we considered are listed in Table 2.

In addition to the microprocessor, there are various co-processors available from manufacturers that can add capabilities to less powerful CPUs. Some of these devices include the Intel Neural Compute Stick (Movidious) and Intel ARRRIA, which provide hardware specialized for neural network computation and FPGA, respectively. These devices were considered, but ultimately not purchased since we believed that our computer vision task could be carried out by the built-in resources of all our microprocessors and would not be significantly sped up by these co-processors.

We acquired the Raspberry Pi 3 B+, an Intel Compute Stick, and the NVIDIA Jetson Nano[1]. The Raspberry Pi 3 B+ was ultimately used in the demonstration sky imager. Its combination of a great user community with many example codes, an ecosystem of PiCam camera modules, official support from Amazon AWS GreenGrass IoT service, as well as the onboard VideoCore IV GPU, all come together to make Raspberry Pi a good first choice.

The Raspberry Pi can be extended through “hats,” which connect to its GPIO pins and sit on top of the device. One such hat that we investigated was the Pi Anywhere.[2] This hat provides the Raspberry Pi with cellular connectivity, powered by Raspberry Pi’s power supply. As NREL had provided two different cellular modems which could interface with the Pi using Ethernet and Wi-Fi, we did not end up purchasing this hat. It may be useful for projects in the future that require a more compact microcontroller assembly.

2.2 Cameras

Digital cameras are cheap and widely available. Most of the microcontrollers in section 2.1 support most USB webcams on the market. One of the challenges with sky imagery is the requirement that the camera face directly into the sun. This can cause a burn-in effect on older

[1] Setup guide: https://www.pyimagesearch.com/2019/05/06/getting-started-with-the-nvidia-jetson-nano/
[2] https://www.pyimagesearch.com/2019/05/06/getting-started-with-the-nvidia-jetson-nano/

image sensors. Newer sensors such as the Sony IMX219 have been used in sky imaging devices successfully and can withstand the high intensity light from the sun for sustained periods.

Table 2: Feature comparison of the two cameras purchased for this project.

Device Name	PiCam V2	ELP 180
Price	$28	$50
Max Resolution	3280x2464	1920x1080
Outdoor Enclosure	No	No
Connector	Pi - CSi Interface	USB
Field of View	160 degree lens (64 degrees default lens)	180 degrees fisheye lens

In searching for a useful camera module, we came across several home surveillance products on the market which package a webcam, microcontroller, and a cloud monitoring service in one product, offering a compelling off-the-shelf solution. Such products include the Amazon Cloud Cam, Logitech Circle, and the Nest Cam Outdoor. These products were considered for this project but were deemed as not suitable due to lack of API access to raw image data, the requirement of a reliable and fast internet connection, and the desire to re-program the device with custom research code. These security cameras may be viable choices for other use cases.

Another class of cameras considered were security cameras, such as the Vivotek S Series FE9381-EHV Outdoor Fish-eye dome, as well as analog security cameras which are very inexpensive. These types of cameras do not have any on board processing beyond video encoding, and some require separate analog-to-digital converters. Furthermore, it was unknown if these cameras could withstand direct sunlight for sustained periods of time, and if they were waterproof when their dome is facing upwards (which would be inverted from normal usage as a security camera, facing downwards).

The PiCam is an open-source camera board which is designed to be connected to a Raspberry Pi. There are two versions of the PiCam. V1 is built around the OV5647 camera module with a sensor resolution of 5MP. The PiCam V2 supports the superior IMX219 camera, which has a sensor resolution of 8MP. Unfortunately, all the camera modules on the market with fish-eye lenses were PiCam V1. Previous work by Richardson (Richardson et al., 2017) used a modified PiCam V2, which they modified by adding a fisheye lens.

The final class of cameras we considered were standard USB webcams, such as the ELP180 USB camera. These webcams use a standard USB interface, and manufacturers may provide drivers for Mac, Linux, and Windows operating systems. Some of these cameras are customizable, with manually controllable exposure and resolution, but some manufacturers may disable these controls.

We ordered a PiCam V2 with a built-in 160º field of view and an ELP180 webcam as a backup option. However, the PiCam V2 we received was dead on arrival, and the RMA process would have added risk to the project timeline. We ultimately used the ELP180 camera which worked out of the box with the Raspberry Pi as well as with the development laptop.

2.3 IoT Cloud Platforms

There are many cloud platform providers with offerings in the IoT space such as Amazon IoT Core, Microsoft Azure IoT, Google Cloud IoT, C3 IoT, and Siemens MindSphere. We chose Amazon IoT for this project as NREL provides ready access to an Amazon AWS account, and this service is supported internally. Further, products such as the NVIDIA Jetson support AWS GreenGrass out of the box, with documentation that is specific to GreenGrass. Although we chose GreenGrass for this project, a more thorough comparison between IoT cloud products should be made. To be clear, this is not a statement of preference for the Amazon product or an implication of its superiority to other products on the market. NREL does not endorse Amazon or Amazon's products by the publication of this report.

3 The CloudCV Sky Imager

In this section, a description of the construction of our in-house sky imager is provided. We describe the choices made in building the enclosure, as well as special considerations for IoT devices being deployed at NREL.

Table 3: Approximate cost of the CloudCV sky imager, by component, in 2018.

Component Name	Cost to Project (Excludes extra parts and services provided by NREL)	Description
ADS1115	$16	Analog to Digital Converter for the Pyranometer
CanaKit Raspberry Pi 3 B+ Starter Kit	$80	Microcontroller
ELP 180 degree Fisheye Lens Wide Angle USB Camera with Housing	$48	Camera
Pelican Case 1200	$45	Enclosure
70mm Acrylic Dome	$10	Window for camera lens
Campbell Scientific Conduit 25746, 25744 and Putty 6596	$13	Waterproof conduit for wires
Cradlepoint IBR600	$0 (NREL provided)	Wireless gateway and router
LICOR LI200	$0 (NREL provided)	Pyranometer
	Total Cost per device: $212	

3.1 Device Enclosure

The sky imager must work outside in a variety of weather conditions. We started with a Pelican Case 1200, which is a waterproof case commonly used for field work. The case was then modified by the NREL machine shop in three ways. First, an acrylic dome was added to the top, which provides a clear view of the sky. This was done by drilling a hole in the top side with a relief around the rim, and then affixing the acrylic dome in place with epoxy. Second, a large hole was drilled in the bottom of the enclosure from which a Campbell Scientific conduit was attached. This conduit uses electrician's putty as a seal, keeping water outside of the enclosure while allowing cables to pass through. The third and final modification made to the enclosure was reflective white spray-paint, which was intended to reflect as much heat energy from the sun as possible, reducing the heat inside the imager.

One concern when designing the sky imager was the potential for over-heating, as the device must sit outside in the sun, and it has a clear dome which is pointed towards the sun. We decided to cover the sky imager in white reflective paint and place a reflective white card at the bottom of the camera dome to reflect as much heat energy as possible back out through the dome. If necessary, we could have also cut holes in the bottom of the device or added a fan to provide for more ventilation. There are several different options when it comes to cooling the sky imager:

- Active cooling of permeable enclosure with an enclosure fan, and downward facing air conduits.

- Active cooling of a sealed enclosure using a thermoelectric heat coupler.

- Passive cooling of permeable enclosure with downward facing air conduits.

- Passive cooling of sealed enclosure using a reflective coating.

One possibility that was not considered, but which may prove useful in future iterations, is to use two separate enclosures: one for the camera, and one for the microcontroller and all other components. This would separate the heat load of the microcontroller from the clear window, allowing the microcontroller to be placed in a nearby but covered location, in the shade and sheltered from the rain.

3.2 Hardware

The main sensor included in the device is the ELP 180º Fisheye Lens Wide Angle USB Camera. In addition to the webcam, the sky imager also collects irradiance data from an onboard LICOR-LI200 Pyranometer. This pyranometer was provided by the NREL Solar Radiation Research Laboratory (SRRL) and outputs a voltage differential in the millivolt range. As the Raspberry Pi does not have analog I/O pins, a separate analog to digital converted (ADC) was purchased and connected to the Raspberry Pi's serial port. We chose the Adafruit ADS1115 ADC, since it features a built in 16x gain, as well as a relatively high 12-bit precision. This device was known to work with the Raspberry Pi and example Python code for this use case was available through the manufacturer.

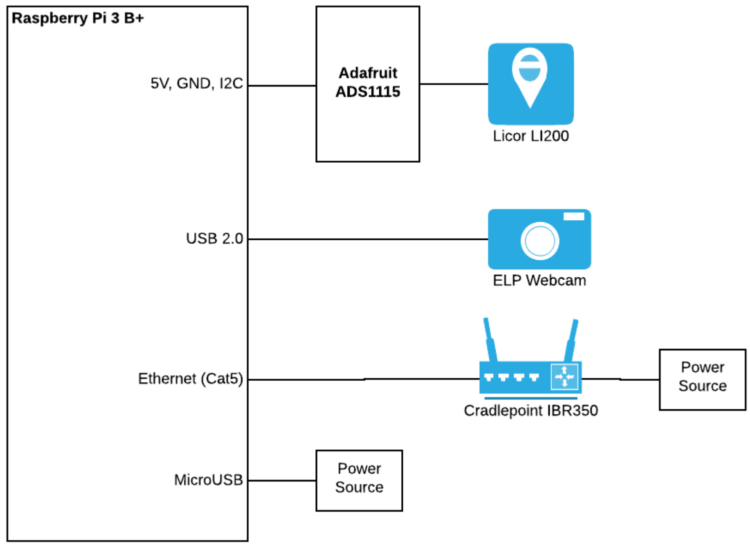

Figure 1: Block Diagram of the hardware in the CloudCV Sky Imager

3.3 Software

The Raspberry Pi was loaded with the Raspbian operating system, which is a distribution of Linux. The software lifecycle onboard the device is handled by the Amazon GreenGrass daemon, which was installed on the device before field deployment. The GreenGrass daemon executes AWS Lambda functions, which may contain arbitrary Python code, and are specified through the AWS management console. The lambda function contains the research code needed to capture images from the camera, perform any image processing, and then upload data to the cloud.

In addition to AWS IoT Core, GreenGrass, and Lambda, we used IoT Events and CloudWatch for event logging, such as recording errors. As the IoT device may be deployed in the field, it is important to be able to log errors and create a feedback loop for software development and debugging. Finally, we used AWS S3, which is an object store, to store images and pyranometer readings for later analysis. A diagram of how we used these products is provided in Figure 2.

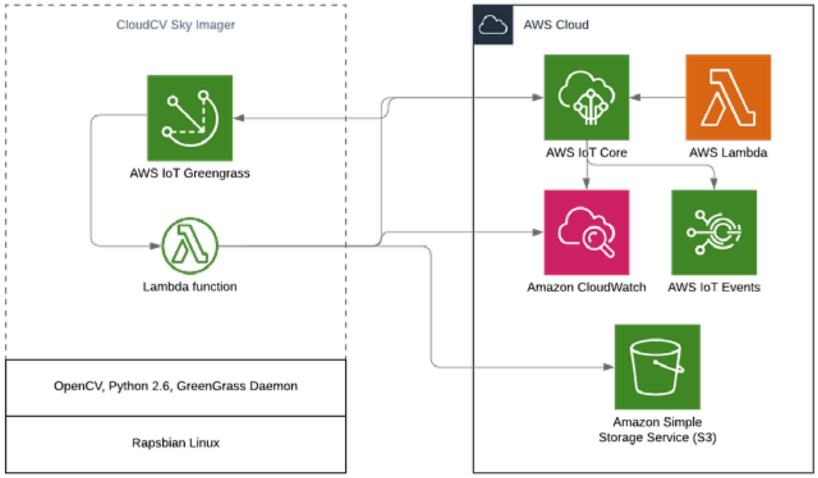

Figure 2: Block diagram of the AWS services used to control the sky imager. The box on the left represents the imager, while the box on the right represents Amazon's cloud infrastructure.

3.4 Deployment at NREL South Table Mountain Campus

The CloudCV sky imager was deployed for three months at NREL's Solar Radiation Research Laboratory Mesa Top facility, located behind NREL's South Table Mountain campus, in Golden, Colorado. This location was chosen due to its "Mezzanine" structure which provides a good platform for solar measurement devices. This platform has tables which are convenient places to strap down new devices, as well as power outlets and mount points for waterproof conduit. Additionally, the Mesa Top facility has other solar sensors for ongoing experiments that are aggregated by the Baseline Measurement System (BMS) and freely available on the internet. By placing the sky imager at the Mesa Top, we receive plentiful solar data from the same location, which can be downloaded for free and used in future analysis.

Before being placed on the Mesa Top facility, the plans for this device were discussed with NREL's Environmental Health and Safety department as well as an internal which reviewed the device for electrical issues as well as for general safety. Some key lessons learned include keeping the electronic devices under 20V DC inside our enclosure and using a NEEMA enclosure for any 120V AC supply current; removing any lithium batteries from the components of the sky imager, as such batteries are flammable and may not withstand high temperatures. For example, many commercial 4G LTE hotspot devices contain a lithium battery backup. This was the main consideration for using a Cradlepoint IBR350, which has a slower throughput since it only supports the 3G cellular network, but it does not contain a battery.

Initially, we had expected to connect the sky imager directly to the Wi-Fi network at the Mesa Top facility. However, it was determined that Raspberry Pi devices could be considered a security threat, as demonstrated by a recent hack of NASA JPL from an unauthorized Raspberry Pi[3]. Therefore, we were forced to take the more complicated approach of setting up a mobile

[3] https://www.zdnet.com/article/nasa-hacked-because-of-unauthorized-raspberry-pi-connected-to-its-network/

internet connection through a cellular gateway. This approach isolated the Raspberry Pi from the NREL enterprise network, but it comes at the expense of a more complicated device, more expensive operational costs, and lower throughput.

Figure 3: Photographs of the CloudCV sky imager and pyranometer as deployed on the mezzanine of the NREL Mesa Top facility.

4 Preliminary Analysis of the Short-Term Prediction of Cloud Cover

This section provides an outline of a preliminary analysis of cloud cover prediction using an algorithm based on optical flow and linear advection, described by (Chow et al., 2011). A short review of the current literature in cloud cover prediction using sky images is provided. Then, we describe the cloud cover prediction algorithm. The performance of this prediction algorithm is reported using the critical success index, which is an event-based metric. Finally, a discussion of the results along with recommendations for future research is given.

4.1 Background

There are several groups in the United States that are working on cloud cover prediction using sky imagers. One group from the University of California in San Diego deployed a sky imaging system to record images of the sky at 30 second intervals (Chow et al., 2011). These images are then used to produce cloud cover forecast estimates through a cloud advection algorithm. The authors of this paper claim their technique improves over baseline algorithms.

Another group is based out of the University of Texas in San Antonio. In their paper (Richardson et al., 2017), cloud position is determined using a computer vision algorithm, and then a ray tracing algorithm is applied to project the shadow of these clouds onto a terrain map. Their goal is to forecast cloud cover at an arbitrary position in some domain around the sky imager, not only at the sky imager's location. In a follow up paper (Moncada et al., 2018), it is shown that deep learning techniques can be used to extract precise irradiance values from an all sky image, without the need for a pyranometer.

Finally, we found an effort at Brookhaven National Laboratory, in collaboration with the Electric Power Research Institute and NCAR, to implement regional forecasting of cloud cover using a network of multiple sky imaging devices (Kalb, 2018).

4.2 Forecast Method

We implement an optical-flow based linear advection algorithm to predict cloud cover, similar to the algorithm described in (Chow et al., 2011). Our forecasting algorithm produces a signal which predicts events where clouds shade the sun at the 60-second time horizon.

First, the images were downscaled to 500x500 pixels to ease computational burden on the sky imager. These images have significant fish-eye distortion, which are assumed to be spherical. We perform a spherical coordinate transformation to map the fish-eye images to undistorted images. The undistorted images are then passed through a Farneback optical flow algorithm (Farnebäck, 2003), as implemented in the OpenCV (Bradski, 2000) software package's "calcOpticalFlowFarneback" function. This algorithm produces an estimate for optical flow between two images on a pixel-by-pixel basis.

The location of clouds is extracted by using a grayscale thresholding algorithm. Cloudy areas have pixels of higher intensity, close to a white color, while the sky has a darker shade. We used a threshold of 240 out of 255. Some other papers use a ratio of red to blue to detect cloudy pixels, but the ELP180 webcam in the CloudCV sky imager struggled to capture the color, so this method was unreliable.

Finally, the optical flow vectors are masked by the presence of cloud cover at their tails. These vectors were then multiplied by a scaling factor proportional to the desired time horizon. The vectors are then masked at their heads by a 25 pixel radius disk around the location of the sun. The heads of the remaining vectors represent clouded areas which are projected to cover the sun disk at the desired time horizon. The quantity of such vectors is then divided by the total number of pixels in the image, yielding a prediction signal. A visual representation of these steps is provided in Figure 4.

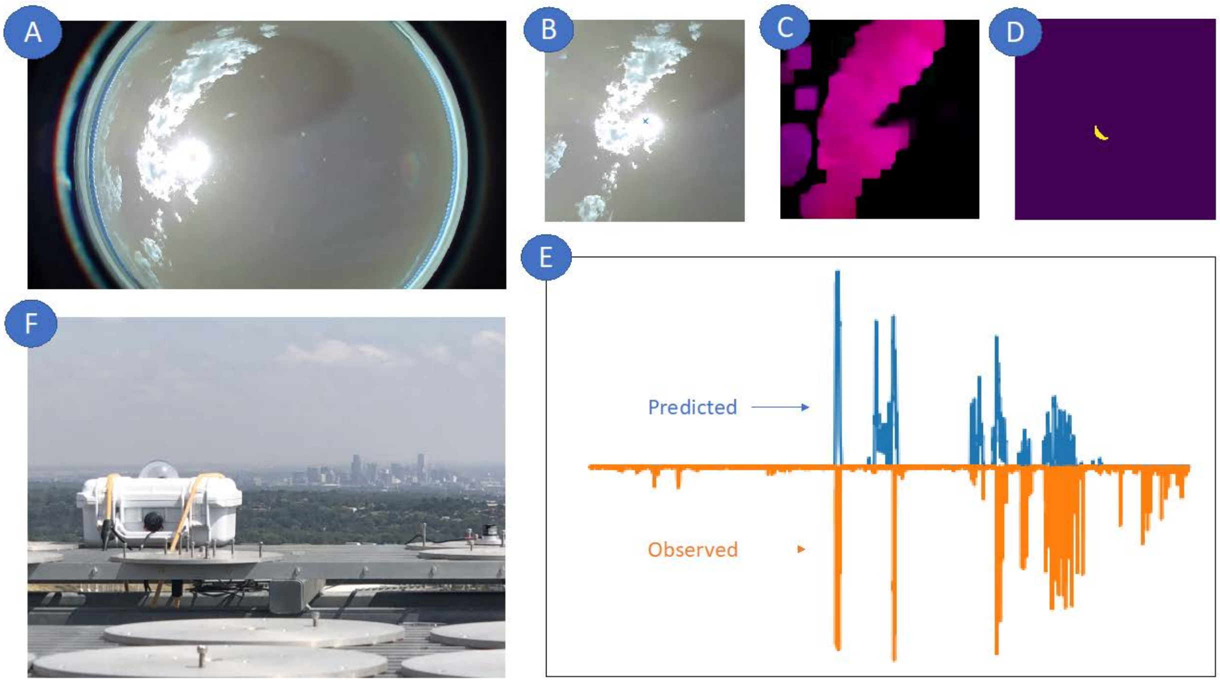

Figure 4 (A) Sample image from the CloudCV sky imager. (B) image after fisheye correction, (C) visualization of the optical flow, (D) linear extrapolation of the optical flow, yellow pixels represent clouds that are projected to shade the sun in the next minute. (E) Visual comparison of the image-based forecast signal (in blue) with the observed irradiance signal from the pyranometer (in orange). (F) Picture of the CloudCV Sky Imager as it was deployed to the NREL Mesa Top facility.

4.3 Forecast Results

This forecast algorithm was used to predict cloud cover events at a 60-second forecast horizon for one day, September 10th, 2019. Here, we evaluate the skill of this forecast using an event-based score. First, the prediction and irradiance signals are converted into discrete events using a threshold. We pick a threshold of 0.35 for the prediction signal (which is a unitless proportion), and 400 W/m^2 for the absolute difference in the irradiance signal. In both cases, we look for events where the signal has not met the threshold for at least ten minutes, followed by the threshold being met for at least a single frame. Using these parameters, we found 4 irradiance prediction events and 4 actual irradiance events on the day being studied.

To determine if a prediction event has a matching subsequent irradiance event, we look for irradiance prediction events which lead an actual irradiance event by 30 seconds to 5 minutes. Using these parameters, we find 3 true positive events, 1 false positive, and 1 false negative. This yields a critical success index (CSI) score of 0.6. It is important to note that this score represents the result of manually tuning the cloud-cover prediction algorithm for one specific day. This day was sunny in the morning, yielding to relatively small clouds in the afternoon.

4.4 Next Steps

The forecasting algorithm described was able to provide some predictive capability for sun shading events on one day with ideal conditions and manually tuned parameters. It is unknown how this algorithm will perform on other days, or how it would perform with different parameters. A more robust analysis of this prediction algorithm, spanning more days of the dataset, different time horizons, and different parameters, would be beneficial. Additional prediction methods, such as deep learning-based methods, may also be studied to further improve prediction accuracy.

Further exploration of the evaluation metrics used for event-based forecasts and the potential market value and utility for improvements in such forecasts. For example, solar generation facilities may be able to use a cloud cover prediction to optimize the use of energy storage devices, such as batteries.

Finally, the sky imager itself could be improved by adding more and higher quality sensors. Such a device would capture more data about the atmospheric state, increasing the amount and quality of input data used in the cloud cover prediction. In addition, multiple sky imaging devices can be deployed over a larger geographic region.

References

Beckman, P., Sankaran, R., Catlett, C., Ferrier, N., Jacob, R., & Papka, M. (2017). Waggle: An open sensor platform for edge computing. *Proceedings of IEEE Sensors*, 1–3. https://doi.org/10.1109/ICSENS.2016.7808975

Bradski, G. (2000). *The OpenCV Library* [Computer software]. Dr. Dobb's Journal of Software Tools. https://github.com/opencv

Chow, C. W., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Shields, J., & Washom, B. (2011). Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. *Solar Energy*, *85*(11), 2881–2893. https://doi.org/10.1016/j.solener.2011.08.025

Farnebäck, G. (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. In J. Bigun & T. Gustavsson (Eds.), *Image Analysis* (Vol. 2749, pp. 363–370). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45103-X_50

Kalb, P. (2018). *Demonstration of Improved Solar Forecasting Incorporating HD Sky Imaging*. Brookhaven National Laboratory.

Moncada, A., Richardson, W., & Vega-Avila, R. (2018). Deep Learning to Forecast Solar Irradiance Using a Six-Month UTSA SkyImager Dataset. *Energies*, *11*(8), 1988. https://doi.org/10.3390/en11081988

Richardson, W., Krishnaswami, H., Vega, R., & Cervantes, M. (2017). A low cost, edge computing, all-sky imager for cloud tracking and intra-hour irradiance forecasting. *Sustainability (Switzerland)*, *9*(4), 1–17. https://doi.org/10.3390/su9040482

Selorm, D., Ameme, K., & Guttromson, R. (2017). *Stochastic Characterization of Communication Network Latency for Wide Area Grid Control Applications*. *October*.