Machine Learning, Sensing, and Image Processing for Snow Research

Project Report
Spring Semester 2022

Senior Design team members:
Luke Aldana (Electrical Engineering)
Andrew Flores (Computer Engineering)
Peter Walsh (Computer Engineering)
Isaac Jacobson (Computer Engineering)
Luke Mason (Mechanical Engineering)
Aaron Tai (Mechanical Engineering)

Other team members:
Hein Thant (PhD Student)
Naing Win (VIP Electrical Engineering)

Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, Colorado 80523

Project advisor: Dr. Branislav Notaros

Approved by: Hein Thant
Date of approval: 5/6/2022

Abstract

The ability to accurately predict the weather is extremely valuable in today's world. Snow drastically impacts the ability to work outside, and the Snowflake Measurement Analysis System (SMAS) gathers information on snowfall in hopes of better predicting the weather in the future. This product is still in the development phase, but there is potential for commercial use: allowing the consumer to gather detailed information about snowflakes and the atmospheric conditions which generated them wherever they please.

While the construction of the device was a major facet of this project, many of the exciting advances happened in the software. The team was able to successfully refine the previous code to take seven simultaneous images, while one camera takes an additional long exposure picture to help determine the fall speed. This improves upon the previous method which only had five cameras that operated with a lower maximum frame rate. Custom 3d printed snowflakes were utilized during testing to further increase the accuracy of the experiments.

The SMAS was delivered to Wallops Island in Virginia earlier this year, though it was a great effort to get it prepared by the deadline. The device is capable of capturing in-focus images of snowflakes, and storing those images to be accessed and processed later. There were many obstacles involved in this process, but multiple water and functionality tests were conducted at room temperature as well as below freezing to simulate a snowstorm. After collecting data nearly continuously for several months, the device has gathered hundreds of thousands of images of snowflakes. Preliminary analysis of the images and the existing literature surrounding snowflake classification gives the team hope that it will be possible to demonstrate very accurate snowflake classification with those images. Future hopes for this project involve a new iteration of this device that is more robust and commercially ready as well as improving the image classification and fall speed determination algorithms. In terms of making the device more robust and commercially ready, the device is currently quite complex with many parts to keep track of at once. It would benefit users if the device was more simple and thus had fewer potential points of failure.

Table of Contents

1. Introduction

As the field of machine learning continues to advance, the ways in which it is used are increasing in both number and complexity. One of the main limitations of these advanced methods of machine learning is that they typically require a large amount of training data in order to work effectively. Therefore, those who hope to use machine learning and especially neural networks must start by obtaining a high volume of samples which are representative of the information the model will be used with. In general, the more training data one has, the more useful the machine learning model will be in terms of accuracy. That generalization is particularly true when considering data quality like resolution and focus, where datasets with high quality samples in their training are likely to perform better than those with lower quality samples as long as the inputs after training have similarly high quality.

The goal of this project is to develop a way to gather high resolution, focused images of snowflakes and then use those images to create a machine learning model which can classify those snowflakes. If we are successful, this information can be used to aid weather forecasting by providing details about what kind of snow is likely falling in any given snowfall situation. That obviously has a myriad of uses from helping ski resorts predict how much artificial snow they may need to create (or even what kind of artificial snow they may need to create to compensate for a lack of 'fluffier' snow) to helping an average person predict if the road will be icy or covered in powdery snow after a snowstorm. Knowing the class of the snowflakes which are falling can also tell atmospheric scientists about the conditions high up in the atmosphere that created the snowflakes. In addition to supporting weather forecasting with additional data in the form of snow images, the device will also collect a variety of other measurements like temperature, humidity, and dust levels. While these measurements will be skewed due to being enclosed with the rest of the measurement device, they could still be valuable tools in situations where there is otherwise nothing else measuring them. Lastly, our project will result in a dataset of high quality black and white snowflake images and detailed environmental characteristic information which may prove useful to other projects in the future. Although the dataset is specifically curated for snowflake classification, it is possible it may be useful in some other capacity, especially given the volume of images expected and the additional data collected alongside the images.

This report contains information about what the team has been able to accomplish in the most recent two semesters of the project as well as the main things the team would like to see improved in the future of the project. For many of the sections, we split the information into a mechanical section, an electrical section, and a computing section because many of the tasks for this project were executed in parallel with regard to those three groups. By dividing our design decisions this way, we hope to make it more clear which of us were managing those decisions and to make it so that readers interested in specific areas of our engineering design can find them more easily. First, we discuss what the teams on this project have done before us and what they left for us. Then, we summarize the various tasks that were completed throughout the Fall semester including certain decisions that were made and why those decisions were made. After that we describe the work that we did in the Spring semester including the various new approaches we investigated to potentially improve on the design. Next we elaborate on any industry standards we either tried to conform to or succeeded in conforming to so that users can know what they can expect from the device, and lastly we discuss the work we expect future teams assigned to this project to complete. There is also an appendix at the end of the report which includes any abbreviations we use, our detailed budget, our project timeline, and any additional images we didn't include in the main section of the report.

2. Summary of Previous Work

2.1 Electrical

The previous iterations of the project completed a great deal of custom electrical engineering work. All seven of the cameras, camera flashes, photodiodes, and the two cross-plane lasers were all connected and mostly functioning prior to this year. That includes a Pickit 3 that was mounted on the side of the SMAS in order to program the main chip that controls the lasers, photodiodes, and flashes. There is also an HP Z440 computer on the SMAS which previous teams added in order to manage the cameras and potentially perform some preliminary image processing in real time. This computer was stripped of its case and mounted to a back panel which was then bolted to the chassis of the SMAS. Also, the lasers, flashes, and camera triggering mechanisms were all connected to one complex PCB (see Figure 1) mounted on the side of the SMAS. This PCB handled all of the timing and synchronization for the SMAS. Since this PCB controlled many operations, it was vital to understand how it worked in order to debug the system.

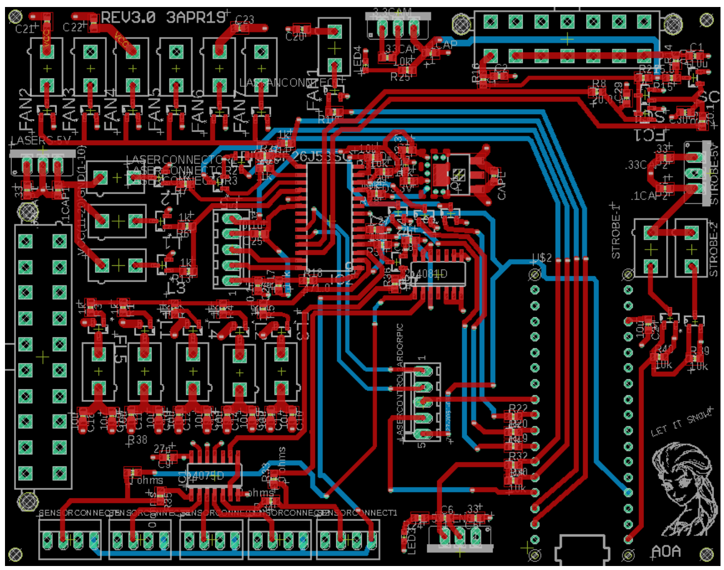


Figure 1: Central custom PCB boardview

There were a few issues with the electrical design done in the past, namely issues revolving around wire management, power flow, and an outdated PCB. There was seemingly very little effort put into managing the various wires connecting all the components which made adding new components extremely difficult. There were a few cases where there were completely unconnected wires hanging from the computer and from the PCB. This was a big issue since the system is already so complex, having to trace out which wire did what and which ones were important in the overall functioning of the system was very difficult. There were some clues in the form of stick-on wire clips that previous teams had attempted to improve the wire management, but it was clear that this would be one of the most important things to improve during the time before shipping the device. There was also an issue relating to the flow of power through the system. The PCB would sometimes have too much power supplied such that, since the lasers and flashes consumed so much current, it allowed some of the components to reach extremely hot temperatures. This is obviously not good and an attempt was made by a previous team to help with this in the form of a MOSFET heat sink, but there was only one heat sink and a few other components needed them too. Additionally, the Pickit 3 needed to supply so much power to the chip on the PCB when being programmed that it would occasionally melt the jumper wire connected to Vdd. This melting of the wire would disrupt the communication between the Pickit 3 and the chip, meaning the wire needed to be replaced before programming the chip again. For the PCB itself, a lot of the functions on it were no longer being used. For example, the PCB included an Arduino Mini that had no use in the current iteration of the circuit. Additionally, a few flashes were no longer being used. All of these problems impacted the performance of the device and therefore it was critical that we come up with solutions to remedy them.

2.2 Computing

Regarding the software portion, previous teams have done a tremendous amount of work. The C++ code for the camera's have been substantially developed, as well as the laser and camera flash C++ code. The code ran just fine and the SMAS was able to take a single picture per camera ever since the beginning of the semester. Additionally, last year's team specifically developed a GUI that makes communicating with the SMAS a lot easier. With this new GUI the SMAS software can be turned on/off, the cameras can be controlled, and various other parameters could be set up.

The GUI is great but it does come with its flaws. It has plenty of buttons and functions; however, none of them are actually set up to function with the current code. So some extensive modification of this GUI code would be needed in order to get it to actually function with the current code. We did not end up using this GUI, but some of its design elements were considered in the design of the web application which allows users to interact with the SMAS remotely.

2.3 Mechanical

In previous years of this project senior design teams have constructed an impressive SMAS chassis complete with various mounts, strategically placed holes to feed electrical wiring through, and an outer shroud to help weatherproof the entire system. The mounts were mostly 3D printed and allowed for the seven different cameras, camera flashes, and lasers to be placed in the most ideal positions to capture snowflake images. The pre-drilled holes in the system allowed for discretely feeding electrical wire anywhere in the system and, to complete this design, the previous teams have weather proofed some of the holes with silicon and electrical tape so water does not seep into the electronics. The outer shroud was probably one of the most important additions to the mechanical side of the project as it fit almost perfectly over the entire system, electronics and all, blocking out snow, water, dust, etc. The shroud also allowed for snow to fall into the center of the SMAS but blocked snow from reaching any of the electronics. Additionally, the shroud looks very well-made and professional. It is complete with handles to help slide the shroud on/off the system and has mounting holes at the top where one could secure it to the rest of the system using nuts.

Another important mechanical aspect that was added in the past was the little trolley meant to aid in the transportation of the SMAS. This trolley is made up of a little cart with four wheels and a long handle to allow for pulling it to the destination. Even though the trolley was purchased by a previous team and not student-built, it is still a vastly important contribution to the mechanical side of things that has been used many times to transport the SMAS.

Though all of the mechanical additions over the last few years to the SMAS system have been fantastic, some of them did not function as well as intended. The weatherproofing silicon, electrical tape, and outer shroud were not enough to waterproof the system to an IPX5 standard, which consists of spraying the device with a low pressure hose for a designated amount of time. This was obviously a problem since the SMAS needs to be able to sit out in wet conditions for extended periods of time. Additionally, only a few of the drilled holes were sealed with silicon, some holes were left wide open still, probably from a last minute change in electrical wiring by a previous team. Additionally, even the shroud did not completely block out water coming from the top. This is a huge issue since a lot of the moisture is expected to be in the form of snow. If the top of the shroud is letting in even the tiniest amount of moisture from snow, our electronics could get fried quickly or slowly over time. Either way it is an issue that needs to be readdressed. Lastly, some of the mounting brackets for the cameras, lasers, and flashes were either broken or loose. So even though the mounting was well done, some more work was needed to get the mounting situation back under control.

3. Fall Semester Progress

Throughout this project, there have been many different subtasks that needed to be completed. As we had electrical, computer and mechanical engineering senior design students, the tasks were typically broken down into the three respective engineering disciplines. This section of the report will cover the progress that was completed during the Fall semester.

3.1 Electrical

Upon receiving the SMAS in the spring, the system was in a partially working condition. One of the most critical pieces that we had to implement was only taking images of the subject when it was in the center of the tube. The original method for triggering the cameras was to have a plane laser pointing into a sensor built from an array of photodiodes which worked well. The downside to this was that snowflakes would trigger the laser even when they were not centered in the tube. To augment this, we successfully implemented a crossplanes approach. This was done by using another plane laser and sensor at a different position so the two plane lasers intersect in the center of the tube (see Figure 2). Due to the spread of the lasers, the trigger point in the tube was about 2x2". Boolean logic had to be implemented to read the data from both of the sensors and only send an output when both lasers were being triggered simultaneously, which would only happen when an object was at the intersection point of the two lasers. A small hand-soldered breadboard combined the two sensor signals with a TI SN7400N 4-gate 2-input NAND IC. The output from these signals was fed into the custom PCB (see Figure 1).

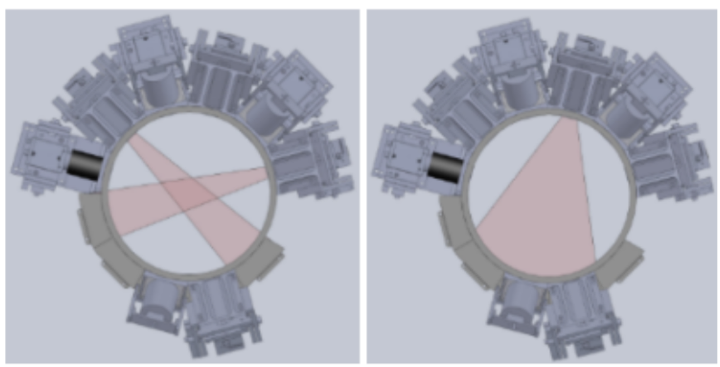


Figure 2: Bird's eye view of plane laser patterns

Given that the SMAS is an extremely complex piece of equipment that incorporates so many sensitive components, it was critical to have an extra system that would log the vitals of the entire SMAS. Thus, exploration into the implementation of a CAN Bus system commenced. The CAN Bus had a few different constraints that ended up influencing the final design. The biggest factor being the vast number of different items that needed continuous monitoring. For example, it was known that 10 temperature sensors, at minimum, were needed along with a humidity sensor, a dust sensor, five rain sensors, and more. This meant that whatever controller was chosen better have upwards of 20-30 GPIO pins at minimum. Another constraint was that the CAN Bus must be completely independent of all other components, including the main computer, since it is crucial to have the CAN Bus operate as reliably as possible. Additionally, earlier in the semester, Arduino-compatible temperature and humidity sensors were already purchased for the project. So, given these constraints and in order to avoid extra costs, it was decided that using Arduinos' might be the best option for the CAN Bus. However, it would be difficult to record this data and keep the data in a central location without the use of the main computer. To combat this issue, it was decided that a Raspberry Pi would interface with two different Arduino Megas'. The two Arduinos' would collect data from all of the sensors that monitored the temperature, humidity, moisture levels, and dust levels via GPIO pins then would forward this data directly to the Raspberry Pi through a UART protocol using a serial TTL connection. It was realized that using a UART protocol for a CAN Bus system was not nearly as efficient as using the actual CAN Bus protocol; however, the data being recorded is not nearly as time sensitive as typical applications of normal CAN Bus systems. Thus it makes sense in this scenario to sacrifice some efficiency for more appealing costs and simplicity since there was also a strict deadline for this project as well.

The Arduinos, Raspberry Pi, and any remaining sensors needed were all ordered and, once arrived, were all set up and interfaced with each other. This process of setting everything up was no easy task and took a few weeks to complete. The first task completed towards this end was plugging in the rain sensors, temperature sensors, and humidity sensor into the GPIO port of the Arduino then programming the Arduino to successfully read the GPIO pins from these sensors. This was straightforward and did not take long. The next step was to interface the Arduino Mega with the Raspberry Pi. This entailed obtaining a TTL to USB cable and connecting the Rx pin of the TTL cable to the Tx pin of the Arduino and the Tx pin of the TTL cable went to the Rx pin on the Arduino board. Then the ground wire was connected to ground on the arduino and the USB portion was plugged into the Raspberry Pi. Once this was done, allowing the Raspberry Pi and Arduino to communicate with each other was just a matter of following a couple of online tutorials. This portion was pretty straightforward and did not take all that long to complete. The last portion was by far the hardest part. This entailed now interfacing the Raspberry Pi with the main computer via both WiFi and serial communication yet again, where WiFi will be used as a backup in case the serial communication fails. This was challenging mainly because there were suddenly so many moving parts and multiple devices interfacing with each other. It was challenging to figure out at this point what the correct interface name should be when coding the interface script in Python. For the Arduinos' these interface names were specified as '/dev/ttyACM0' for the first Arduino and '/dev/ttyACM1' for the second Arduino. The Raspberry Pi however was specified as '/dev/Serial0'. Once this specification was figured out, interfacing the Raspberry Pi with the computer became trivial.

Once all of the interfacing was completed, it was then crucial to test for packet losses between the Arduino-Raspberry Pi communication and between the Raspberry Pi and computer communication. This process entailed coding a test script on one Arduino that would send data once per second and another test script on the Raspberry Pi that would receive data the Arduino data and store it in a log file. The script would run for 30 minutes meaning that 1 packet * 60 seconds/minute * 30 minutes = 1800 packets should have been sent. After 30 minutes, the data being collected on the Raspberry Pi was viewed, revealing that the actual amount of packets received was 1280. Performing the error calculation found in Figure 1 below, the packet loss was found to be:100 * |(1280-1800)/1800| = 29%. This means that the communication between the Raspberry Pi and Arduino alone had a packet loss of about 29% which is not good at all. This test was reported a few times and the results were consistent around a 29% packet loss each time. The packet loss between the Raspberry Pi and computer was tested in the same exact manner. The packet loss found here ended up around 13%. Adding up the two packet losses comes out to a grand total of 42% packet loss between the entire communication system of the CAN Bus. This is obviously egregious since almost half of the sent packets are being lost to the void and, unfortunately, there was little time at this point to order new components in an attempt to correct this since the SMAS shipping deadline was quickly approaching. So a 42% packet loss would have to be sufficient for now.

$$PercentError = 100 * |(Expected - Theoretical)|/Theoretical$$

Figure 3: Percent error equation, created using LaTeX

Once the packet loss testing had concluded, scripts collecting real, live data were implemented and tested on the Arduinos' and the receive script on the Raspberry Pi was modified to accommodate this data. The Pi would then reformat the data into a more human-readable format and then transmit this new data to the main computer via the TTL cable. The data that was received on the computer would then be logged in a file as well as displayed on the locally hosted website so that a live display of the SMAS system vitals could be viewed in real time on the website.

In addition to the CAN Bus, there was also quite a bit of electrical wiring to redo and some power issues to resolve. The previous years teams did a good job of wiring up everything; however, most of the wires were either crossing over each other, free hanging, or not needed anymore. As there were many electrical components attached to the SMAS, it needed to be clear where the cables run through to prevent a bunch of wasted time tracing back wires if a power issue arises. Additionally, having loose, free hanging wires would not be a good idea when it comes to shipping the SMAS. If it were shipped in that state, the probability that a wire would get ripped out is much higher than it would be if the wires were neatly tucked. In total, the team spent about 80 hours making the wires and cables look more neat and efficient due to the complexity of the device. Apart from that, the team also had to 3D print multiple fixtures to secure loose electrical components. Cable management and 3D printing fixtures were done simultaneously as they were complementary parts and both had the functions of making the inside of the device more robust and more aesthetically pleasing. The figure below shows the cable management inside the device before and after the SMAS team worked on it. Regarding power, somehow the SMAS experienced two different power surges in the same semester. Nobody is entirely sure how it happened or if it was caused by an external factor like a short circuit but, nonetheless, it was a priority now to prevent any possibility of future surges and power issues in general. After a bit of research, it was decided that an uninterruptible power supply (UPS) in conjunction with a surge protector would solve the power surge issues in addition to any power outages the SMAS might experience out in the field. The UPS and power supply were both ordered, and immediately installed into the SMAS once arrived. So far there have not been any more power surges with the SMAS.

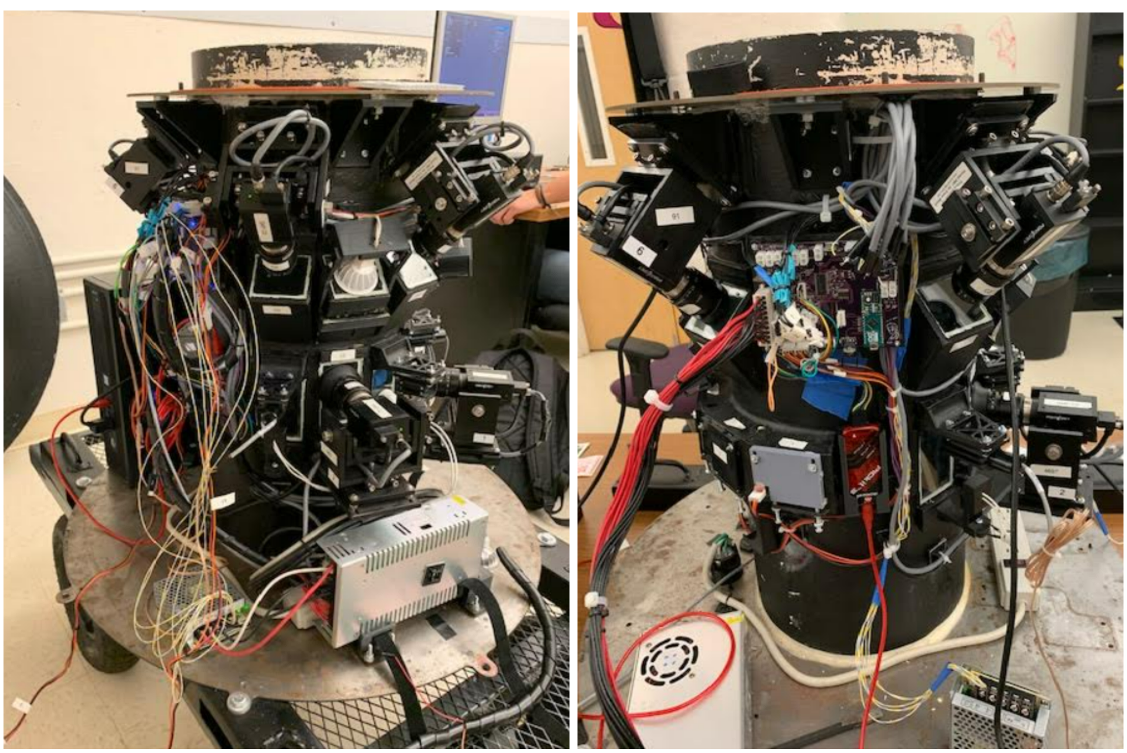


Figure 4: Before cable management by the team (left), after cable management by the team (right)

3.2 Computing

The computing aspect of the SMAS includes anything related to code, the main computer, and all of the smaller computers which control the sensors, lasers, flashes, and cameras. Over the past semester, a wide variety of tasks have been accomplished regarding these parts of the system including adjusting the lasers to properly trigger only when something is in the center of the tube, creating a web-based interface for interacting with the SMAS in a limited capacity, developing a vitals-monitoring system with various sensors, modifying the code to take an additional long exposure image with one camera for fall speed calculation, and testing all of these things.

It is crucial to be able to interact with the SMAS remotely so that its users can ensure it is working properly, get preliminary results, and even start the code which saves images without having to attach a bulky monitor to the computer or even be near it at all. To that end, we created a very simple user interface based on a locally hosted website which provides all of those functionalities. With the click of one button on the website, one can turn on the code which begins managing the cameras and saving images when new ones are taken. As long as the website's helper script is running (both the website script and the helper script run automatically on startup) the website also displays the most recently taken 7 images, which allows the user to ensure all the cameras are clear of debris, are focused properly, and are actively taking and saving images when they're expected to. The last thing the website is currently capable of is displaying sensor readouts in real time. That same helper script reads in data from the sensors using a serial connection to the Raspberry Pi which manages them, and then that data gets displayed on the website. The website automatically refreshes itself every 5 seconds, so even though the images and sensor data may be updated faster than that behind the scenes, the user will only see those updates once every 5 seconds. This decision was made because a higher refresh rate isn't necessary for these two uses, especially considering the sensor data is automatically logged at a higher refresh rate so when an anomaly is noticed the user can go back and investigate the logs.

Another design objective is to be able to measure the speed at which the snowflakes are falling as they go through the tube. Initially, we wanted to have one of the cameras take 7 additional images very quickly so that one could extrapolate the fall speed from the distance traveled between each of the 7 images. Unfortunately, we were unable to complete that approach before one of our testing deadlines passed due to difficulties manipulating one of the cameras to take that many pictures so quickly. We believe the connection between the computer and camera was not capable of saving those images fast enough either because of a lack of bandwidth or because of a lack of computing power available to manage the transfer. Therefore, we switched to a technique where one camera takes a longer exposure, reduced gain image so that the snowflake would be blurry in the image. Then, by considering the exposure length one is able to determine how long it took the snowflake to travel the distance from the top of the blur to the bottom of the blur, and since the size of the snowflake can be determined by the other image from the same camera, its fall speed can be calculated as the distance the center of the snowflake traveled divided by the time it took to travel that distance.

3.3 Mechanical

Mechanically speaking, the objective was to create a device that is robust enough to be transported via aircraft and vehicle. The device needs to be able to withstand the vibrations during transportation without being damaged. On top of that, the device also has to stay outdoors for long periods of time and collect data in extreme weather conditions, such as heavy snow and heavy rain.

To approach this problem, our team decided to first create a mounting frame for the device. The mounting frame is extremely important for two reasons. Not only does the mounting frame ensure the device is in an upright position while collecting data for fall speed calculations, it also makes sure that snow and water will be collected underneath the device instead of around it. Multiple iterations of the mounting frame was done to create a fixture that could withstand a 300lb weight and 50mph wind speed without breaking or tipping. While creating the frame, an important consideration was to minimize the erector set connections to minimize possible torsion forces. Hand calculations were done to check the capabilities of the mounting frame and a picture of the final iteration of the mounting frame is shown below:

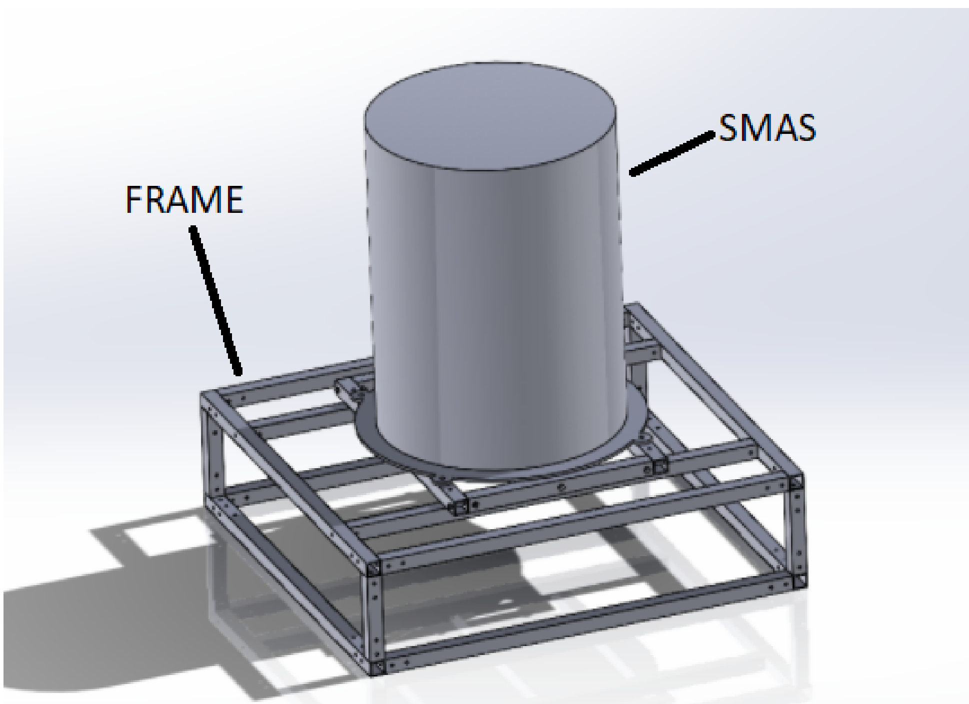


Figure 5: Mounting frame with SMAS on it

The next task that our team had to work on was waterproofing the device. To do so, our team first replaced the old gaskets on the device with brand new gaskets. This was an important step as the old gaskets were wearing out and were no longer preventing water from leaking into the device. After that we investigated the device and determined the potential water entry points of the device. We applied a sufficient amount of silicone to all of those entry points and allowed it to dry before testing (which will be discussed in the Standards section). The team decided that silicone was the best choice due to its ability to expand and contract without breaking. That's an important factor as the device will be collecting data in a huge range of temperatures. To further waterproof our device, the team also applied silicone to all the holes where nuts and bolts were used as fasteners to ensure all of the holes were water tight.

Once the device was properly waterproofed and weatherproofed, it was time to move onto performance testing under varying temperatures. The water test procedures and results are shown in section 5 (Standards). To conduct these temperature tests, our team installed 10 temperature sensors inside the device at critical locations. We then allowed the device to run at full capacity for a whole day at two different temperatures – room temperature (around 20C) and inside of a freezer (around -25C). The parameters of these tests were used to simulate the maximum and minimum temperatures that the device would have to withstand while collecting data. The goal of the test was to ensure all temperatures ranged in between -10C and 50C as those are the functioning temperature of all the electrical components inside the device. Here is the result of our temperature test:

Temperature Sensor	Room Temperature Test	Freezer Test
T1	42.2C	-5.2C
T2	40.1C	-3.6C
T3	37.8C	-3.9C
T4	39.7C	-7.2C
T5	45.0C	-8.3C
T6	43.2C	-6.7C
T7	44.8C	-5.5C
T8	39.6C	-3.9C
T9	48.2C	-9.2C
T10	44.1C	-5.6C

Table 1: Temperature tests results

4. Spring Semester Progress

Much like section 3, this section covers the specific tasks that were progressed in the Spring semester, organized by electrical, computer, and mechanical contributions again.

4.1 Electrical

The current version of the SMAS has temperature, humidity, and particle detection sensors that measure the ambient surroundings; however, having the capabilities of measuring the atmospheric temperature and humidity could allow for additional information about the type of storm and what snowflakes to expect. Interestingly, it is possible to measure atmospheric temperature and humidity by using a lidar pointed at the sky. This concept is currently being used to determine atmospheric temperature and humidity in the Raman lidar which uses an Nd:YAG solid state laser at wavelengths of 1064nm, 532nm, and 355nm [2]. The water molecules in the air will scatter the photons from the lidar causing some to reflect back to the lidar. It is this photon reflection that the lidar is able to measure and the amount of photon reflection is correlated to both the humidity and temperature in the atmosphere. A similar concept will be needed on the new design for the SMAS if atmospheric conditions are to be measured. Some limited testing has been done on this matter using a simple $20 lidar. It was determined that this lidar may not be the best fit for measuring humidity and temperature since it is mainly suited for distance measurements using a single wavelength. It is hypothesized that obtaining a laser that can produce the 1064nm, 532nm, and 355nm wavelengths that the Raman lidar uses would be best; however, much more research is needed on this matter. Additionally, the Raman lidar utilizes some fancy photonic counting techniques to determine the amount of backscatter from the atmosphere. More research is needed in this area as well before another lidar or laser is purchased. Lastly, a simple plexiglass box has been constructed for future temperature and humidity lidar testing. With this box, humidity is able to be introduced to the chamber via a humidifier which can then be used to test future lidars with varying humidity and temperatures. It may be desired to include a waterproofed humidity and temperature sensor inside the chamber so that these values can be recorded for each test.

In addition to the atmospheric lidar that was discussed above, an RF doppler radar will need to be implemented onto the new SMAS system. This doppler radar will have the capabilities of detecting individual snowflakes, measuring snowflake fall speed, and calculating the densities of individual snowflakes. The idea behind calculating snowflake densities and other physical parameters based on RF radar measurements has been studied by numerous researchers and one particular study found that using three different radar frequencies provided much better results in this regard than single frequency radars [3]. A lot of testing and progress has been made this semester towards a small and simple 24 GHz doppler radar module called the CDM324. This module is very low power and single frequency so the results that can be achieved are somewhat limited compared to a module of the three frequency variety as described above. However, when performing isolated testing on the CDM324 using pyramidal RF absorbers to limit the amount of unwanted RF energy bouncing back to the module, the results that were achieved seemed promising. Clear voltage spikes could be seen for any metal, glass, or plastic object that was dropped in front of the module, even for snowflake sized metal bearings, with the exception of styrofoam. This is excellent progress but none of the objects dropped in front of the RF module had any clear distinction between voltage spikes (i.e. the smallest bearings produced the same amplitude spikes as the big bearings), so no conclusion can be drawn from these tests yet. Additionally, Fast Fourier Transforms (FFT) were performed on all of the data in hope that some correlation could be drawn from looking at the frequency domains of the signal. The FFT was performed in Matlab by using Matlab's fft() command. Unfortunately, there was no clear correlation found in the frequency domain when different objects were detected by the RF module. So, it can safely be concluded that the CDM324 is not the best doppler radar module to use for this project and a three-frequency module would be preferred or three modules operating at different frequencies.

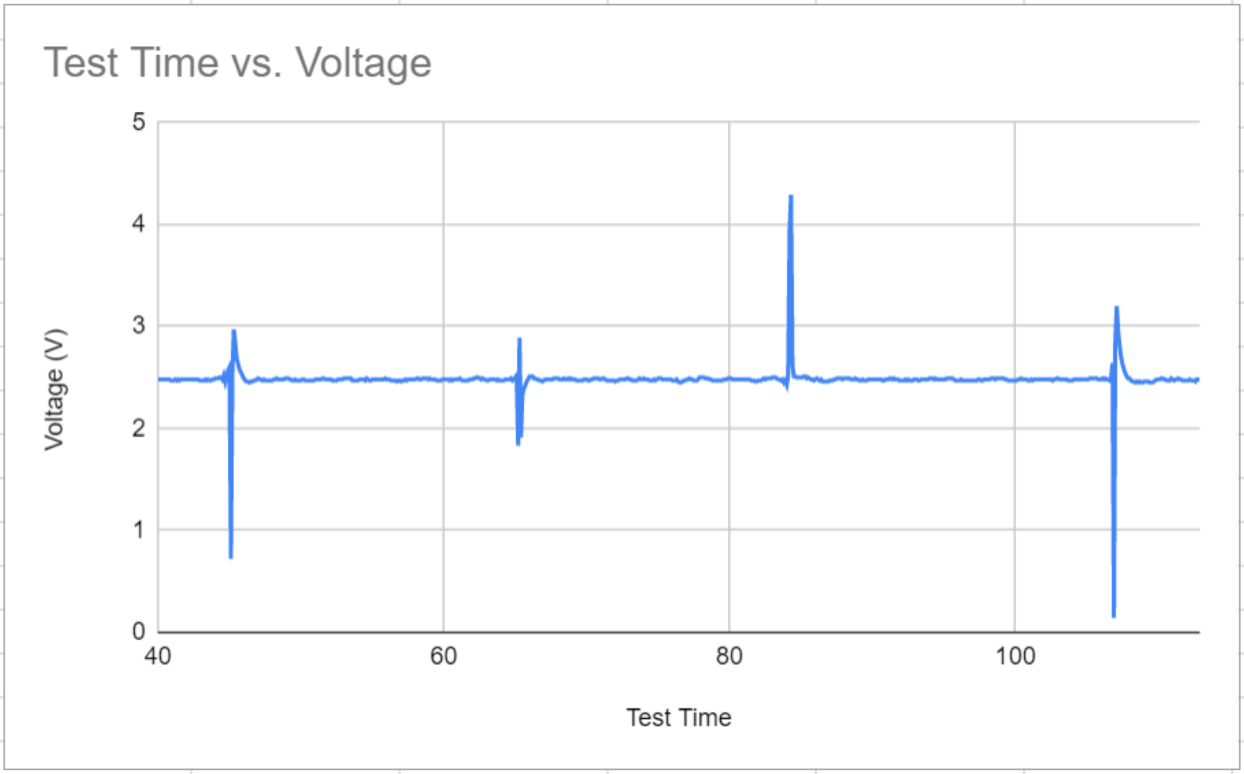


Figure 6: Smallest bearing test with the CDM324 doppler radar module

Adding onto the topics previously discussed, a new CAN Bus system may also need to be implemented on the next iteration of the SMAS. The current SMAS has a make-shift CAN Bus implemented using two Arduino Megas and a Raspberry Pi as discussed in section 3.1. This system is certainly not as robust or efficient as it needs to be. A lot of experimentation with a new CAN Bus module called the MCP2515 went underway this semester. This module is an actual CAN module, meaning that it utilizes the same CAN protocols that the industry uses. In addition, these modules are versatile and numerous modules can be implemented on the same system to communicate information back and forth. With these modules, the SMAS system's current 'CAN Bus' would experience minimal packet loss, high communication speeds, and less cost compared to an Arduino Mega or Raspberry Pi. The current testing with the MCP2515 has been somewhat limited and definitely needs to be continued in the future. In order to get the module to work with a Raspberry Pi, a 5V power supply PCB trace needed to be cut and a 3.3V supply wire needed to be soldered in its place. This is because the Raspberry Pi is capable of supplying 5V but it is not able to handle a 5V input signal on any of the GPIO pins. This is a problem since the module could otherwise provide a 5V input signal to the Pi, causing the Pi to potentially get damaged. Once this was completed, a temperature sensor was hooked up to the MCP2515, which itself was interfaced to the Raspberry Pi. Then, the module was tested using sample code that was provided. Unfortunately, the module did not seem to communicate with the Pi. This may have been caused by faulty code or a faulty board (the PCB trace may have been cut incorrectly or did accidental damage to the board). More research into this aspect should for sure be done to improve the SMAS CAN Bus system.

4.2 Computing

The SMAS took images of snowflakes at the NASA facility in Wallops Island, Virginia from approximately November through February. While deployed, it captured more than 200,000 usable images of snowflakes. This presented the team with a few unique challenges about the next steps. For example, it became apparent that it would not be feasible to go through and manually label each image while the device was deployed because the internet connection while it was deployed was somewhat unstable. Additionally, this data originally took up more than a terabyte of space on the local solid state disk drive (SSD) so it also became apparent that if there was some way to reduce the storage required to store the data that would be very beneficial. Lastly, progress still had to be made in terms of determining the fall speed of the snowflakes and developing machine learning models to classify the snowflakes. In the interest of making the best use of the time remaining, those problems were broken into several sub-tasks that could be fixed in parallel.

First, to solve the problem of the images taking up too much room on the SSD, various compression methods were applied and research was initiated into using mirrors instead of using so many cameras so that each image would have less wasted space. After careful analysis of several different kinds of compression, the Portable Network Graphics format, more commonly known as PNG format, was chosen because it is lossless but still compresses most images to less than a third of the original storage space. Between converting the images from their original bitmap image format to PNG and then sending them to a compressed .ZIP file, significant storage savings were achieved. Over the course of a few weeks, many of the images were converted into PNG's, sent to a compressed .ZIP file, and uploaded to Microsoft OneDrive, where they could be more easily downloaded.

In efforts to reduce the overall cost of the machine, we began looking into using mirrors to simulate multiple camera angles instead of using one camera for each snowflake angle. The concept behind this is fairly simple: if a single camera can point at mirrors oriented at different angles around a central object, each mirror would produce a slightly different virtual image of that object, simulating multiple cameras taking images at each of those angles. In the figure below, the mirrors on the far edges of the mirror banks form a 90° angle which can be incredibly useful for determining if an object (snowflake) is in the center of the tube. This was previously determined by using two plane lasers whose intersection was in the center of the tube. If the object can be seen in both perpendicular mirrors, the object is centered. With some image processing, each virtual image can be cropped and resized so the object of interest can be better seen. A few points of consideration for this approach include camera viewing angle, subject illumination, image resolution, and mirror reflectivity. The camera viewing angle must be offset from the plane the mirrors are positioned on in order to prevent infinite mirror effects with the opposite banks of mirrors. The subject must be sufficiently illuminated so that the wide field of view and exposure time of the camera may be calibrated correctly to produce quality images. The resolution of the snowflakes will highly depend on the field of view and physical distance of the camera. If the camera can see across six mirrors instead of zooming in on one, we will be sacrificing resolution for quantity. The tradeoff may be worth it, but more research is needed to weigh the costs and benefits. Finally, the reflectivity of the mirror must be considered. If there is significant distortion to the images because of the reflection introduced, a mirror with a smoother finish or coating must be used.

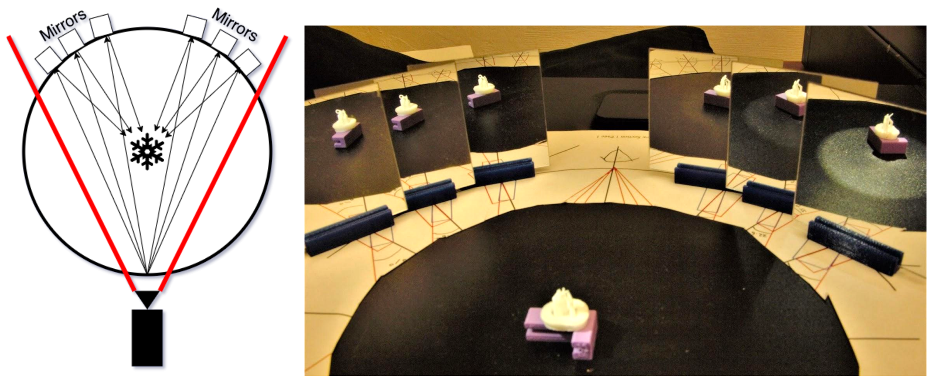


Figure 7: Single camera, multi-mirror setup

While the images were being converted and uploaded, work also began on designing the system for classifying the images. Convolutional Neural Networks (CNNs) are very helpful in applying deep learning technology for image classification purposes so they were chosen as the type of model which would be used at the end of the process to classify the images. To help a convolutional neural network learn to classify images of snowflakes more quickly, it is useful to crop each image so that it mainly contains the snowflake and contains the least extraneous information possible. It is important to crop each image to the same shape so that the images do not need to be resized later to be inputs to the convolutional neural network. This task is sometimes very difficult because, although the snowflake is supposed to be the brightest part of each image, there are many cases in which cameras experienced glare from the flashes or from water dripping down the side of the tube. For a detailed description of the process which crops snowflakes out of the images, see Appendix D and the corresponding flowchart in Appendix E. An example of this cropping can be seen in Figure 8.

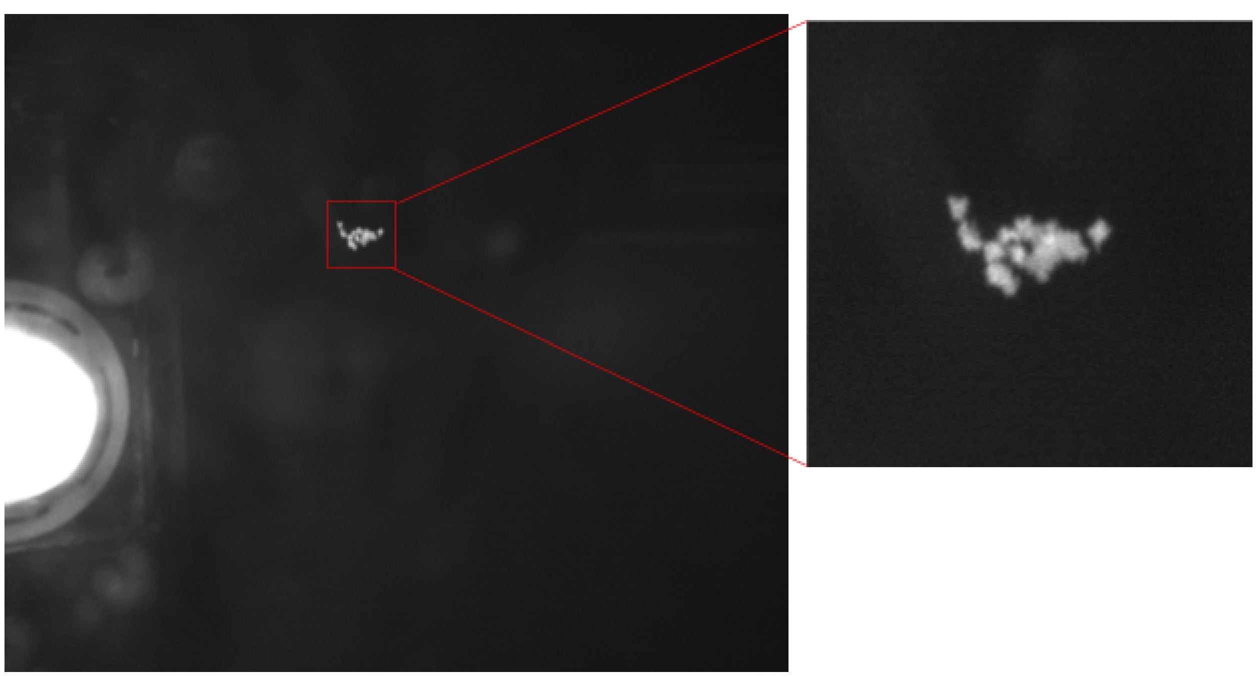


Figure 8: An example of what it looks like to crop the snowflake out of the larger image.

CNNs are a kind of supervised learning, which means they require a dataset of labeled training data and the larger the dataset, generally the more accurate the CNN can become. As the project progresses, there will be a need to add more and more images to the dataset used to train the image classification model in the hopes of improving its accuracy. There is not enough time in a single semester to label hundreds of thousands of images manually, so instead time was invested in making sure labeling the images in the future could be done relatively quickly, easily, and accurately. To that end, a relatively simple graphical user interface which can be seen in Figure 9 was developed. It will allow future groups to easily adjust the classes they would like to label the images with as well as then label those images relatively easily.

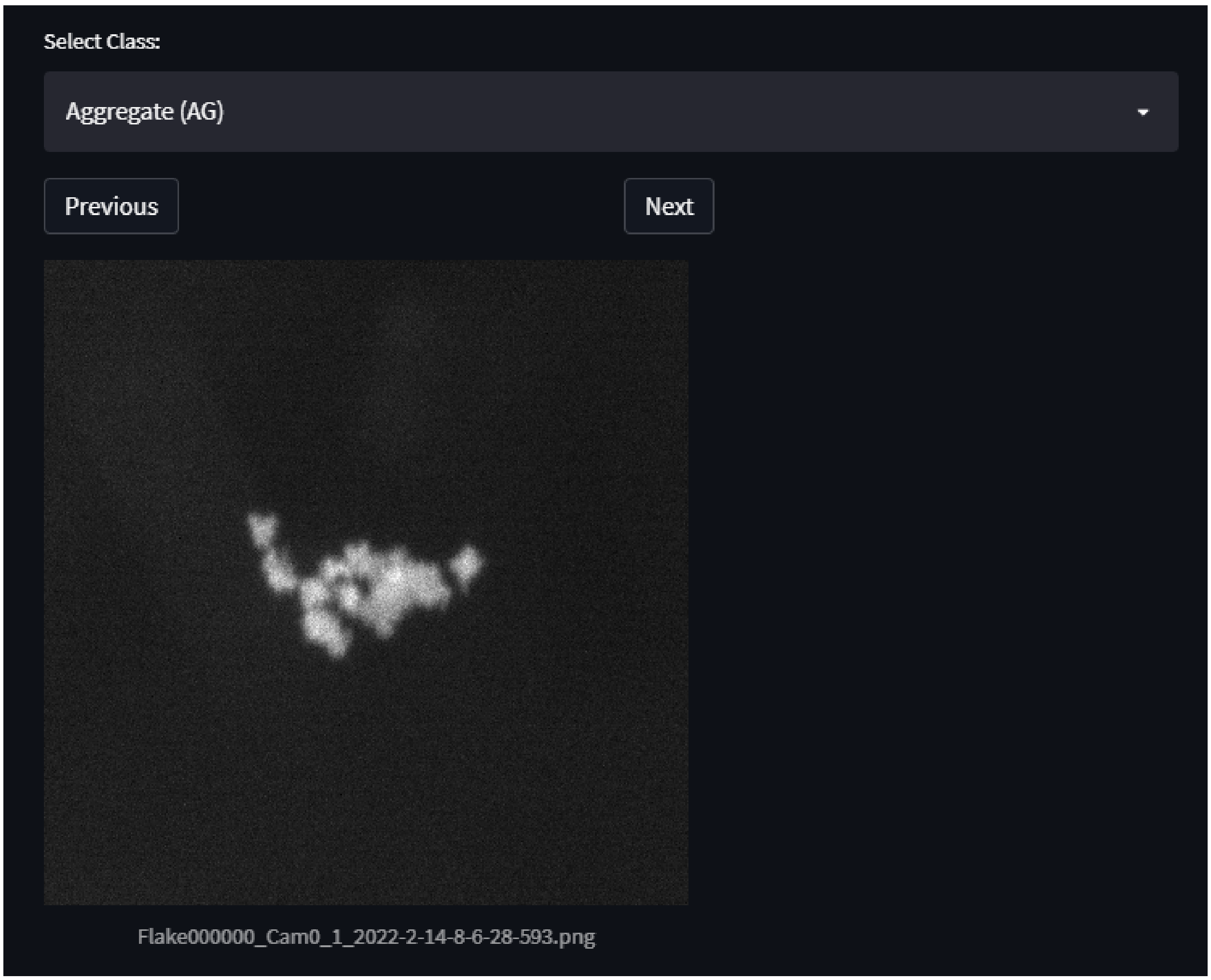


Figure 9: A screenshot of what the labeling user interface looks like.

Another goal of ours was to accurately record the fall speed of each snowflake passing through our system. Knowing the fall speed provides insights into the density and riming degree of the snowflake and helps classify it more accurately. To find the fallspeed of each flake, we take two pictures in quick succession from the same camera, a regular and a long exposure shot shown in Figure 10. By comparing the length of the streak with the size of the snowflake and knowing the shutter speed of the long exposure shot we can derive the apparent fall speed. However, this is a naive approach that only works with single snowflakes and most of our images have multiple snowflakes. To make this work for all flakes we devised a three step method of matching snowflakes from the first image with streaks in the second image.

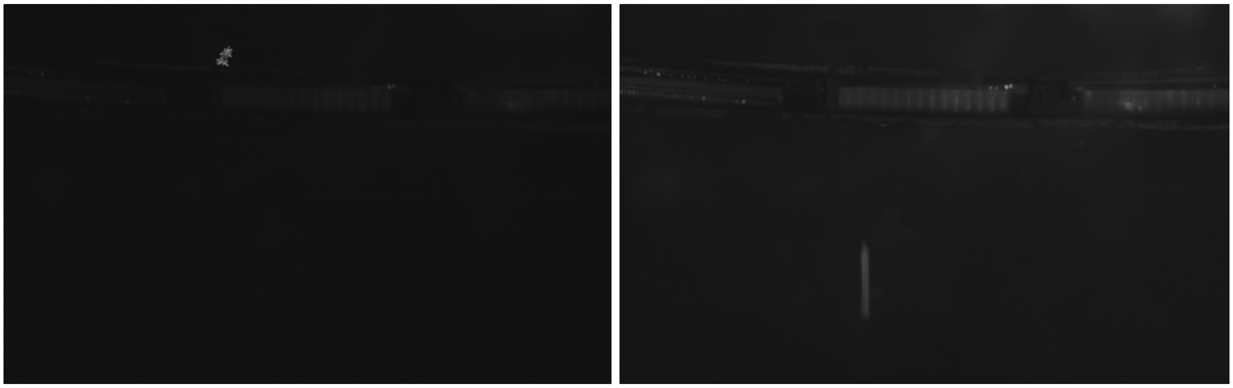


Figure 10: An example of a matching pair of short exposure (left) and long exposure (right) pictures of the same snowflake.

The first step is to identify all of the snowflakes in our first image and streaks in the second. The process of identifying snowflakes is the same as introduced above and discussed in appendixes D and E. This approach was very successful for snowflakes but not streaks because the pixel intensity of the streaks was much more similar to the background pixels in the image. To solve this problem, we trained a neural-network based object detector to identify and label all the streaks visible in an image. This involved hand-labeling hundreds of images which were then used as a training set to retrain an existing object detector architecture to only detect streaks. The pretrained object detector we used was efficientdet-lite4, a lightweight object detector designed and trained on a general dataset by Google. An example image, labeled by our object detector is shown in Figure 11.

Figure 11: Streaks labeled by our object detector in an image with dozens of snowflakes.

After finding all the flakes and streaks, the second step is to match them together as accurately as possible. This requires two algorithms, one to find the slope of the streaks and one to do the actual matching. In order to find the slope of the streaks, we process each detected streak individually and perform edge detection to get a clear outline of the streaks' shape. Next we sample the streak at different points along its height to find the middle, from this collection of midpoints we find the line of best fit for the streak which is analogous to the slope. For each streak we save the slope and the top midpoint which represents an intersection point for the line defined by the slope, for each flake we save just the midpoint of the snowflake. The slope represents the direction of travel for that snowflake so an imaginary line with the found slope and crossing through the saved intersection point is a one-dimensional representation of the space we would expect to find the corresponding snowflake in. The matching algorithm takes these lines and the points representing the snowflakes and uses the line-point distance theorem to create a distance matrix representing the distance between every snowflake and every hypothetical line. Using this matrix to find the optimal pairing of flakes to streaks gives us a highly accurate guess as to which snowflake corresponds to which streak.

The final step of finding the fallspeed is to actually find the fall speed of each flake in the image individually. This is done using the original approach of comparing snowflake size with the length of its streak in the long exposure picture.

4.3 Mechanical

One of the major data collection goals for this project is to determine the actual fall speed of the snowflakes that are cataloged. In order to do this, we needed a control video with a small object moving at a known speed. Another major focus of this semester was to determine what camera angles would be useful and also determine whether mirrors would be a viable solution to providing more angles of the snowflake without increasing cameras. We attempted to tackle both of these problems by building one rig.

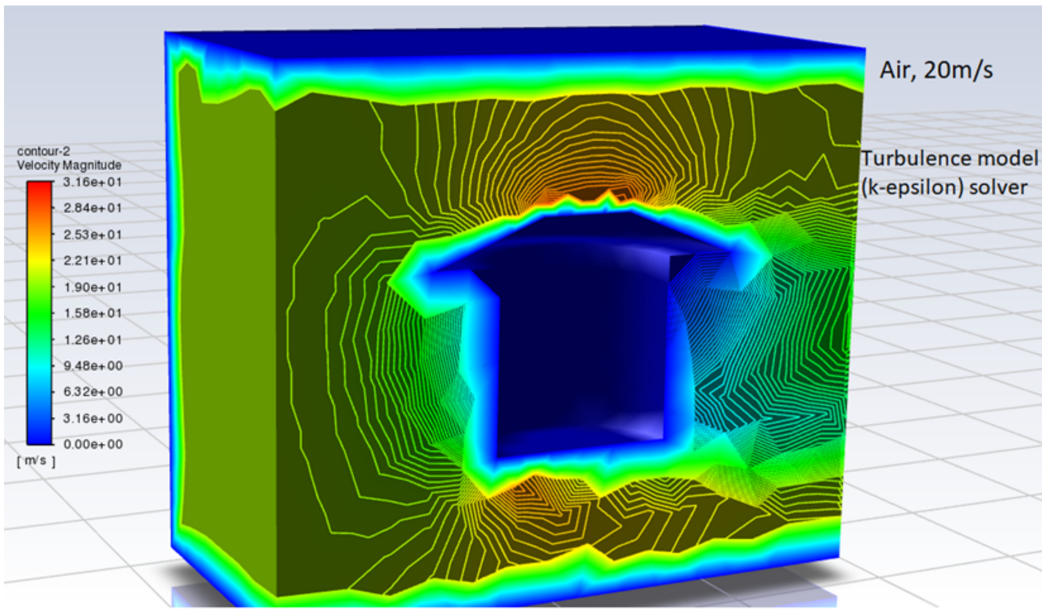
Figure 12: This is the device created to test fall speed detection along with camera placement.

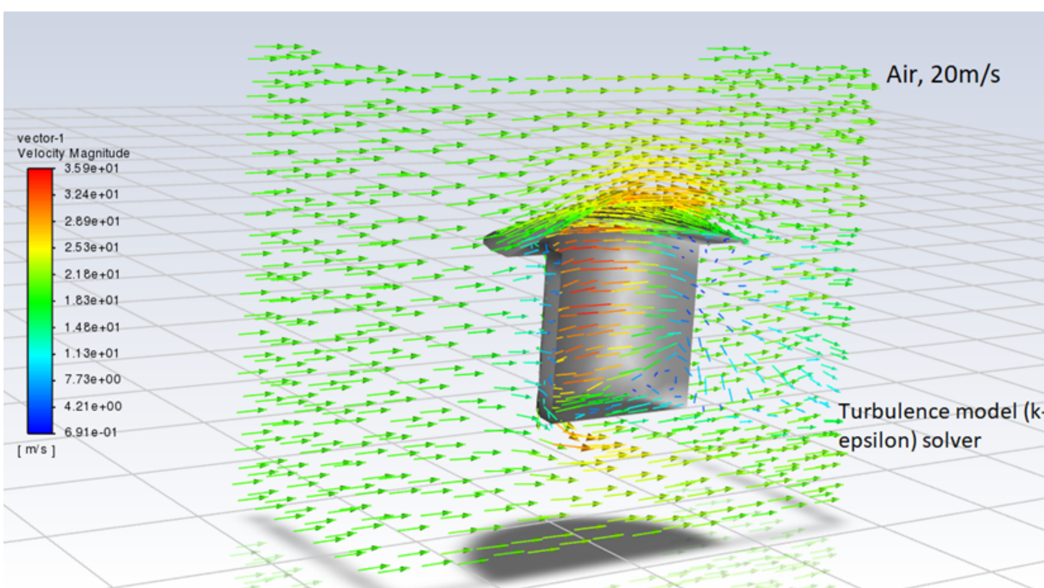
Figure 12 above shows this device; the pink and white ring is designed with a customizable mount that allows us to use this set up to hold both a camera and a set of mirrors on opposite sides of each other. The servo motor is attached to a pole at the top of the device so it hangs right over the center. Then a fishing line can be run through the track on the wheel and move a small white object (we used a piece of cotton) at a known velocity. This device is close to complete, however the 3d printed ring is not currently capable of the friction required to move the ring without slipping.

The mirror setup was finished and is ready to be tested for picture quality. The camera can be mounted on one side and the mirrors across from it, and the fishing line can be used to suspend a small enough object to test detail. This device was finished too close to the end of the semester in order to complete adequate testing; however, we are excited to hear of the results that come from it from next year's team.

One of the functions of the device is to measure the snowflake fall speed. It is important to notice that with the design of our device, as wind blows across the top and bottom of the tube where snowflakes fall into, turbulence can be created and disturbances will be introduced. As our cameras and sensors are all inside the system, at the center of the tube, we can only measure the apparent fall speed of the snowflake. To be able to capture the actual fall speed of the snowflake, we are taking the approach of measuring the disturbance. We are planning to use the apparent fall speed and measured disturbance to accurately acquire the actual fall speed of the snowflakes. To capture the disturbance, I have conducted Computational Fluid Dynamics Simulations.

First, I modeled a 3D simulation with an ambient air speed of 20m/s, as shown in Figure 13 below.





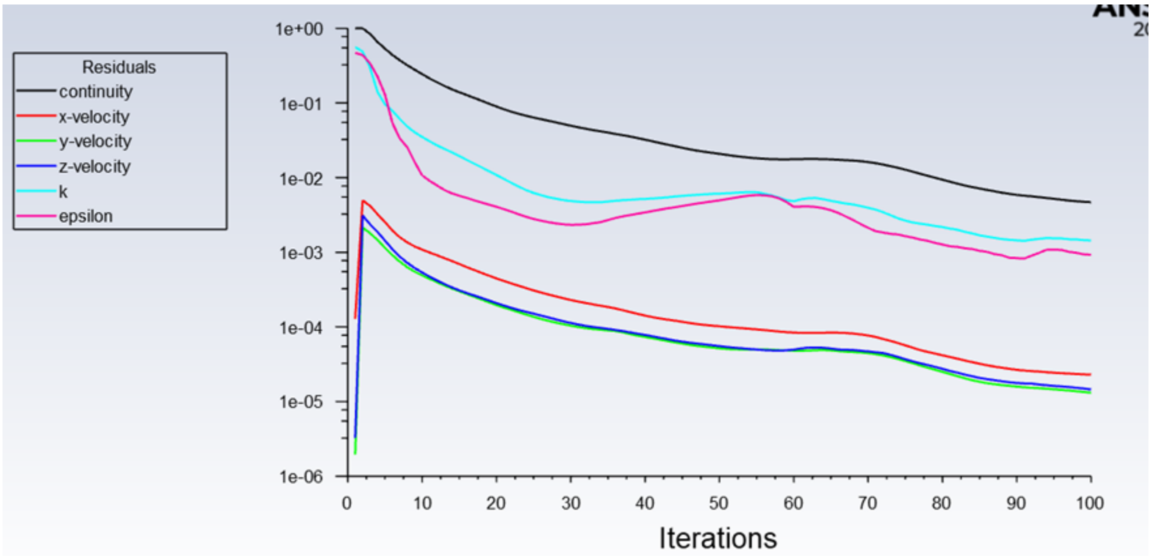


Figure 13: CFD simulation and results with 20m/s ambient air speed

The velocity vector and contour are shown to help understand how the wind speed varies across the device. The residuals plot is also shown to ensure that the solution has converged and the results are accurate. From the velocity vector figure, we can see that the maximum wind speed across the top and bottom of the tube is around 27m/s, keep in mind that this is caused by an ambient wind speed of 20m/s in the x-direction.

A 2D model was then conducted with the 27m/s wind speed across the top and bottom of the tube. The results are shown below.

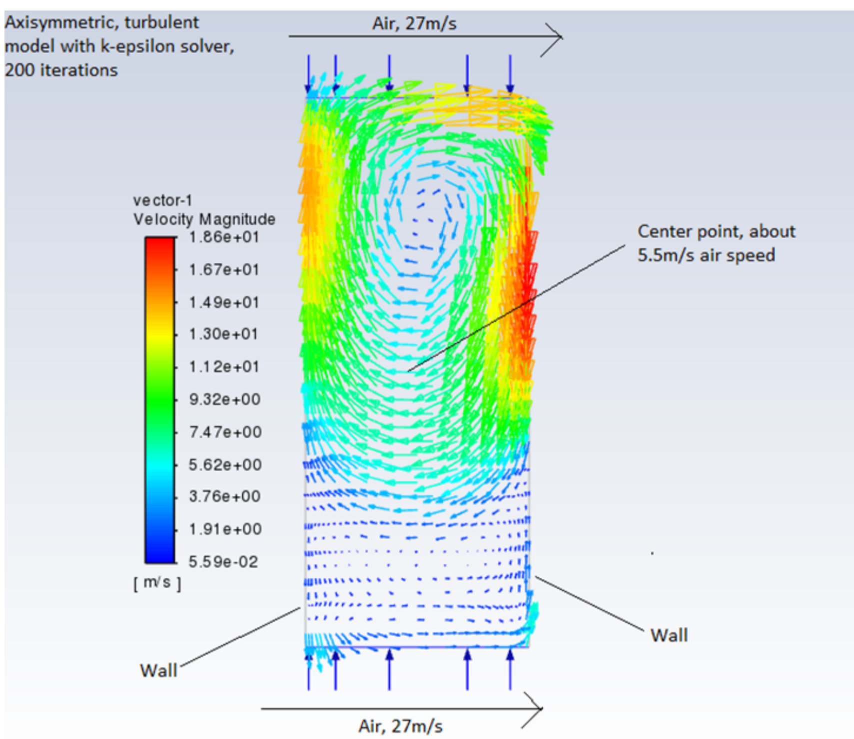


Figure 14: CFD simulation with 27m/s ambient air speed

It can be seen that based on an ambient wind speed of 20m/s, about 5.5m/s of disturbance is created at where the snowflake fall speed will be measured by the cameras. It is important to note that the simulations conducted were based on a turbulence model. Turbulence or laminar solver needs to be used according to different ambient conditions. To date, the team still has not found an exact solution as to how to capture the actual fall speed of the snowflake based on the apparent fall speed and disturbances found. Thus, this work will be carried forward to the next semester.

To improve and solve this issue, the team has thought about different designs to reduce the disturbance caused by the device itself. For instance, the team is looking into a more skeletal structure, as seen in the figure below:

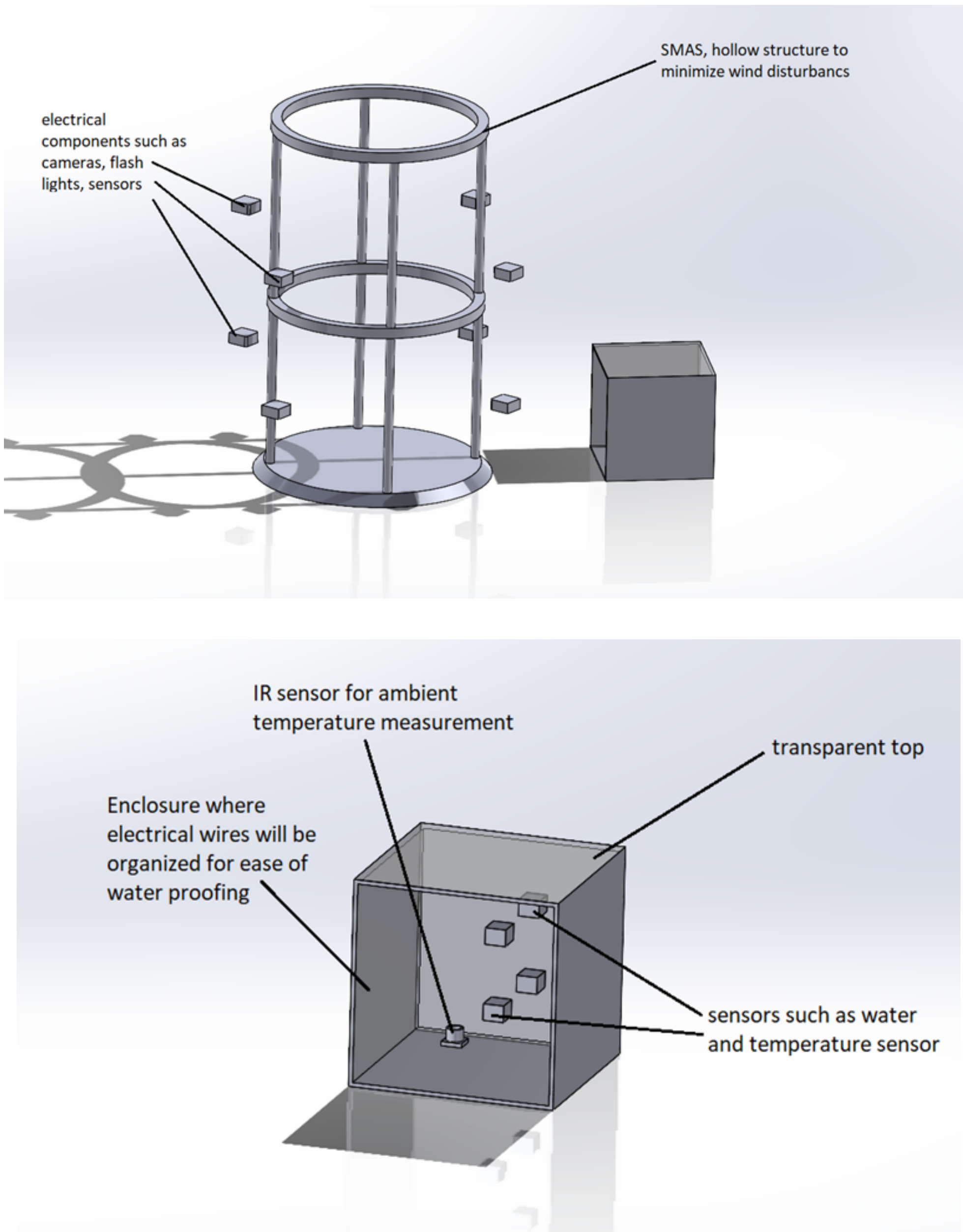
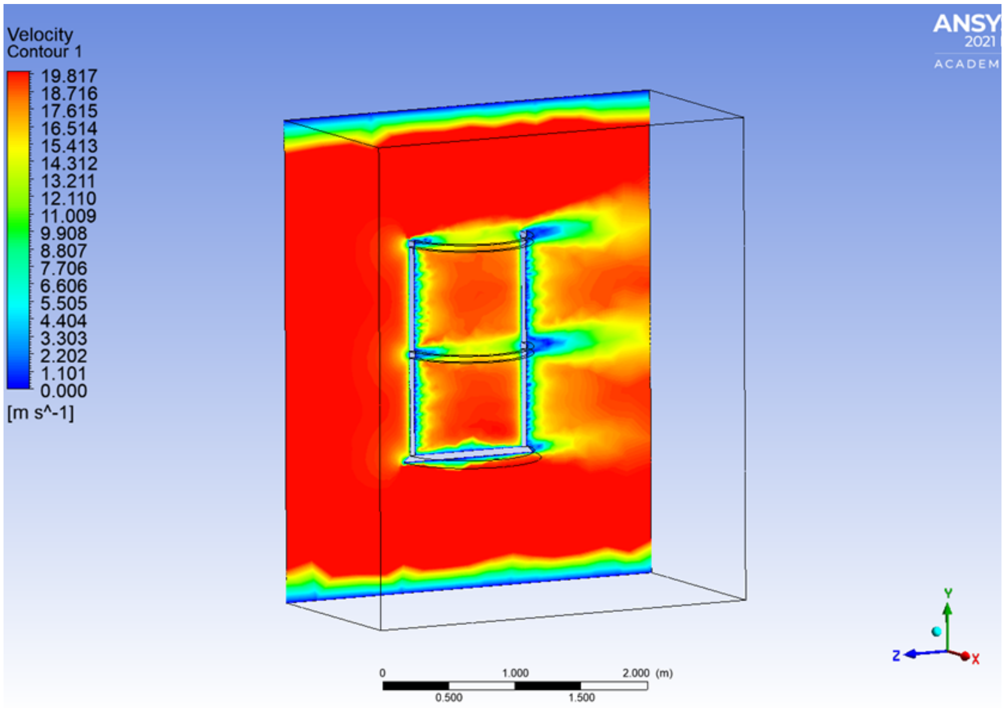


Figure 15: Proposed revision to SMAS design

CFD simulations have also been conducted to investigate if a hollow structure will create less disturbance, and the results are shown below. It can be seen that at the center, we have less disturbance compared to the previous design.



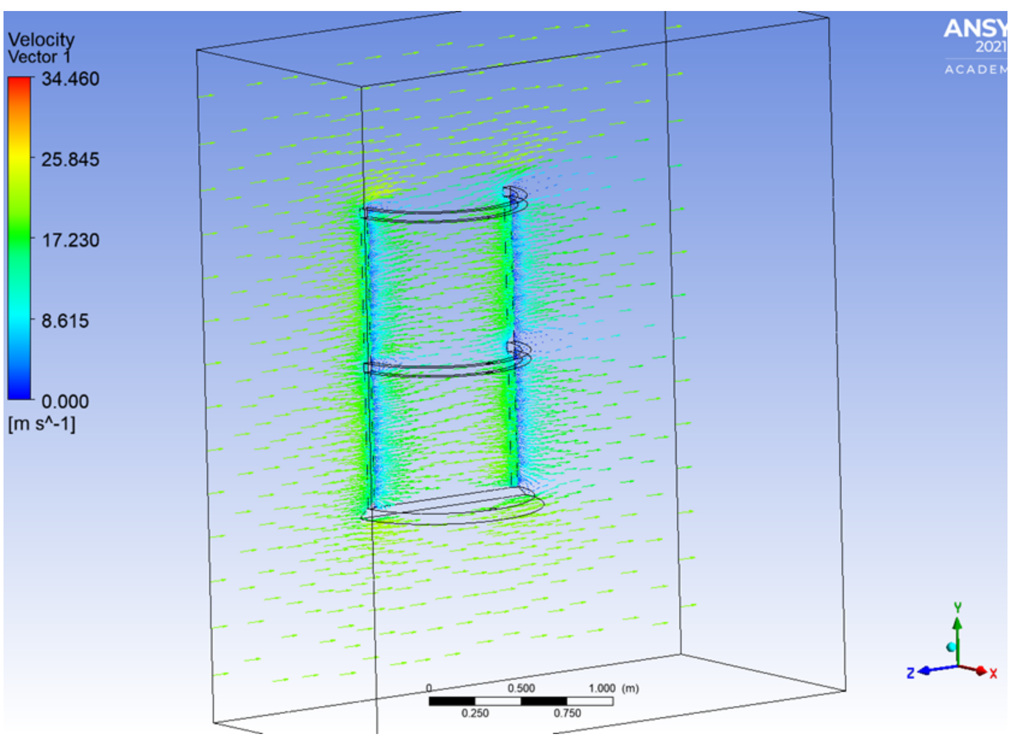


Figure 16: CFD simulation on new model

5. Standards

The SMAS abides by the Ingress Protection Code, otherwise known as the IP rating. This rating is set to determine how well a device can withstand dust and water elements. As SMAS has to be mounted outdoors to collect data in extreme weather conditions for long periods of time, our team decided that an IP rating of X-5 will be a good goal. Note that an IP rating usually comes with two numbers, such as IP32 or IP46. The first number denotes how well the device will fare against dust and the second number is for water resistance. Since our device doesn't get affected by dust significantly, we used most of our time to achieve the water resistant goal.

To ensure that the device is IPX5 rated, we had to run multiple water tests. An IPX5 water test procedure includes spraying the device with low water pressure from all angles. The water flow has to be at least 10 liters per minute. We conducted the water test by using a water hose at the engineering building and tested the water flow with a timer and a container with known volume. To observe the slightest amount of water entry, the team attached cobalt chloride strips at all potential water entry points. The figures below show the cobalt chloride strips used along with the water test conducted.

Figure 17: Cobalt Chloride Strips

Figure 18: Water test conducted by a team member in front of the engineering building

The first two water tests were a failure as some cobalt chloride strips turned pink after the water test. The team had to further waterproof the device after each failure. However, the device passed the third water test successfully.

6. Conclusion

During the Fall semester, the Snowflake team has worked diligently to finish the Snowflake Measurement and Analysis System (SMAS). It was not an easy goal to complete and demonstrate a system that could autonomously capture snowflake images, reconstruct 3D snowflakes, measure fall speed and classify snowflakes. Coming into the Fall semester, the team knew being very proactive and determined was pretty much a standard as the system needed a lot of improvement. On top of that, the hard deadline to ship the SMAS to NASA on November 8th was a constantly looming reality check. However, through the effective management skills of our project lead and the determination of the senior design students, the team was able to meet the deadline successfully.

After the SMAS was shipped on November 8th, the team diligently began researching and experimenting with ways to improve the next iteration of the SMAS all throughout the Spring semester too. This research involved new sensing techniques, new camera configurations, new mechanical designs, and a more solidified SMAS design as a whole. Though most of what was researched in the Spring semester is still unfinished, there is definitely a clear path forward for the next senior design team to work on this project.

Regardless of how intimidating this project first looked, our team was happy to work on it and are happy with what we have accomplished thus far. The experience was unforgettable and future teams will no doubt have just as memorable and exciting experiences as this team did!

7. Future Work

Having completed the SMAS project, the senior design students are aware of its current flaws and what could be further improved. As this is a huge project with a lot of different components, we have separated our future work into three sections: Electrical, Computing and Mechanical.

7.1 Electrical

Though the current SMAS system does function electrically just fine, there were numerous aspects that could use plenty of improvement in the near future. Specifically, the PCB that was used to control the cameras and lasers needs to be redesigned, the CAN Bus system needs to be redesigned, and better cameras might need to be considered.

Regarding the PCB, there are numerous components on the board that no longer perform any function in the current SMAS. This means that these components should be taken off the board completely to allow for a more concise and less complicated PCB. Additionally, the camera triggering circuit will most likely change anyways once better cameras and lasers have been selected. So either way the board will need to be redesigned at some point in the future. Unfortunately, this task has had no progress this semester as the SMAS is still at NASA, so it remains a lingering objective for future teams.

Even with a new and improved design for the SMAS, it could never be a truly complete, robust system if there was not a way to monitor the system's vitals reliably. The CAN Bus that was designed for the current SMAS was severely lacking in multiple departments, the biggest issues being absurd packet losses, UART vs CAN Bus protocol transmission speeds, and general reliability issues. A lot of design work has already gone into the CAN Bus for the next iteration of the SMAS, as described in section 4.1; however, more design and implementation work is going to be needed since the MCP2515 module did not function as needed before.

In addition to the CAN Bus and PCB, the selection of new cameras needs to be decided in the near future for the next rendition of the SMAS. The current cameras are impossibly complicated and perform many extra functions that this project does not need. Though the camera's resolution and triggering mechanisms work well for this project, it may be best to find cheaper cameras with maybe less complicated features that still perform to the levels that this project needs. More research into this matter is definitely needed in the very near future.

Another important feature that still needs more progress is the remote sensing techniques described in section 4.1. Specifically, the doppler radar module for snowflake density measurements and the lidar for measuring humidity and temperature. Though a plentiful amount of work has been done on these topics, more research and experimentation is needed to construct an actual working remote sensing system.

7.2 Computing

Moving forward, there are several things to be improved on including polishing the website to be more than a barebones skeleton, reworking the codebase and including documentation for the project's posterity and creating a robust source control system using something like GitHub, and transitioning the project from being based on a Windows device to being based on a Linux device. We also hope to make significant progress with making a working machine learning workflow to classify the snowflakes we image.

Currently, the web interface is locally hosted and has just a few of the simplest buttons and some floating images and text. We hope to develop it further not only to increase functionality, but also to make it more usable and more visually appealing. For example, we'd like to include a subpage where the user can visually inspect some of the most recent images (more than just the most recent 7) and place them in some sort of image viewing component so that the user can easily navigate between images instead of simply scrolling down until they get to the image they want. Additionally, we hope to make the website available online so that SMAS users can interact with it without being on the same local network as the device.

The code which saves the images and adjusts the camera's characteristics is very difficult to interpret and is filled with various debugging print statements which clutter the device's output. In the upcoming semester we plan on either completely rewriting the code to work with new cameras or at least heavily modifying the existing code to get rid of the extraneous debugging print statements and add a great deal of documentation so that future generations of this project can more easily understand what the code does. Additionally, there is currently no established source control, which means when someone is iterating on the code and perhaps makes a mistake which reduces its functionality it becomes very difficult to go back to a previously working state. By creating a strict policy for using a tool like GitHub, this project will be easier to share with future groups and they will have an online trail of various edits that have happened throughout the years.

While Windows is a very popular and widely known operating system, we intend to transition away from having one central computer which is big and bulky and instead do more data processing on distributed, smaller, Linux-based systems which then communicate to a smaller, also Linux-based computer. Then we will be able to leverage the many advantages Linux has over Windows for a project like this, including more open source libraries and a greater degree of control over various processes and subprocesses.

There is existing research into the classification of snowflakes using Neural Networks [1]. We hope to leverage that research and apply it to images taken with the SMAS instead of images taken with the MASC, where we expect to see increased classification accuracy due to the additional camera angles provided. More image processing for feature selection might also improve classification accuracy. Professor Azimi-Sadjadi has offered to become a co-advisor on this project in return for his help working with future students to develop various holistic approaches to image processing so that they can be more easily compared. Many of the kinds of image processing which would be useful to a project like this one are beginning to reach into graduate student territory, so Professor Azimi-Sadjadi's expertise could prove particularly useful. Additionally, we plan on developing a system which will use that neural network model and classify snowflakes in real time as they are imaged. This may require use of specialized hardware accelerators for neural networks, which we would implement on the Linux system. Perhaps most immediately, however, it will be important to label many of the images that have been collected. Without labeled images, it is extremely unlikely that an image classifier will achieve high levels of accuracy but it may also be worthwhile to look into unsupervised learning in the form of k-nearest neighbor clustering and even hybrid approaches in order to minimize the amount of labeling that is necessary.

7.3 Mechanical

One of the biggest flaws that the SMAS has is that it is extremely compact and parts are non-interchangeable. Being a very compact device, the SMAS needs to be handled gently to avoid any unintentional damage. This becomes a huge problem for shipping inside an aircraft or a vehicle. The amount of wires and cables inside the device are also a nightmare. The team has spent at least 80 hours in cable management to significantly increase the neatness of the cables. The SMAS has all of its electrical components attached to its main body. This causes concern for ease of damage when the device is being transported or moved.

With that in mind, one of the most important future work is to redesign the system for robustness and interchangeability. Whereas the previous teams first designed the device and then the mounting frame, we are planning on first designing the mounting frame. We would like the mounting frame to be the “chassis” of the device. This design would be much more reliable as the robustness of the device will be improved by a considerable amount. Apart from that, having most components attached to the mounting frame also allows us to swap out faulty components more easily. The cables would also be managed more appropriately as we will have more room and clear paths for the cables to run through.

Another aspect of future mechanical work is directed towards testing of the new iteration of the SMAS. One of the important things that will be tested is the reliability of the mounting frame. This means running Finite Element Analysis to determine the maximum strength of the frame and Computational Fluid Dynamics Simulations to observe its stability. We’d like to have a design that is efficient in terms of minimizing weight and maximizing strength.

Apart from testing the mounting frame, we would also test the image capturing capabilities of the new iteration of SMAS. To do this, the team plans to 3D print tiny snowflakes to simulate actual falling snowflakes. This is a challenge as it is not a simple task to 3D print extremely small objects with detailed features. We also plan to create a camera fixture that could hold up to 6 cameras at a time and allow them to move with precise specified distance. The function of this fixture would be to allow experimentation with different camera positions and how well the images can be captured and 3D reconstructed with each different camera position.

References

[1] Hicks, A., and B. M. Notaroš. (2019) Method for classification of snowflakes based on images by a multi-angle snowflake camera using convolutional neural networks. *Journal of Atmospheric and Oceanic Technology* 36, no. 12: 2267-2282.

[2] “Raman Lidar and Theoretical Background” Accessed on: May 4 2022, [Online]. Available: https://www.herts.ac.uk/research/centres/cacp/atmospheric-remote-sensing-laboratory/instruments/raman-lidar-theoretical-background

[3] Jussi Leinonen, et al, “Retrieval of snowflake microphysical properties from multifrequency radar observations” amt.copernicus.org, https://amt.copernicus.org/articles/11/5471/2018/, Accessed on: May 4, 2022

Bibliography

G. Heng. *GS3 U3 Technical Reference*. Accessed: Oct. 2021. [Online]. Available: https://flir.app.boxcn.net/s/msy6d75zxymvye1tfg3rcdj4f598bmyb

Appendix

Appendix A: Abbreviations

3D: Three Dimensional
CAN Bus: Controller Area Network Bus
CNN: Convolutional Neural Network
GUI: Graphical User Interface
IPX-5: Ingress Protection Rating to protect against sustained low pressure water stream from any angle
LED: Light Emitting Diode
MASC: Multi-Angle Snowflake Camera
MOSFET: Metal-Oxide-Semiconductor Field-Effect Transistor
NAND: AND NOT logic operation
NASA: National Aeronautics and Space Administration
NSF: National Science Foundation
PCB: Printed Circuit Board
PNG: Portable Network Graphics
PMXX: Particulate Matter inhalable particles XXmm in diameter
SMAS: Snowflake Measurement and Analysis System
SMD: Surface-Mount Device
SSD: Solid State Disk [Drive]
SSS: Snowflake Sensing System
TTL: Transistor-transistor logic
UART: Universal Asynchronous Receiver-Transmitter

Appendix B: Budget

The budget for this project was relatively loose, but still well-defined. Each Senior Design member was allocated \$200 towards the project from the ECE Department. The project was also funded by the National Science Foundation (NSF) with an undefined amount. The objective at the beginning of the project was to deploy the SSS by the November 8th deadline by any means necessary. It was very important that the device was deployed during the winter since it is the only time of year that we can collect real-world data. The team worked hard from June-November and purchased necessary hardware to add to the project and to replace failed components. Below is a detailed breakdown of the budget including product, date, vendor, cost, and running expense total.

Income:

Source of Income:	Description:	Amount:	Running Total:
Stipends	The \$200 stipend per	\$600	\$600

	ECE senior design student. There are three ECE seniors which means a total of $600.		
NSF Grant	Money allocated towards the project from the NSF grant.	Undefined	$600+

Cost:

Materials Needed:	Purpose/description:	Quantity:	Estimated Cost:	Vendors:	Running Total Cost:	Purchased Yet?:
Zip Ties	For cable management	1 pack	$8.97	Home Depot	$8.97	Yes
Power Supply	For our broken down computer that we need back up and running	1	$69.85	Ebay.com	$78.82	Yes
Motherboard	Again, for our broken down computer	1	$80.61	Ebay.com	$159.43	Yes
3D Printing Filament	Filament for 3D printing camera mounts, laser mounts, etc	1	$20.99	Amazon.com	$180.42	Yes
Nema-17 Stepper Motor	Stepper for our fall speed calibration tests	1	$8.99	Amazon.com	$189.41	Yes
A4988 Stepper Motor Driver Arduino Shield	Stepper motor driver for the Nema-17 that sits on top of an Arduino	1	$17.89	Amazon.com	$207.30	Yes
Electrical Tape	For insulating wire	1	$1.97	Home Depot	$209.27	Yes
Velcro	For cable management and light mounting	2 packs	$3.96	Home Depot	$217.19	Yes

Acetone	To dissolve super glue	1	$5.84	Home Depot	$223.03	Yes
Surge Protector	To protect device from power surges	1	$39.77	Home Depot	$262.80	Yes
Raspberry Pi	For CANBUS	1	$68.99	Amazon	$331.79	Yes
Arduino Mega	For controlling the actual sensors	2	$31.98	Amazon	$363.77	Yes
Jumper wires	For connecting all of the sensors	1 pack	$7.51	Amazon	$371.28	Yes
Rain sensors	To detect water leakages in our system	15	$20.97	Amazon	$392.25	Yes
Raspberry Pi power cable	To power the Pi correctly, current supply isn't strong enough	1	$10.70	Amazon	$402.95	Yes
CP2102 Module	To interface the RPI with a computer	1	$5.39	Amazon	$408.34	Yes
Micro SD card	For RPi	1	$8.39	Amazon	$416.73	Yes
Uninterruptible Power Supply (UPS)	Regulates the power going into the device to help prevent damage from surges and maintain performance during short power outages	1	$201.45	Home Depot	$618.18	Yes
2 TB SSD	For mass image storage	1	$174.25	Amazon	$792.43	Yes
Extension cord	For power supplies	1	$11.19	Walmart	$803.62	Yes
IC7400 and thru-hole pcb	For crossplane lasers	1	$7.12	Mountain States Electronics	$810.74	Yes

Erector sets	For mounting the SMAS system	1	$75.68	80/20 website	$886.42	Yes
Nuts & Bolts	Connecting mounts to SMAS	4 packs	$20.56	Ace Hardware	$906.98	Yes
Velcro	Connecting Hard disks to CPU mount	1 pack	$5.12	Ace Hardware	$912.10	Yes
Nuts & Bolts	Connecting IMUs to mount	4 packs	$25.78	Ace Hardware	$937.88	Yes
Tape	Sticky Paper	2 Rolls	$7.59	Ace Hardware	$945.47	Yes
Syringes	Humidity sensor testing	1 Pack	$5.99	Ace Hardware	$951.46	Yes
2x2 Wood Stud	Temporary supports for SSS	1	$9.99	Ace Hardware	$961.45	Yes
3d Printer Filament	Printing supports for flashes and cameras	2 Rolls	$40.84	Amazon	$1002.29	Yes
6mm Timing Belt	Belt to run on fall speed calibration	1	$11.50	Amazon	$1013.79	Yes
Cobalt Chloride Test paper	Water testing strips for weather testing	1	$9.67	Amazon	$1023.46	Yes
Ball Bearing	For wheel on fall speed calibration	1 Pack	$6.44	Amazon	$1029.90	Yes
4ft Grounding Rod	To ground entire system from static charge	1	$21.50	Amazon	$1051.40	Yes
8ft of Weather Stripping	To seal bottom of device	1	$11.27	Home Depot	$1067.67	Yes
8ft of 4x4 wood	To construct Crates	20	$240.00	Home Depot	$1300.67	Yes
1/2in of 4x8 OSB	To construct Crates	8	$216.00	Home Depot	$1516.67	Yes

Appendix C: Project Plan Evolution

Revision 1

Present - November 1, 2021 - Complete development and durability testing
Present - November 8, 2021 - Once functional, begin limit testing code.
November 1, 2021 - November 8, 2021 - Prepare SSS for shipping
~~November 8, 2021~~ - Device ships to Virginia
Present - November, 2021 - Durability testing of MASC (Currently waiting for it to be returned from repairs)
November, 2021 - December, 2021 - Finalize documentation for first SSS model
November, 2021 - March, 2022 - Conceptualize and begin development of revised SSS
March, 2022 - May, 2022 - Finalize documentation for SSS and prepare for E-Days

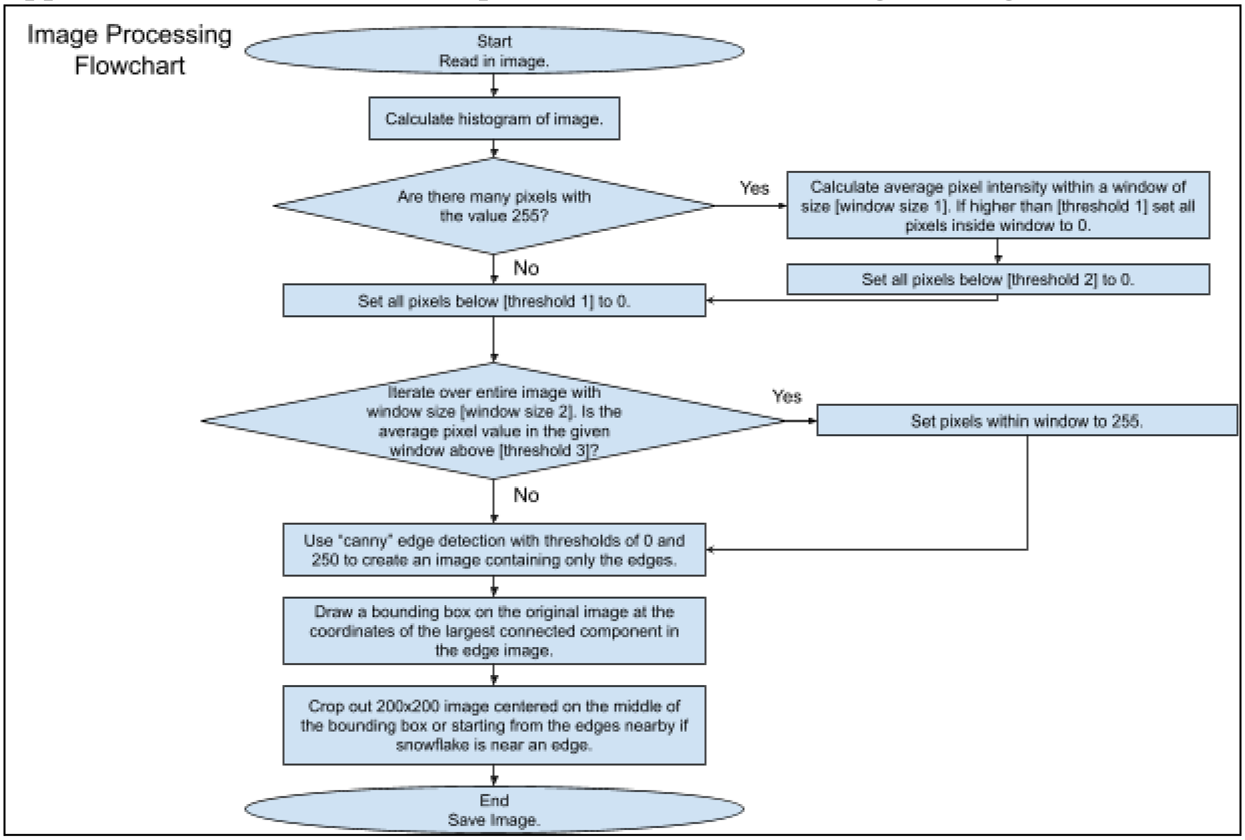
Revision 2 (up to date)

Present - December 17, 2021 - Finish mid-semester report
Present - March, 2022 - Collect data with SSS
Present - January 18, 2022 - Research and order new components
January 18, 2022 - April, 2022 - Design and test new SSS
March, 2022 - April, 2022 - Process and classify captured images
April, 2022 - May, 2022 - Develop documentation and presentation for E-Days

Appendix D: Process to Crop Snowflakes from Larger Image - Description

First, the images are put through various preprocessing steps in order to isolate the snowflakes from background information like flashes, water dripping down the sides, or rain drops, and then to crop out a 200x200 image containing the largest snowflake visible in the image. The first preprocessing step is to calculate the histogram of the image and if enough pixels have the value 255 that is an indicator that there is a flash or glare taking up a significant portion of the image. If that is the case, the entire image is iterated over via a given window size (currently set to 400 pixels by 400 pixels) and calculated the sum of the intensities inside that window. If the average of the pixel intensity within that window is above a given threshold (currently set to 40) all the pixels in that window get set to 0. Then the entire image gets thresholded such that any pixels below a second threshold (currently set to 150) gets set to 0. At this point, large flashes and glares like those seen in images from camera angle 0 will have been set to 0 in the image. That is the end of the conditional execution based on the histogram and the rest of the process is applied to all the images regardless of their initial histogram. From there, images get thresholded so that any pixels below the first threshold (currently set to 40) get set to 0. This step helps remove dark gray things in the image like raindrops or some parts of the background. Then, iterate over the entire image with a different window size (currently set to 20 pixels by 20 pixels) and calculate the average intensity of the pixels within the image. If the average intensity is greater than a third threshold (currently set to 40) then set all the pixels in that window to 255. This will help keep entire snowflakes together as one connected component when bounding boxes are drawn around connected components. Then, use canny edge detection with thresholds currently set to 0 and 250 to create an image containing the edges of the objects in the image. These edges should look a bit like a tetris block or something from an atari game that almost completely covers the snowflake because of the previous step. Then, draw a bounding box on the original image at the coordinates of the bounding box around the largest connected component in the image of the edges. Crop the original image down to a 200x200 pixel image centered on the center of the bounding box, being careful to adjust if the center of the bounding box is fewer than 100 pixels from any edges. Save the new, cropped image.

Appendix E: Process to Crop Snowflakes from Larger Image - Flowchart



Appendix F: Additional Images

Figure 19: Completely finished SMAS