All Sky Meteor Detection

Michael Suehle | msuehle@umd.edu
Science, Discovery, and the Universe
Computer Science Major
Mentor: Elizabeth Warner | warnerem@astro.umd.edu

Background

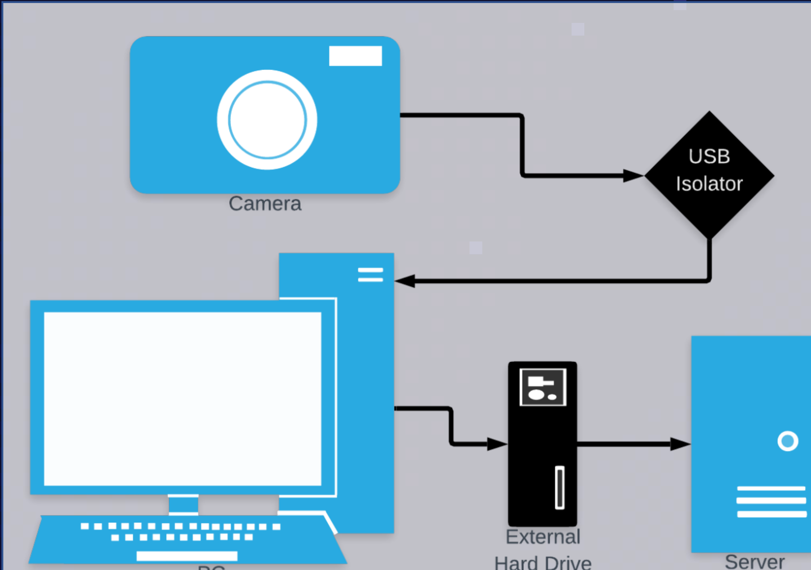
The University of Maryland Astronomy Observatory uses a network consisting of All Sky Cameras, a computer, a hard drive, and a server to maintain a database of observations in the night sky. These are available on mdallsky.astro.umd.edu. For my capstone, I helped refine the meteor detection algorithm the department uses to filter out observations with meteors for their website.

Methodology

The algorithm I worked on is written in Python and uses and open-source package package, Astride. To improve results, I modified the code so that it had masking options to help filter out things that were detected, but could not be meteors, like trees. I also tested different parameters for astride until I found ones that worked well.

All Sky Camera

photo taken from: https://mdallsky.astro.umd.edu/



All Sky Network

photo taken from: https://mdallsky.astro.umd.edu/

Findings

Masking greatly improves the functionality of the meteor detection algorithm, but too much masking makes the algorithm too selective. In a dataset of 50 images where 2 images have meteors:

- No masking
  - 1 of 2 meteors detected
  - 116 false positives
- Circle mask
  - 1 of 2 meteors detected (different meteor from before)
  - 87 false positives
- Difference mask
  - 2 of 2 meteors detected
  - 22 false positives
- Difference and circle mask
  - 1 of 2 meteors detected
  - 0 false positives

I also improved my ability to read documentation of code and learned how to use GitHub to contribute to projects.

No masking

```
streak.detect()
```

Without applying a mask, the algorithm just uses Astride's detect function on the raw fits file.

Circle mask

```
for y in range(0, len(image_data)):
    for x in range(0, len(image_data[0])):
        distance_from_cen = np.sqrt((x-x_cen)**2 + (y-y_cen)**2)
        if distance_from_cen > radius:
            image_data[y,x] = 0
```

A circle mask is applied by looping through each pixel and blacking out the pixels that are a distance greater than the radius away.

Difference mask

```
mask = np.zeros(image_data1.shape).astype(image_data1.dtype)
cv2.fillPoly(mask, rects, (60000,60000,60000))
image_data0 = cv2.bitwise_and(image_data0, mask)
image_data1 = cv2.bitwise_and(image_data1, mask)
```

The difference mask uses OpenCV to detect all the differences between two images and blacks out everything that is not different.

The same image with different masks applied

screenshots taken by me during testing

Future

During my research, I found parameters that detected meteors well, but they still detected false positives at a rate of around 10%. Ways to further improve the algorithm could include:

- finding more effective parameters for astride
- finding more effective masking techniques
- comparing the results from this algorithm with other meteor detection algorithms

In addition to this, the algorithm still needs to be implemented into the All Sky Network so that it can automatically process daily observations.

Acknowledgments

Thank you so much Ms. Elizabeth Warner, Dr. Peter Teuben, Ramsey Karim, and Dr. Geoff Ryan for all your support and help with the project! Also thank you, Dr. Peel and Ms. Thomson, for all that you do for SDU!