Dylan Winer and Jake Kasitz
Python A - Dr. Kyle Shaw
July 22, 2022

Title: Analyzing Astronomy and Light Pollution with Python

Introduction

The artificial light in the sky is measured through light pollution, which as cities continue growing, poses an ever increasing problem. Using lights that shine upward contribute to the skyglow that also prevents stars from being visible in the sky to astronomers. This also has negative impacts on the stargazing abilities for normal people as well as on ecological systems. Artificial light from cities interrupts many animals' circadian cycle, which leads to difficulty sleeping or even sleep deprivation (National Geographic Society, 2020). In our project, we try to give the user a sense of what they should expect to see in the sky. We focus on light pollution, the amount of stars, constellations, moon phases, and we output a light map with coordinate data.

The scientific problem we are solving is determining astronomical data, including the number of stars, constellations visible, moon phase, and finally light pollution from input images of the night sky. Afterwards, we also want to output the user's location on top of a map of the Earth at night to expose city lights based on the longitude and latitude coordinates provided. The season must also be inputted because the Earth rotates, so the constellations gradually shift to the west and move out of view. Now in the summer, if we might find constellations like Scorpius and Hercules, but we would not be able to find winster constellations like Orion.

Astronomy has always been interesting to us, and we would like to combine our Python skills to produce intriguing plots of the sky. Likewise, constellations and planets are more unique and their detection would show how modern technology can be used to more scientifically observe celestial phenomena first noticed thousands of years ago. It is hard to believe that gargantuan balls of gas only register to us as small dots in the sky, but by counting and observing them, we can learn more about the cosmos. There are Python websites that have used different libraries, including OpenCV, to input an image and finally detect the number of stars or constellations. However, not as much work has been done on detecting moon phases as well as analyzing the data to determine light pollution. In the citations, I have also included sources that explain how to draw constellations in the sky as well as how to identify planets based on the flickering of their brightness. Lastly, I included a source that explains light pollution to ensure we both understand our problem and how to analyze light pollution.

Methods

Our program uses a few packages to analyze images: OpenCV reads and manipulates images, Skimage provides the blob_log function to detect stars, and Colorthief finds the average color of an image. We also use Numpy to count up black and white pixels and Matplotlib for plotting maps, charts, and images. The program first greets the user with a GUI that asks them for some information about the sky. Then the program asks for their latitude and longitude coordinates, the season of year, a picture of the stars or night sky in general, and the region where the moon is illuminated (left or right) if it is included in the image. Afterwards, the program takes the information to run a variety of functions.

The first astronomical function we created counts stars by first taking the provided image, and through OpenCV, converting it grayscale. The stars are then detected by the blob_log function which finds the 'blobs' of an image by finding a region of a specified size that is significantly brighter than the region around it. The program then reads the star count on a GUI, and outputs a plot to the user where the stars were detected by highlighting orange circles around the stars.

Using the detected stars, the findAverageandDominate function then removes these stars from the image for light pollution calculations. The reason we do this is to only get a light pollution rating based on the night sky and not the density or brightness of stars. Using Colorthief, the average color is calculated from the image without stars. The program then takes this color and scores it based on a few factors. The hue of the color is measured by the getYellowness function and scored based on how close the hue is to a yellow hue. The brightness and saturation of the image is also measured. A more yellow, brighter, and saturated color will get a higher score.

The program converts this score into a Bortle rating with the get_rating function and shows it to the user. Based on the ratio of polluted pixels to total pixels, a bortle rating from 1-9 will be assigned as well as a category from "excellent dark-sky site" to "inner-city sky", based on regular intervals of 0.1 from zero to one, which is then outputted to the GUI for the user.

The next function finds what constellations are visible. Using the seasons (winter, fall…) and latitude, the program will find which hemisphere the user is located in depending on a positive or negative latitude. Then the find_constellatoins function determines which constellations can be seen in perfect light conditions from either the northern or southern hemisphere and outputs the diagrams of the constellations to the user using cv2. Then, a second function, analyze_const, is run that first classifies all available constellations into three categories: bright, middle, and low. Subsequently, the user inputs into the GUI the names of the constellations that they can see. Afterwards, the function iterates through each visible constellation and assigns a rating for night-sky quality, as dimmer constellations will be harder to see and therefore illustrate less light pollution. Ultimately, the quality of the constellation observation is compared to the original bortle scale light pollution and returns whether those values match or differ.

The last item the user inputted, the location of the moon's illumination, is used to calculate the moon phase seen in the image. Using the selectROI tool, the user can crop the image to a square around the moon and then convert to black and white. Then, the program eliminates the background noise like stars by applying a GaussianBlue and then takes the ratio of white pixels to black pixels to calculate the phase. The side of the moon's illumination is also taken into account to assign whether it is a waxing, waning, 1st, or last quarter phase.

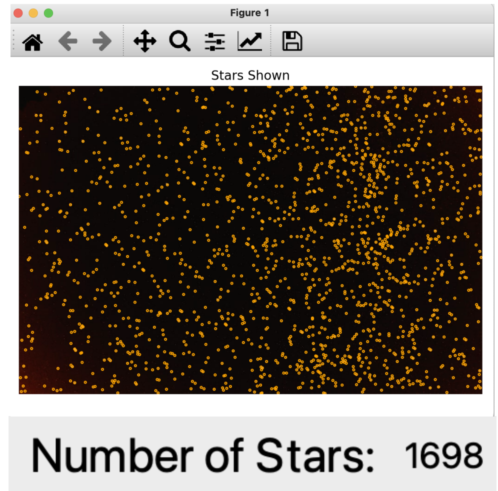
The final function of the project takes the longitude and latitude coordinates that the user gave, and using Matplotlib, plots it on a map of NASA's Blackmarble png. The Blackmarble image shows the light that is seen from space that radiates from cities. The more light seen from space, the greater the light pollution, allowing the user to analyze whether their readings for light pollution align with their proximity to city centers.

Results

We originally planned on identifying planets based on video data, but we transitioned to observing moon phases as that was more achievable within the time frame. Likewise, we initially desired the program to recognize constellations, but we switched to allowing the user to detect them for more accurate results. Furthermore, finding the light pollution was quite challenging because bright stars in the sky or camera settings can make great skies with little light pollution appear to have some light pollution from within our program, which we finally adjusted by calculating mean values for the whole image.

Star Counting Results:

Original Image

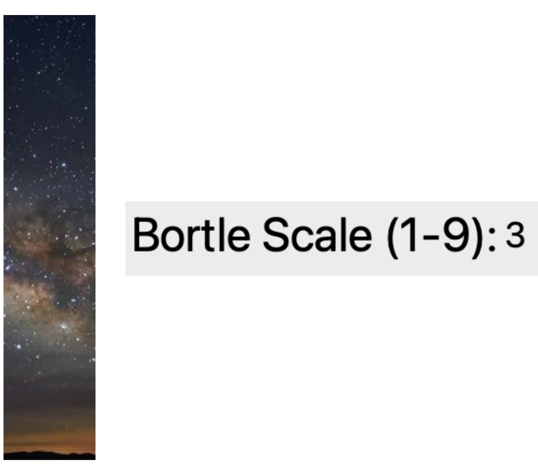


Number of Stars Output

Original Image

Number of Stars Output

Detecting Light Pollution Results

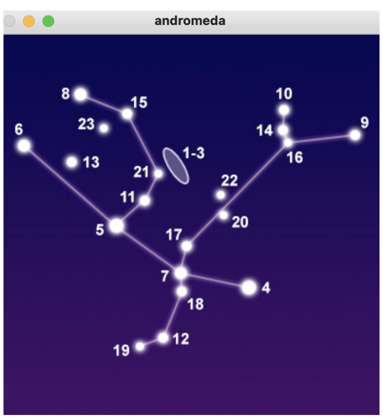


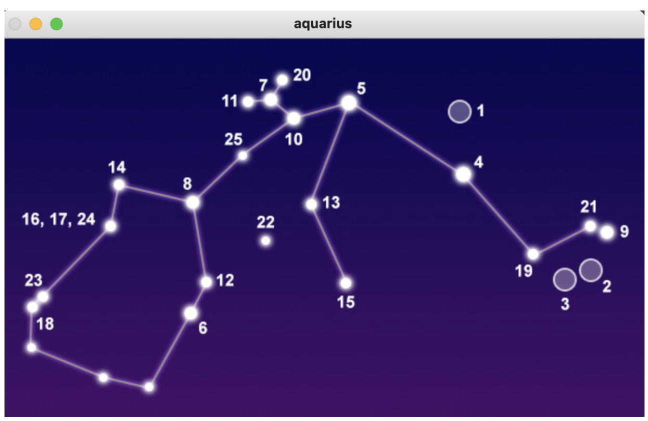
Bortle Scale (1-9): 6

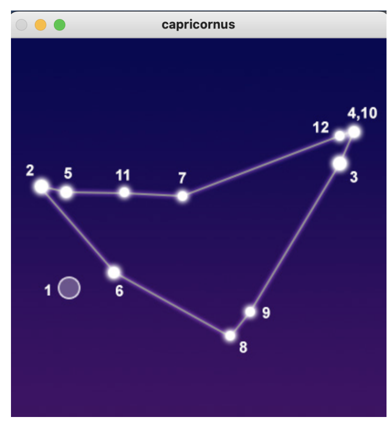
The brightness of the above image, especially in the lower half of the image, likely resulted in the higher bortle scale rating. Overall, the program can sometimes be confused by bright space phenomena like bright clusters in the sky, which might lead to slightly higher bortle scales than they would be otherwise.

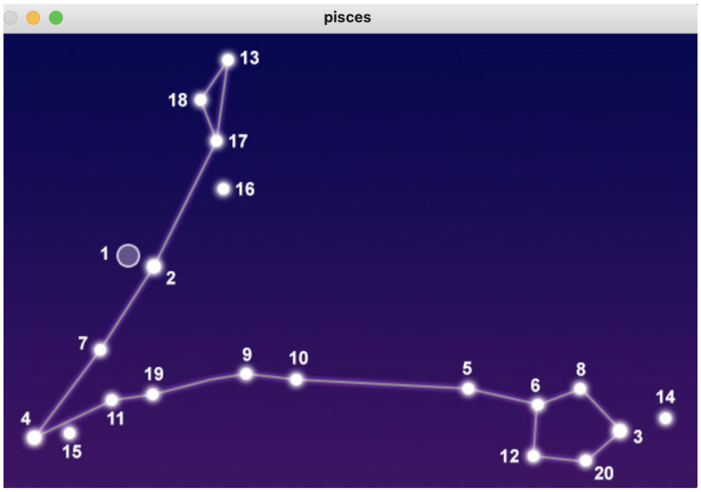
Constellations Results

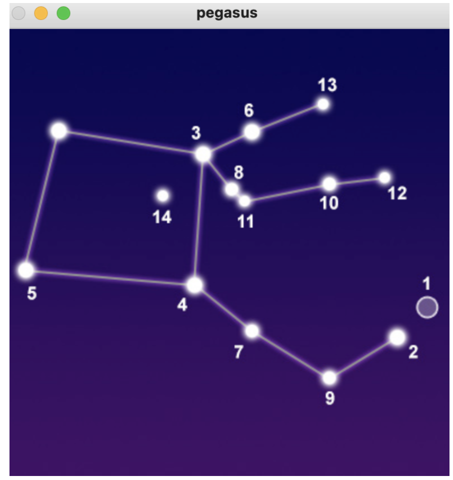
These are the constellations outputted for a southern hemisphere during the spring:



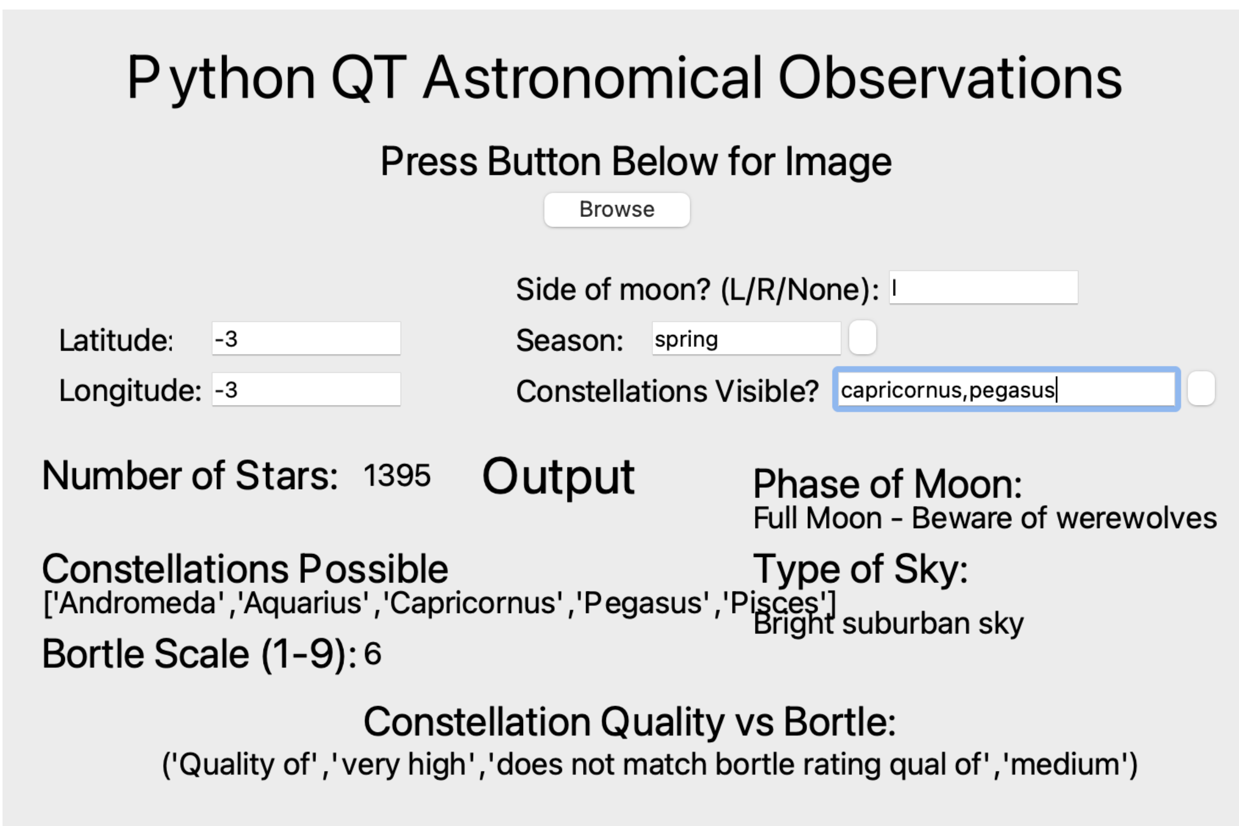






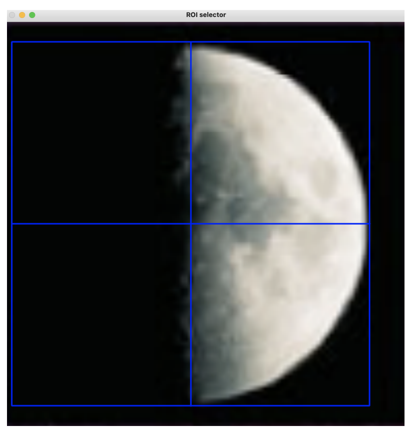


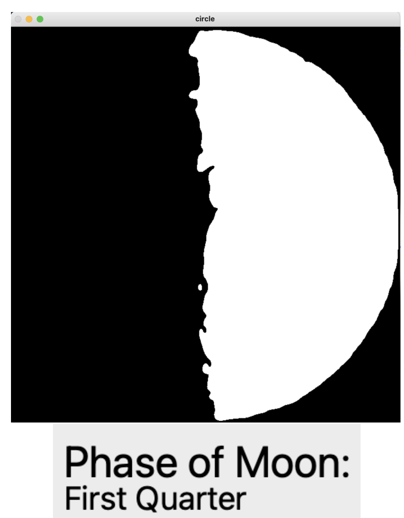
If the user then inputs that they only see capricornus and pegasus, then the program outputs:



The quality does not match because the bortle scale of 6 for this below image is only a medium quality, while being able to see the dim capricornus constellation gives the quality of maximum quality of 3/3. Because these two values of bortle and constellations do not match, the program outputs they do not match. The below image was used for the above test.

Moon Results:

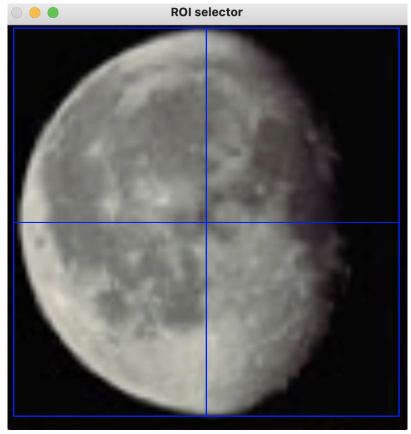


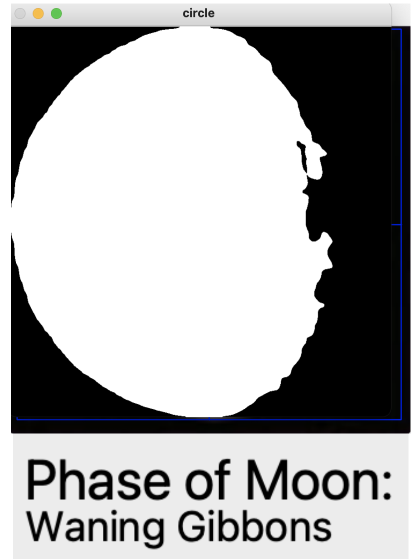


Input Image

Crop Moon

Output

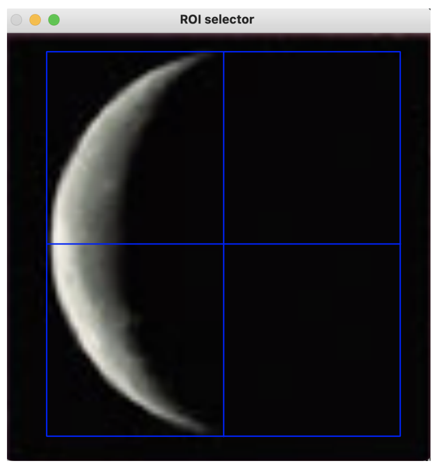


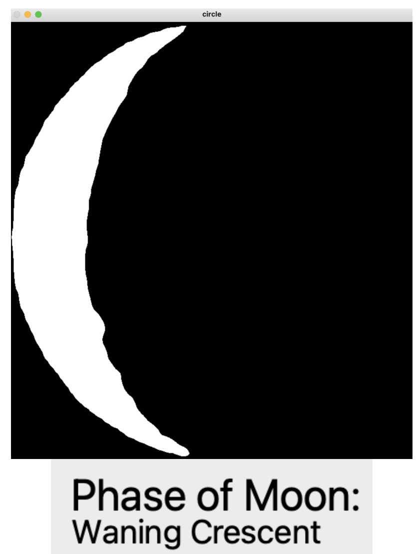


Input Image

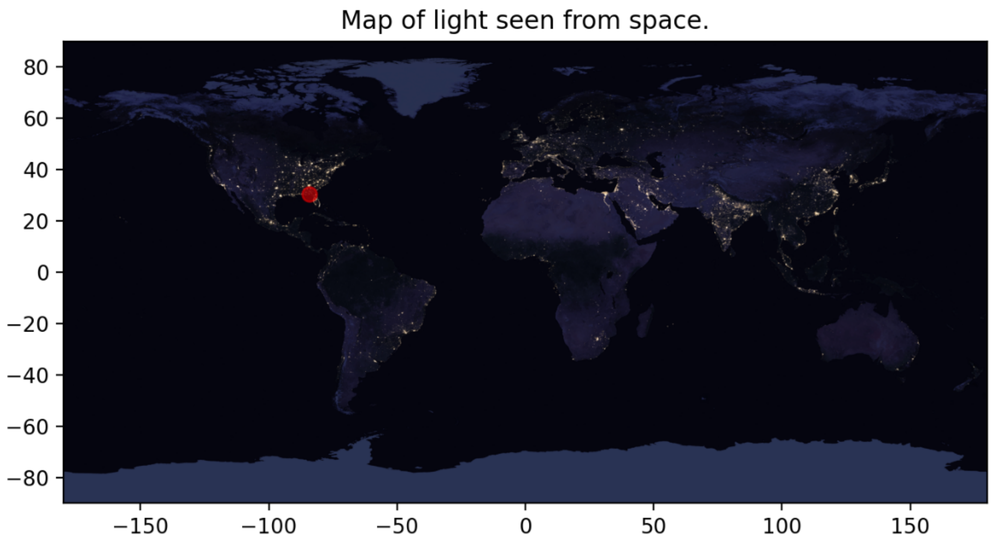
ROI Moon Crop

Output Phase

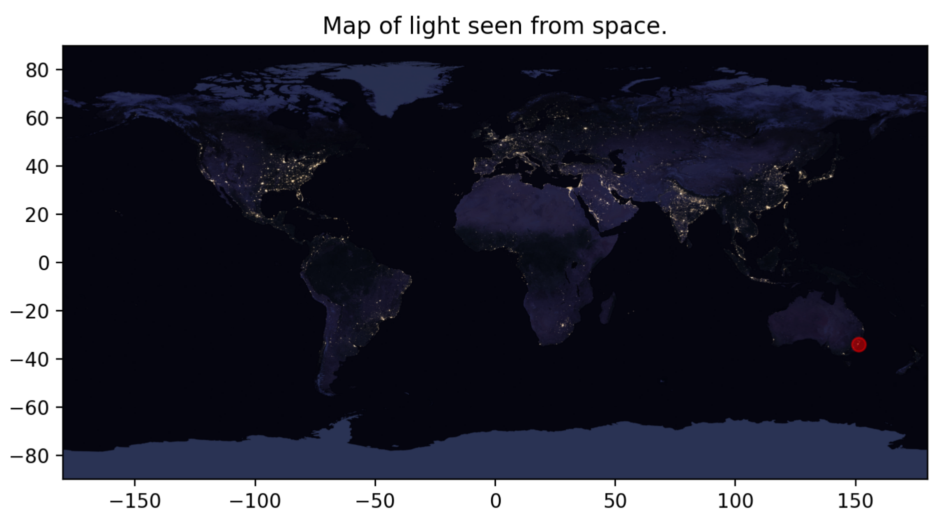




Map Output (1st coordinate is FSU; 2nd is Sydney, Australia)



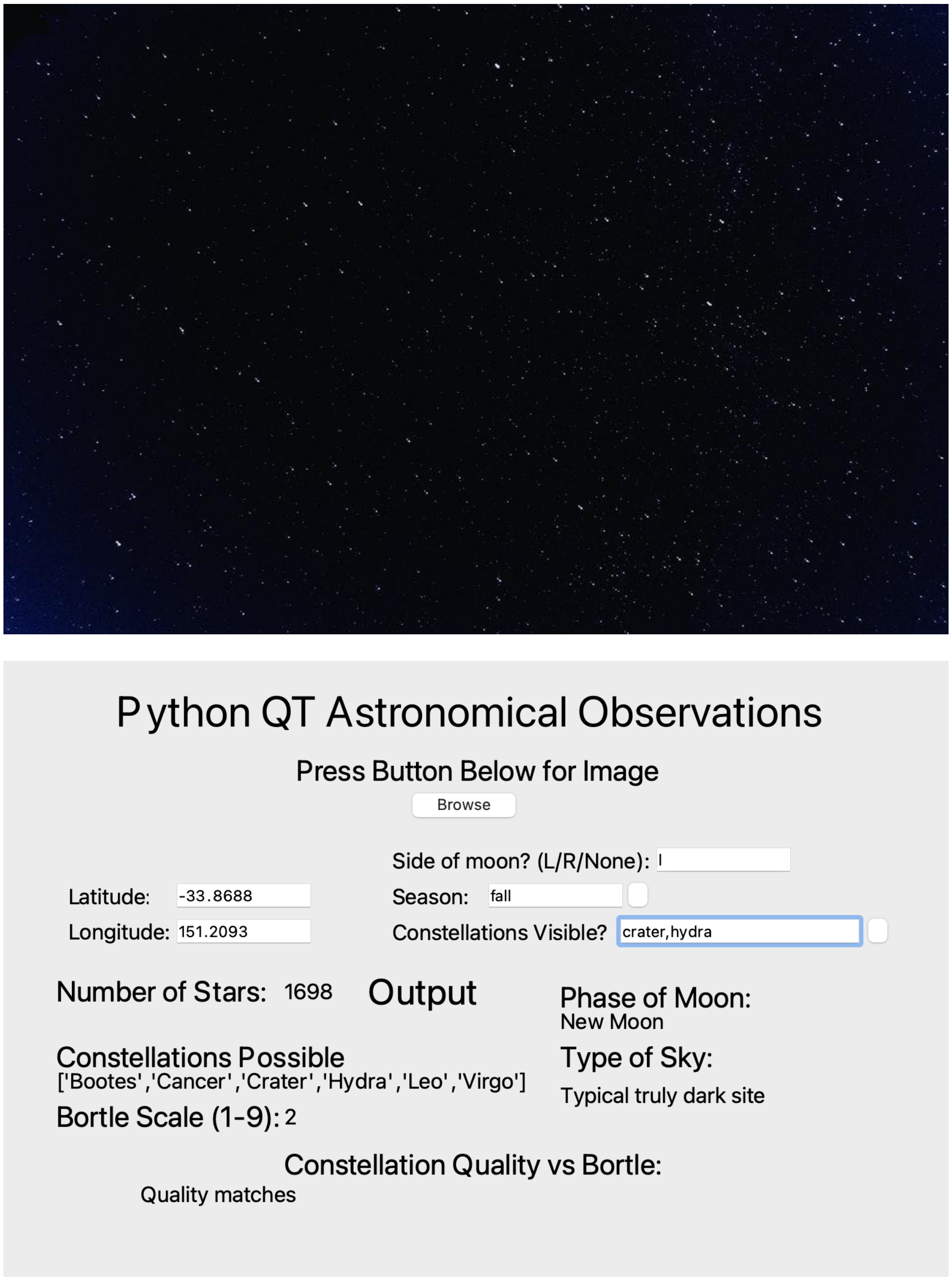
30.4419° N, 84.2985° W (-)



Latitude: -33.8688

Longitude: 151.2093

All together:



Note: program will assume a new moon if no moon is present because no moon can be selected from the image. Additionally, another difficulty was compiling lists of bright, middle, and dim constellations as that required piecing together a few sources explaining their visibilities.

References

Counting Stars. (n.d.). Notebook Community. Retrieved July 12, 2022, from https://notebook.community/dedx/STAR2015/notebooks/CountingStars

Exploring the Southern Hemisphere Constellations - Windows to the Universe. (2012). Windows 2 Universe. https://www.windows2universe.org/the_universe/Constellations/south_constellations.html

Hingston, D. (n.d.). *Star Recognition Using Computer Vision (OpenCV)*. Instructables. Retrieved July 12, 2022, from https://www.instructables.com/Star-Recognition-Using-Computer-Vision-OpenCV/

How the Moon Phases Work. (n.d.). The Grove. Retrieved July 22, 2022, from https://grove.ccsd59.org/wp-content/uploads/sites/10/2015/10/2.1a-How-The-Moon-Phases-Workpdf.pdf

Miller, J. (2016, March 20). *Star Constellations & Brightest Stars* -. Astronomy Trek. https://www.astronomytrek.com/star-constellations-brightest-stars/

Minoia, A. (2019, November 29). *What Is The Bortle Scale? Light Pollution Classes Explained*. Night Sky Pix. https://nightskypix.com/bortle-scale/

National Geographic Society. (2020, July 15). *Light Pollution | National Geographic Society*. Education.nationalgeographic.org. https://education.nationalgeographic.org/resource/light-pollution

NASA VIIRS Land Products. (2022, February 2). Viirsland.gsfc.nasa.gov. https://viirsland.gsfc.nasa.gov/Products/NASA/BlackMarble.html

Old Farmer's Almanac. (2019). *First Quarter Moon Begins*. Old Farmer’s Almanac. https://www.almanac.com/astronomy/moon/calendar

published, T. K. (2022, January 25). *Can you see stars in light polluted skies?* Space.com. https://www.space.com/stars-in-light-polluted-skies

Science A-Z. (n.d.). Science A-Z. Retrieved July 22, 2022, from https://www.sciencea-z.com/science/resource/SL_Gr_5_Earth

Shapiro, A. (2022, July 17). There were 344 ways the Webb telescope could fail after launch — it still succeeded. *NPR*. https://www.npr.org/2022/07/17/1111714756/james-webb-telescope-big-bang-galaxy-image-interview-project-manager-bill-ochs