Title

Astrophysical Image Processing using James Webb Space Telescope Observations on Pillars of Creation in The Milky Way Galaxy

Group Members

Milad Bafarassat
Rasul Ahmad Barak
Arya Hassibi
Kourosh Sharifi

Date

December 26th, 2022

Written by

Kourosh Sharifi

Code Repository

On GitHub

Abstract

The goal of this project was to use computational methods to analyze and process the quantitative data provided by the James Webb Space Telescope (JWST). Specifically, the team worked with images taken by the JWST's Near-Infrared Camera (NIRCam) instrument. These images were unique due to the use of advanced electron absorption technology, which enabled the team to capture highly detailed infrared wavelength data with unparalleled precision. The project aimed to detect and count the stars in a specific region of the Milky Way Galaxy called the *Pillars of Creation*. To accomplish this, various image processing techniques were employed. The rest of the report is dedicated to explaining each phase of the project in more detail.

Introduction

The James Webb Space Telescope is a space observatory optimized for infrared wavelengths. This optimization for infrared wavelengths will enable scientists to go back further in time and see red-shifted light as well as inside nebulas, and other objects that are harder to observe with the visible light spectrum. This information sheds light on the universe's past, present, and future. To tackle this project, a team of four sophomore computer science students has set out to find the answer to the question below:

"How to count the number of stars in a specific region of an image taken by the James Webb Space Telescope (JWST), and to categorize them based on their mass and temperature, in order to compare these findings with other academic literature for different galaxies"

To be able to record and measure these wavelengths, Webb uses the previously mentioned instruments in specific conditions. All devices are kept cold at temperatures below 54 Kelvin, with MIRI being kept at only 7K, to reduce the unwanted noise from the instruments as much as possible. Thereafter, the light gets reflected back to the secondary mirror by the 18 primary mirrors that are precisely positioned relative to each other, and are coated with a layer of gold to maximize the reflection of infrared light. The primary goal of the JWST is to study galaxy, star, and planet formations. The JWST has a total of 4 instruments in the ISIM. Each of these instruments is specialized for a specific set of tasks. For instance, NIRCam is Webb's primary camera, which covers the infrared wavelength range from 0.6 - 5.0 micrometers. NIRCam's data, and its analysis, are the primary subjects of this group project. NIRCam uses its near-IR HgCdTe detector to start its electron sensing process. Meanwhile, a semiconductor absorbs an incoming photon, which generates mobile electron-hole pairs. These electrons travel under the influence of pre-built and applied electric fields until they find their way to where they can be collected.

Methods & Materials

A wide variety of tools were employed to develop this project. Namely:

- **Python 3.8:** As the main programming language
- **Astropy:** For acquiring, reading, and employing FIST data
- **NumPy:** For applying a mathematical function to the data
- **Matplotlib:** For visualizing the data using graphs and charts
- **SciPy:** For optimizing part of the code concerned with computation
- **Mikulski Archive for Space Telescopes:** To acquire FITS data
- **OpenCV:** For image manipulation, algorithm implementation, and image generation
- **ScikitLearn:** For linear regression

The algorithms utilized were many, including

- Canny Edge Detection
- Median Blurring
- Otsu Thresholding

More information can be found in the upcoming pages, regarding algorithms used, a broad description of each, their use-case, why they were used, and what was obtained by implementing them in the code. As for a brief description of the whole process:

1. The required data was collected from an archive.
2. The downloaded data was processed.
3. A number of statistics were obtained after the modifications.
4. Conclusions were made after visualizing and reasoning the data

Data Collection

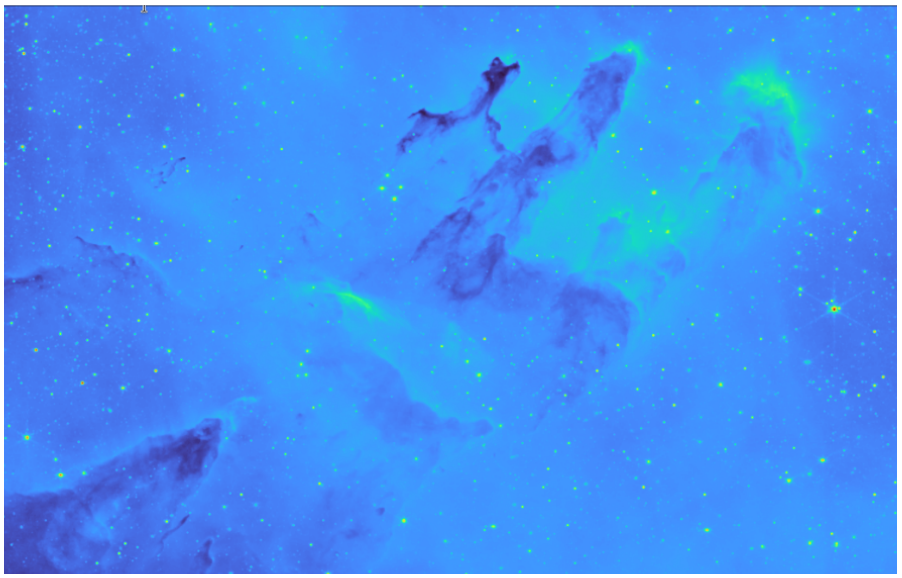
Source

The first step was to obtain information about the m16 region (also known as the *Pillars of Creation*) via Flexible Image Transport System (FITS) from the Mikulski Archive for Space Telescopes (MAST). By specifying the desired space telescope as James Webb, the portal then displayed the necessary parameters to choose and filter the data.

As for the camera, the Near Infrared Camera (NIRCam) was chosen as the observer. Thereafter, the file was downloaded, unzipped, and then stored with the Jupyter notebook at hand. For the sake of learning, some sample runs were conducted to understand the content of the dataset, and how to visualize the data.

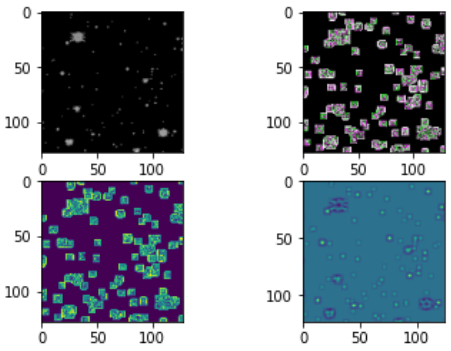
Python Library

After that, the team began to construct the first part of the program using the Astropy library. This library is a collection of software packages for astronomy and astrophysics. It provides a wide range of tools and functions for working with astronomical data, including functions for reading and writing data files, handling celestial coordinates, performing photometry and spectroscopy, and much more.



Background Estimation

Background estimation is a technique used in image processing to estimate the regions that do not contain foreground objects. The goal of background estimation is to separate the foreground objects from the background so that they can be analyzed or processed separately. During this stage, the aim was to cancel the noises in the m16 image, so that the results would be more accurate. The team used convolution and a customized kernel to achieve this goal via Gaussian blurring.

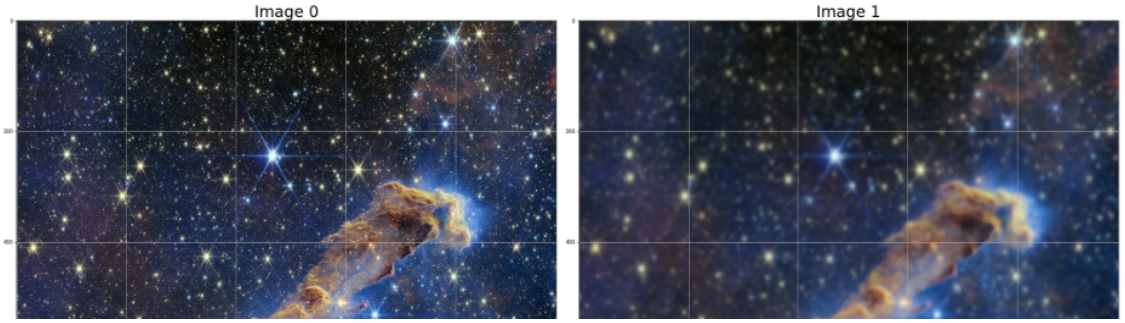


Gaussian Distribution

The Gaussian distribution, also known as the normal distribution, is a continuous probability distribution that is defined by a bell-shaped curve. It is commonly used as a mathematical model for the distribution of intensity values in an image.

$$f(x) = \frac{1}{(\sigma\sqrt{2\pi})} \cdot e^{(-(x-\mu)^2/(2\sigma^2))}$$

Gaussian blurring involves the use of a 5x5 kernel of pixels and the application of coefficients to each of the 25 squares within the kernel. These coefficients are higher for the central pixels and lower for those on the edges. The resulting pixel value for the kernel is obtained by considering the values of all 25 squares. This process results in a bell-shaped or normal distribution of pixel values within the kernel. The effect of this process is to soften edges and enhance the intensity of central pixels, improving the accuracy of object detection.



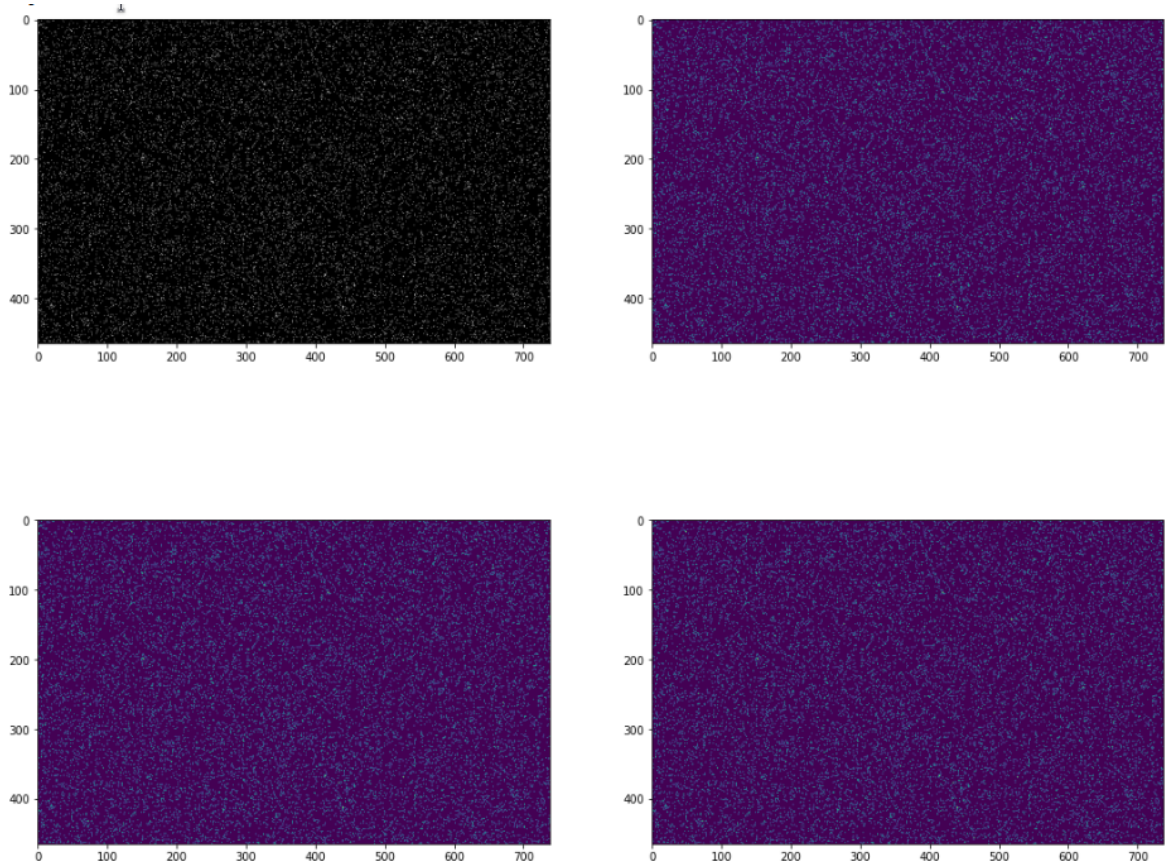
Median Blur

Median blur is suitable for noise reduction and smoothing. It works by replacing each pixel in the image with the median value of the pixels in a surrounding neighborhood. It is typically used as a pre-processing step to reduce noise before applying other image-processing techniques, such as edge detection or image segmentation. Therefore, the team decided to apply the raw image once before performing more advanced algorithms on the data.

To apply median blur to an image, a kernel size would be defined, and then the kernel would be applied over the image, computing the median value of the pixels in the kernel for each position.

$$Median(x,y) = median(I(x-1,y-1), I(x,y-1), I(x+1,y-1), I(x-1,y), I(x,y), I(x+1,y), I(x-1,y+1), I(x,y+1), I(x+1,y+1))$$

To compare the black and white version of the image with its 3 other variations, which are: gray, median-blur filtered, and thresholded version.

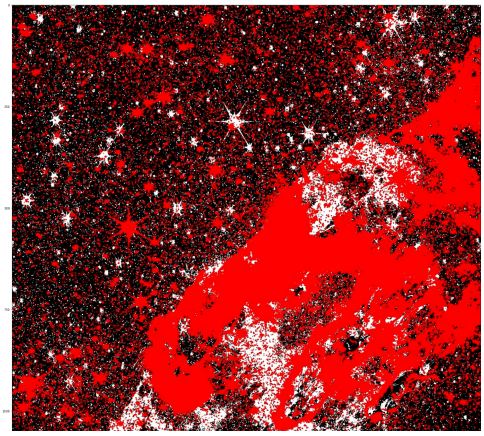


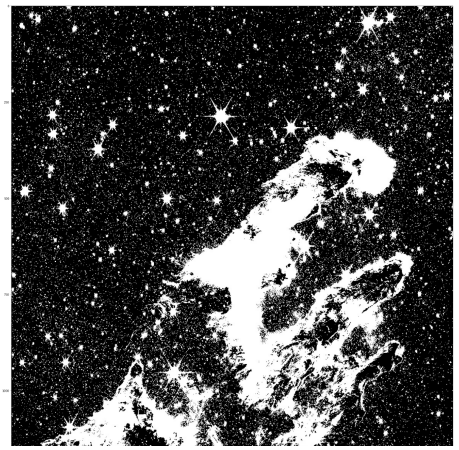
Otsu Method

The Otsu algorithm determines a threshold value that can be conditioned to separate the foreground and background of an image. The goal of the algorithm is to minimize the sum of squared distances between the foreground pixels and the background pixels while maximizing the number of foreground pixels.

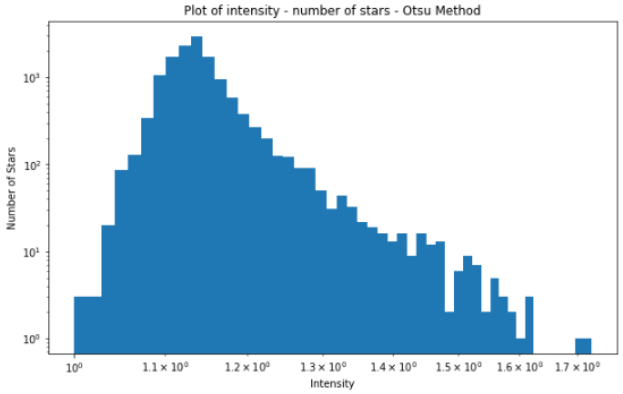
$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

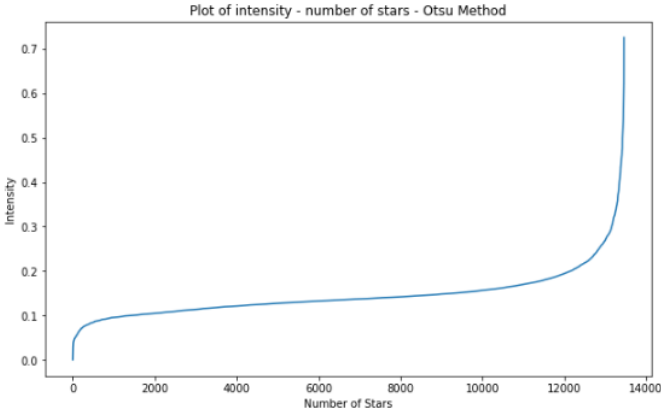
After removing the noise (via median blurring), star detection began, by using Otsu's method. By performing an image segmentation, which involves the division of an image into foreground and background pixels, the weighted variance between them was calculated. The process resumes by iteratively updating the estimates for the foreground and background until the grouping with the lowest variance is found. The resulting threshold is then used to divide the image into foreground and background pixels. Once the division was over, a Hough circle detection algorithm was applied on top of the findings to aid in a better understanding of the distinguishing process.





By printing the number of total non-zero (or non-black) pixels, the number **685,695** was reported as the total number of stars, which was a far-fetched count for such an area. The problem was that each pixel was considered an individual star, which could not be further from the truth. Therefore, by correcting the course of the research, an edge detection algorithm was chosen as the next best approach.

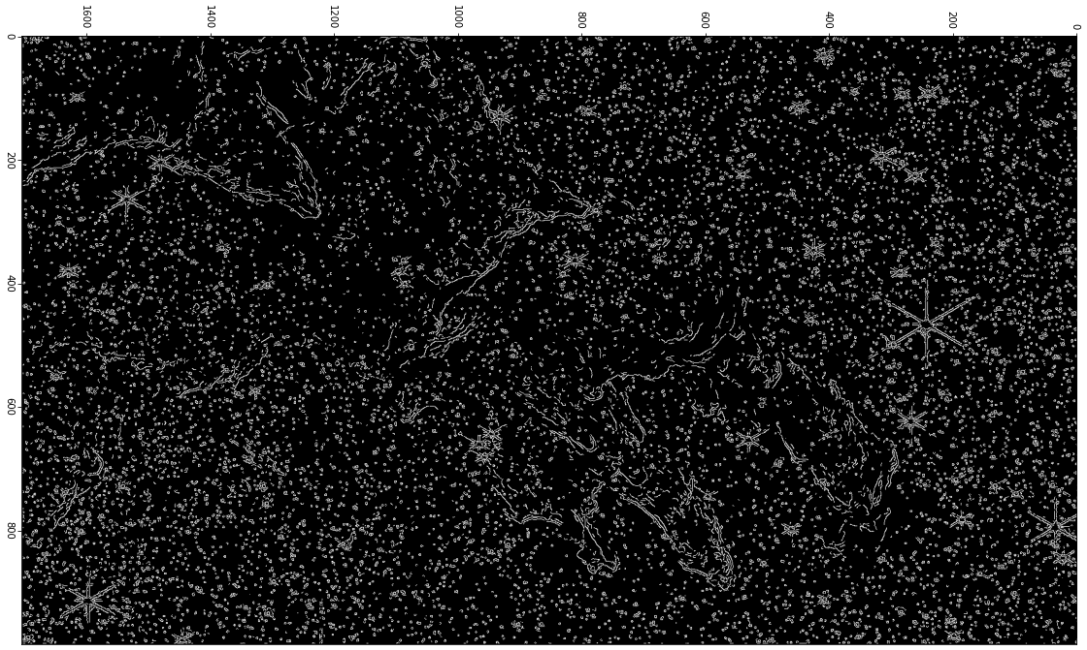




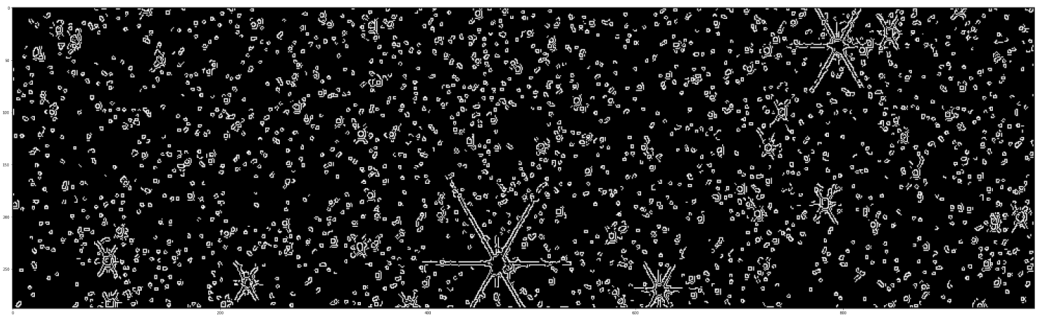
Canny Edge Detection

The Canny edge detection algorithm is a method used in image processing to detect edges in an image with multiple stages that consist of the following steps: 1) noise reduction, 2) gradient calculation, 3) non-maximum suppression, 4) double thresholding, 5) edge tracking

The algorithm computes the gradient magnitude and direction of the smoothed image using a gradient operator. Next, it applies non-maximum suppression to thin the gradient magnitude image and eliminate weak edges that are not likely to correspond to object boundaries. Finally, the algorithm applies hysteresis thresholding to identify strong edges that are likely to be real object boundaries and to suppress weak edges that are likely to be noise.

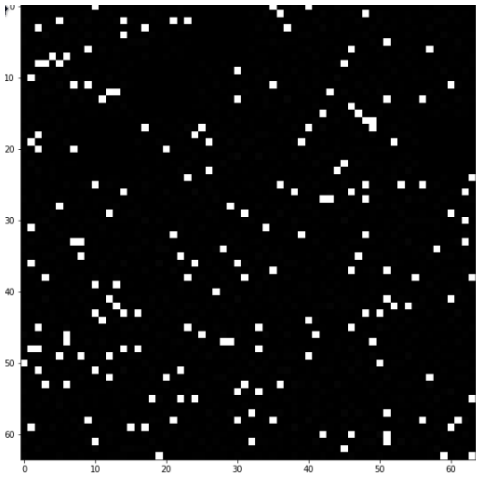


The Canny edge detection algorithm can be used for star detection in astrophysical images by applying the algorithm to the image and using the resulting edge map to identify the locations of the stars in the image. It can be seen that the artifacts are detected too with this algorithm. The number of stars detected in this step was **15,697**.



Rectangle Detection (OpenCV)

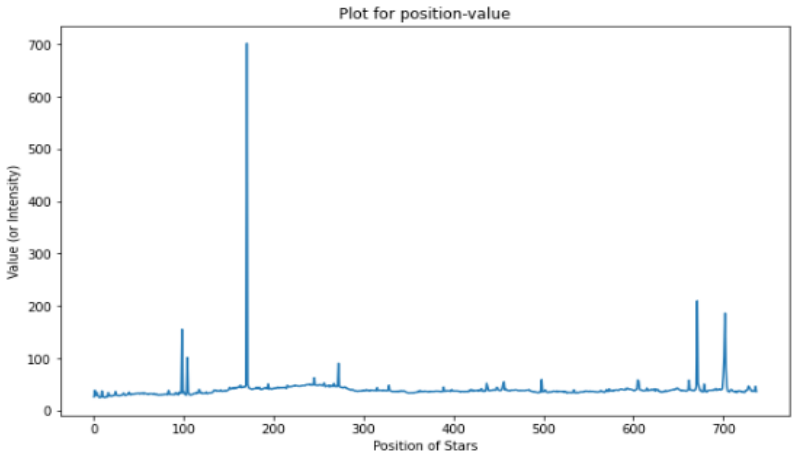
The rectangle detection algorithm in OpenCV is a method of detecting rectangular objects in an image. It is based on the Hough transform, which is a feature extraction technique that can be used to detect lines and shapes in an image. Although the Hough circle detection was already tested, its transform function performs a different task. Once the edges are found (using Canny), the rectangles resulting from the filtering can be identified using the Hough functions. Thereafter, rectangles that encapsulate the objects can be drawn. On top of that, a number of manual star counts were also conducted to compare the findings of the program with human findings. Based on the 5 samples chosen from different **64 by 64** pixel squared regions of the pre-processed image, an **8-11%** difference was observed between the 2 reports, meaning that this technique offered an approximately **90%** star-detection accuracy.



Meanwhile, the same edges that were related to the contours found by the Canny algorithm were now suitable for pinpointing the minimum and maximum x and y coordinates for each edge. By accessing the FITS file and obtaining the intensities within the range of these minimum and maximum values, it was possible to calculate the total intensity of each star.

Plotting Intensity per Pixel

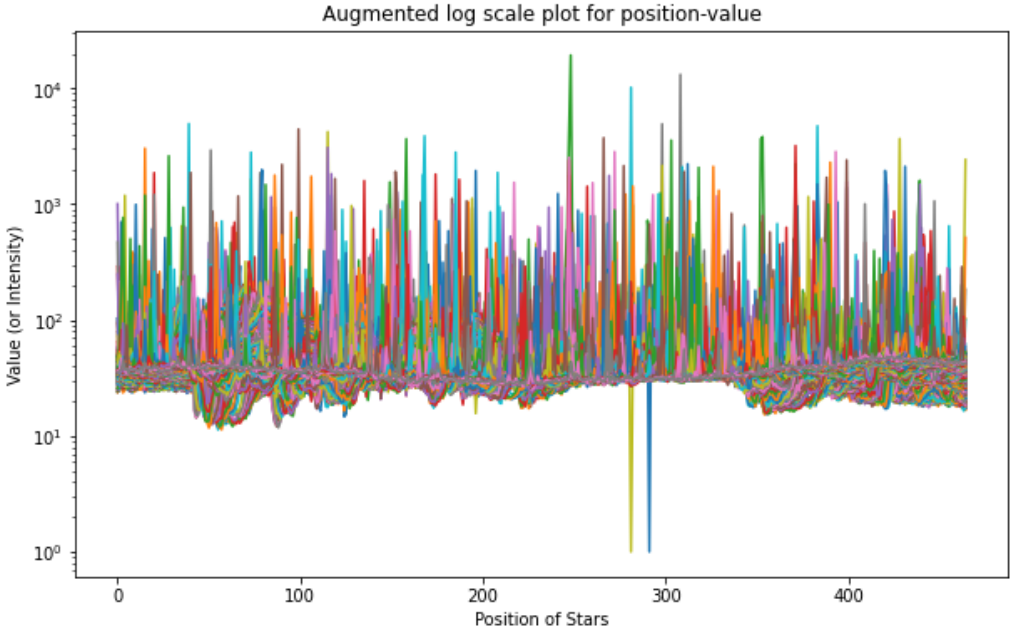
First, only the first block of the image data was selected to be graphed, having its pixel coordinates and related intensities. The normal scale of this initial task can be seen below:



By plotting the complete original dataset without any filtering, one can acquire the logarithmic graph that shows the correlation between the position of the stars (array-wise) and their pure intensity. Note that a slight modification was applied to the data points that had the value of 0 as their intensity, by simply adding 1 to their value.

$$\forall a_{i,j} \in A : a_{i,j} := a_{i,j} + 1$$

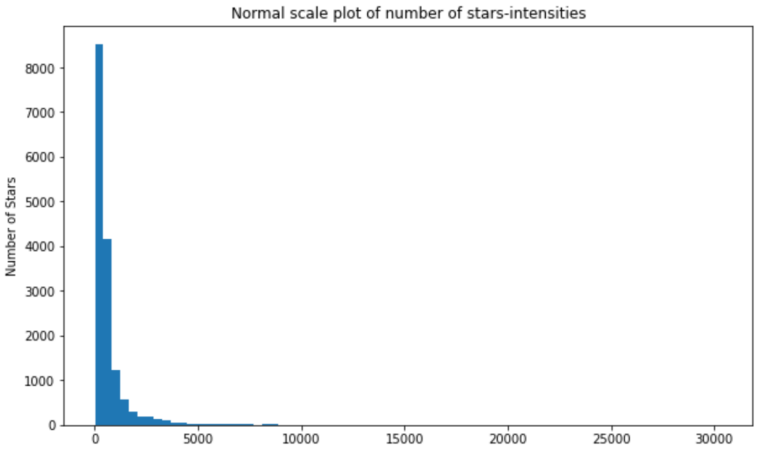
Thereafter, the plot below was obtained for showcasing the desired relation between the parameters:



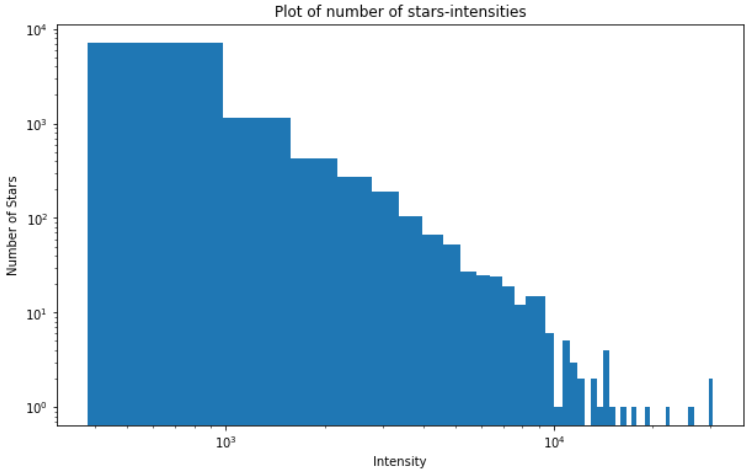
Plotting the Data

After this step, the last phase of the project was to plot the histogram of intensities of the stars detected to analyze the accuracy of the algorithm and make the assumption of the number of stars with more confidence. The below figures below show the histogram for all **15,697** intensities of stars detected based on 2 scales:

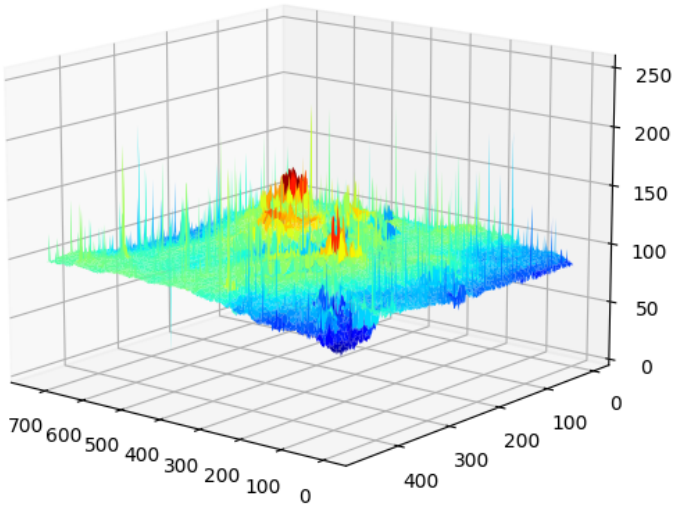
Normal Scale Plot



Log Scale Plot



3D Plot

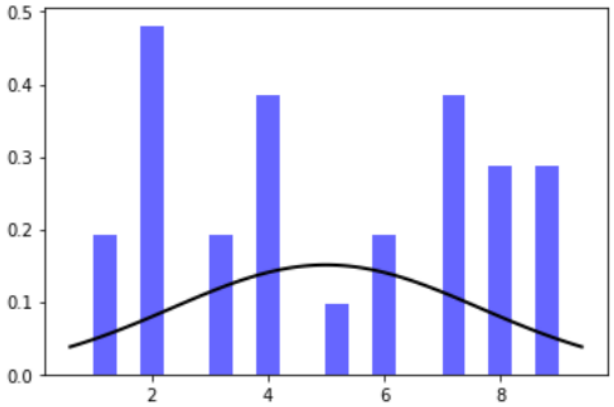


Comparison Using Anderson-Darling Test

One way to check whether the normal distribution is a good fit is to use a goodness-of-fit test, such as the Anderson-Darling test. This test determines whether the data is significantly different from a theoretical distribution.

$$\mathrm{AD} = -n - \frac{1}{n}\sum_{i=1} n(2i-1)\left[\ln(F(X_i)) + \ln(1 - F(X_{n-i+1}))\right]$$

As an example:



As for the test's results, the statistic was reported as **2,725.445**, which makes it a non-normal distribution, since the number is too high. Therefore, this dataset is not close to a normal distribution.

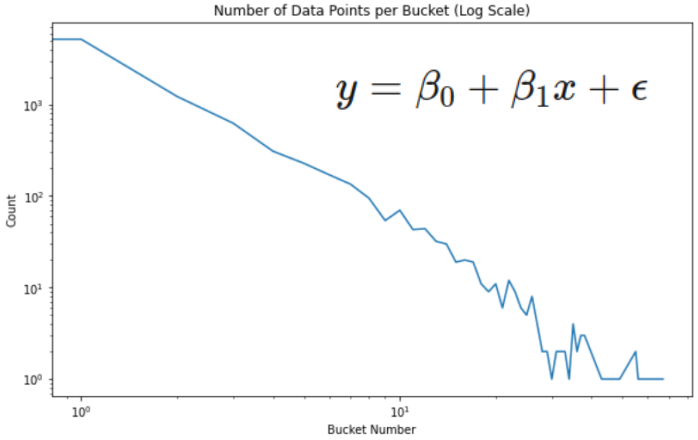
Regression & Best Fit Curve

Regression is a statistical technique used to model the relationship between a dependent variable and one or more independent variables. It is used to make predictions about the value of the dependent variable based on the values of the independent variables. There are several different types of regression, including linear regression, logistic regression, and polynomial regression.

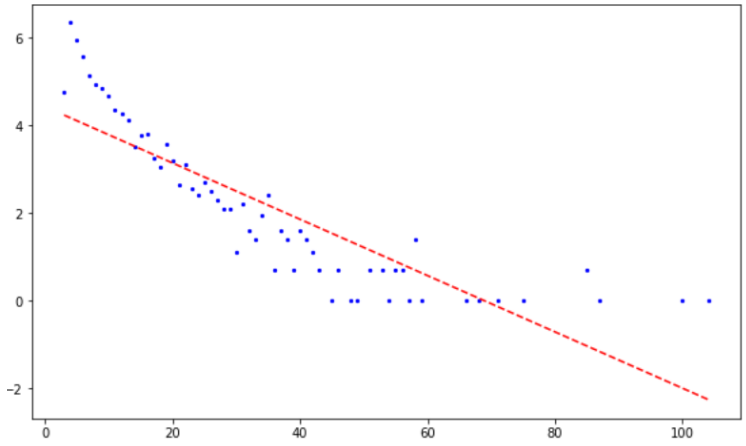
In statistical modeling, a best-fit curve is a curve that is the best approximation of the data points. This curve is found by minimizing the sum of the squares of the differences between the data points and the curve. The resulting curve is often used to make predictions about the response variable based on the predictor variable. The type of curve that is used as the best-fit curve depends on the nature of the data and the type of relationship that is expected between the variables. Some common types of best-fit curves include linear curves, polynomial curves, and exponential curves.

Linear Regression

Linear regression is used to model the relationship between a continuous dependent variable and one or more independent variables by fitting a straight line to the data. In this scenario, a linear regression model was applied after grouping the data into distinct buckets.



In this equation, y is the dependent variable, x is the independent variable, β0 is the intercept term, β1 is the slope of the line, and ε is the error term. However, it should be noted that this model only applies when the logarithmic scale of the data is being inputted since it follows an exponential trend. Therefore, the power law was chosen as the proper trend.

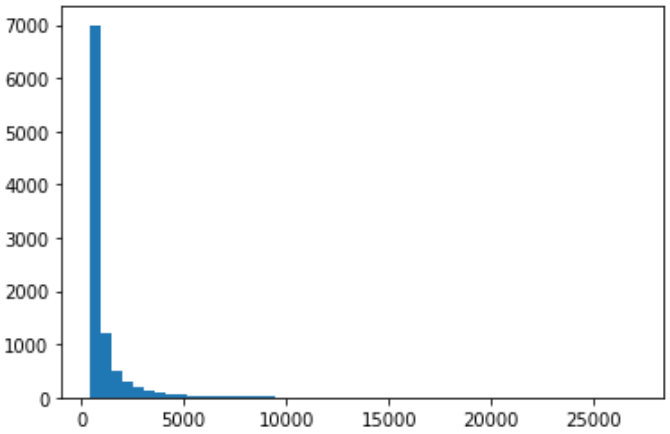


Power Law Trend

In a power law trend, the frequency of an event is inversely proportional to its rank or importance. This means that a small number of events or outcomes occur much more frequently than the majority of circumstances.

$$y = ax^b$$

The power law exponent, *b,* is typically between 0 and 1 but can be greater than 1 or less than 0 in some cases. A value of *b* greater than 1 indicates a steep slope, meaning that the frequency of the events decreases rapidly as their rank or importance increases. A value of *b* less than 1 indicates a shallower slope, meaning that the frequency of the events decreases more slowly as their rank or importance increases. A value of *b* equal to 1 indicates a linear relationship between the frequency and rank of the events.



After a certain intensity value, a power law trend is easy to detect in the histogram above. This trend is a pattern that occurs when the frequency of an event or phenomenon is plotted against the size of the event. This type of relationship is often characterized by a straight line on a log-log plot, with the slope of the line representing the power law exponent (see p. 12).

The threshold set for this stage of the experiment was **380**. Based on this number, a total number of **9,666** stars were reported to be found.

Results

Minimum & Maximum Values

For more context, the maximum and minimum intensity values of the original m16 FITS dataset are **1,9372.150390625** and **0.0** (without adding 1).

Slope & Intercept of Best Fit Line

Based on the calculations of the linear regression model applied to the logarithmic scale, the statistics below were acquired:

- coefficient of determination = 0.7418312620853642
- slope = -0.06416629
- intercept = 4.422324992172783

Where the coefficient of determination is represented by the equation below.

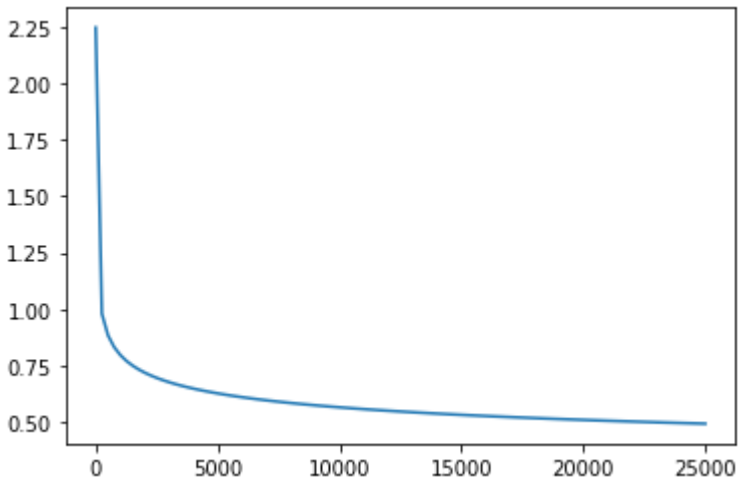
$$R^2 = \left(\frac{\sum_{i=1}^{n} (\hat{y}i - \bar{y})(x_i - \bar{x})}{\sqrt{\sum i = 1^n (\hat{y}i - \bar{y})^2} \sqrt{\sum i = 1^n (x_i - \bar{x})^2}} \right)^2$$

- $\hat{y}_i$ is the predicted value of the response variable for the ith data point
- $\bar{y}$ is the mean of the response variable
- x_i is the value of the predictor variable for the ith data point
- $\bar{x}$ is the mean of the predictor variable. The symbol ρ represents the Pearson correlation coefficient.

Note that the coefficient of determination is a measure of the strength of the relationship between a predictor variable and a response variable in a statistical model. It is defined as the square of the Pearson correlation coefficient between the predictor and response variables. Generally speaking, a number above 0.5 is considered a good predictor.

Power Law Trend's Parameters

Without taking the logarithm of the data, the normal scale representation of the intensity vs. the number of stars can be seen in the graph below. Taking the previously mentioned formula into account:



$$\log(N) = -0.14967724453716968 \cdot \log(I) + 0.08096595652259753$$

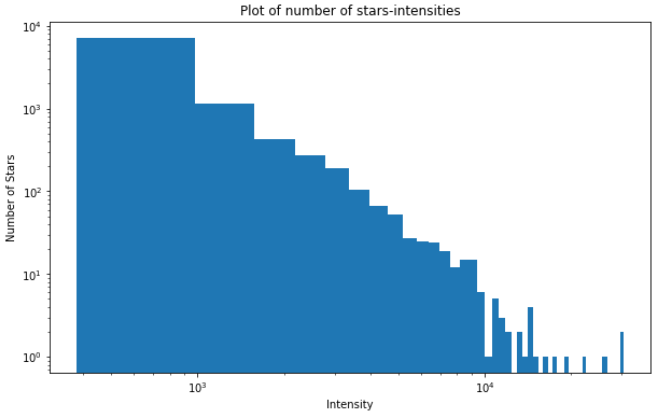
Where '*I*' represents the intensity value, and '*N*' represents the number of stars corresponding to that intensity.

Initial hypothesis

As for the first hypothesis the team had, a strong correlation between the intensity of the stars and the number of stars was expected. Such a relation can be observed in nature quite frequently, hence, it was predicted that space would not be an exception to the linear trend theorem. On top of that, a normal distribution of sorts was anticipated beforehand, since the central limit theorem states the distribution of the sum or average of a large number of independent, identically distributed variables will be approximately normal, regardless of the distribution of the individual variables (e.g., star intensity, position, etc.)

Findings

The results of the executed experiments point to the fact that not only a strong correlation exists between the variables, but also that the program was successful in detecting and counting the stars with a high degree of precision.



Future Predictions

To predict future values, a confidence interval can be obtained. Based on previous experiments, a confidence level of 90% seems reasonable.

$$CI = \bar{x} \pm z\frac{s}{\sqrt{n}}$$

Where:

- CI = confidence interval
- $\bar{x}$ = sample mean
- z = confidence level value
- s = sample standard deviation
- n = sample size

Knowing this, one can guess the possible number of stars that share a specific intensity, so that without having access to more data, future predictions could be made. A confidence interval of 90.0% was reported as the included range of **[717.40, 735.42]**.

Discussion and Conclusion

Achievements

After processing the FITS file for the Pillars of Creation, the team applied Gaussian filtering to smooth the edges. Otsu's method was initially used to count the number of stars, resulting in a count of **685,695**. However, this number was deemed to be an overestimation due to the assumption that every pixel of the foreground represented a separate star. To improve the accuracy of the estimation, the team applied the Canny algorithm, which resulted in a count of **15,697** stars. Upon plotting the histogram of the intensities of these stars, the team identified a linear trend and used it to estimate the final number of stars, which was determined to be **9,666**. The threshold intensity chosen for this estimation was **380**.

The team chose to model the results using a linear trend because power law relationships are commonly observed in natural and social phenomena and are a widely used model for a variety of phenomena. While the linear equation used in this case accurately estimated the number of stars for lower intensities, there was a gap in the estimate for higher intensities. The team attributed this gap to a lack of information on the distance of the stars from the JWST camera, which prevented the program from distinguishing between stars with similar intensities. Additionally, the team cited the potential for errors in the algorithm itself and the camera's measurement as contributing factors.

Needed Improvements

Even though the team was able to reach its main goal of identifying and counting the stars in the Pillars of Creation image, two other tasks were unfulfilled:

- Finding the masses and temperatures of the detected stars (via GAIA API)
- Comparing the results with reputable sources

Unfortunately, these objectives were not carried out due to time constraints and the numerous difficulties faced during the many trials and errors the team encountered.

Next Steps

As the next step for this project (pending approval from the supervisor), the team plans to utilize the GAIA API to adjust the intensities obtained for the stars based on their distance from the camera. This adjustment will allow for the development of a more accurate model of the relationship between the number of stars and their intensities. The team will then seek to identify the best-fit line/curve for the said relationship and use it to refine the estimation of the number of stars in the image.

In addition, the team plans to apply these algorithms to other FITS files and compare the results to those obtained using other methods and to other findings in the field. This will allow the team to evaluate the accuracy and effectiveness of the algorithms in estimating the number of stars in a given image.

Moreover, the members aim to develop a more user-friendly interface for the program in order to facilitate code execution and enable the possibility of working with software that is equipped with a graphical user interface (GUI).