PHY3287 – Computational Astronomy
Assignment – Radio Image Source Finder
Deadline: Tuesday 14th January 2020

Introduction

In this assignment, your task will be to develop a script that takes a radio image as input, and returns a list of locations and general outline of where possible radio sources are present. You will be given a number of images to test your script on, and eventually, your submitted solution will be tested on a blind dataset and scored.

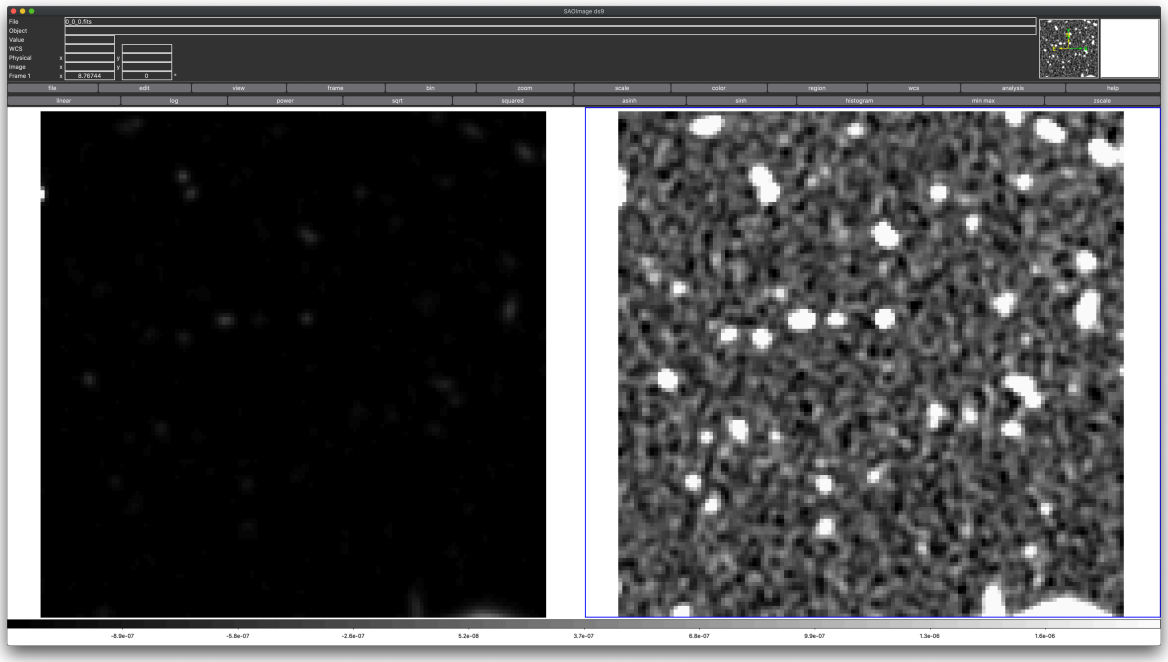
Software Tools

The files to be processed by your script will be FITS files – a very common format for astronomical images. In order to work with FITS files programmatically, you should make use of the Astropy package (https://www.astropy.org). Specifically, the documentation related to FITS files is very helpful (https://docs.astropy.org/en/stable/io/fits/).

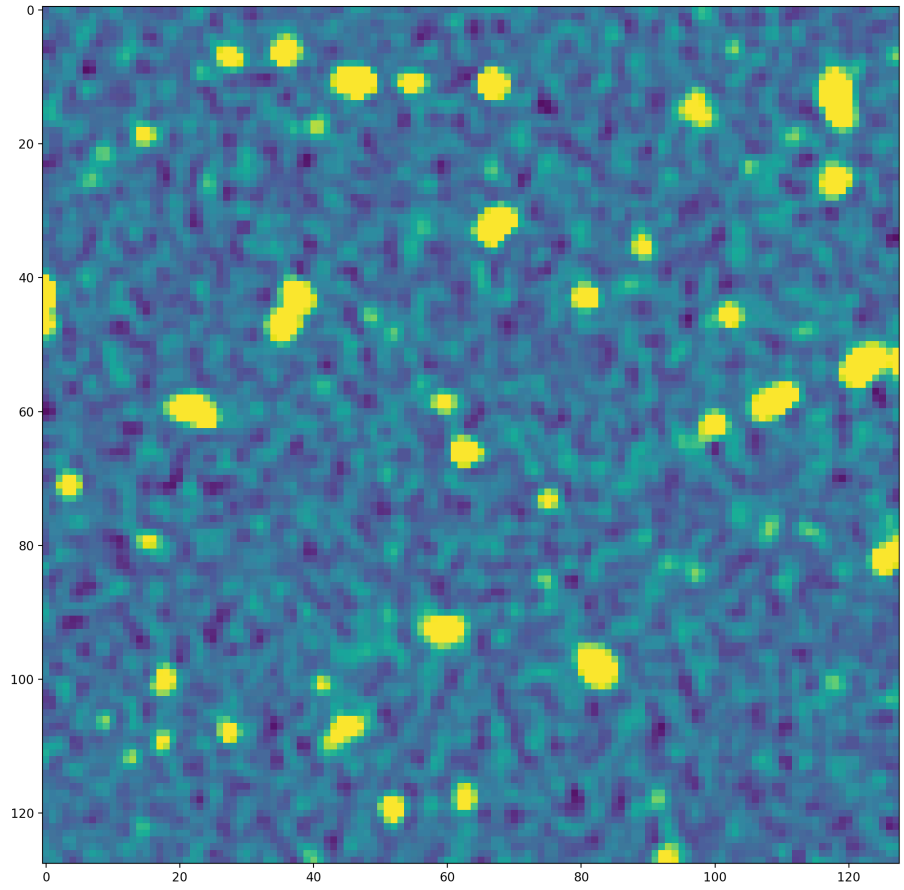
Furthermore, it will help you to visualize the FITS images with a tool such as SAO DS9 (http://ds9.si.edu/site/Home.html).

System Input

The images that you will be processing are two-dimensional 128x128 pixel images, that look like the sample below. This sample has been is show in both raw and Z-scaled format (a common colour-scaling method to enhance source contrast with the background):



More information about how Z-Scaling works is available in the **Appendix**, though Astropy has methods to do the conversion for you. A typical result from Astropy Z-Scaling will look like the following:



Part 1: Image Masking

The first task of this assignment is to clean up the image in order to have clearly visible blobs with all the background sky data re-valued to zero intensity, and sky data belonging to a source re-valued to an intensity of one. This process is known as masking. In order to find what the threshold should be for masking the image, a clipping analysis is required. To help you understand the various operations, refer to: https://photutils.readthedocs.io/en/stable/background.html

There are a number of steps you need to perform to clip and mask the data, in the following sequence:

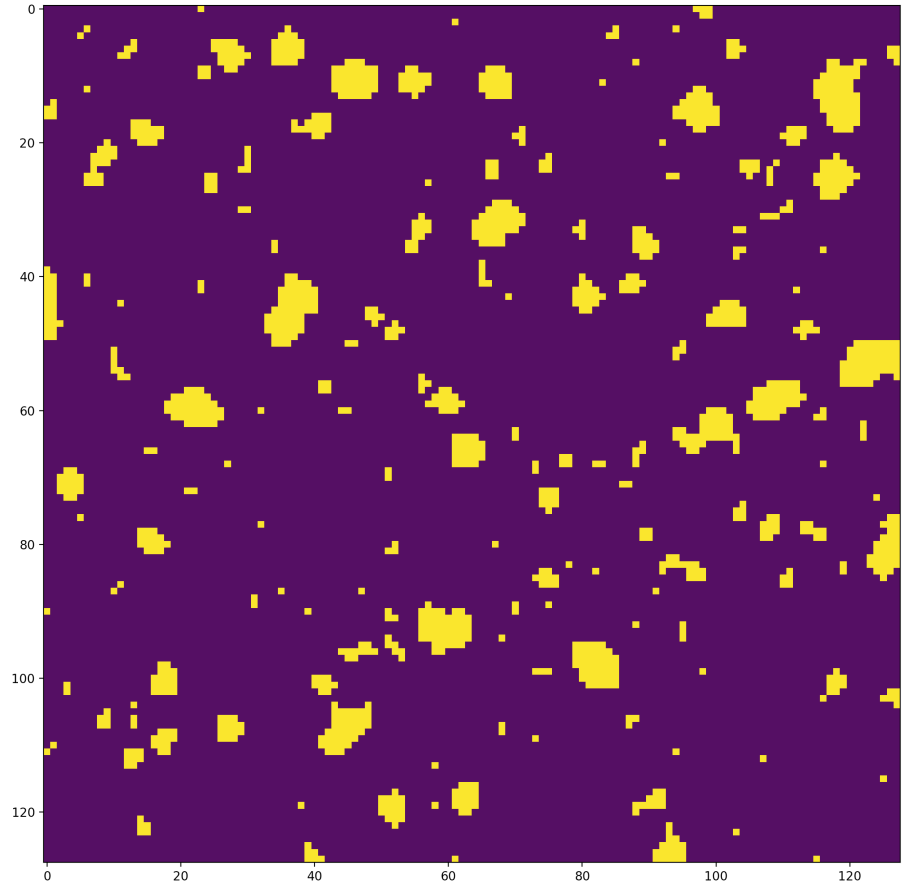
1. Define a Sigma Clipping object, with the number of standard deviations to use for both the lower and upper clipping limit. The default value should be '3'
2. Create a MedianBackground background estimator object
3. Extract the background information for the image, based on the SigmaClip and MedianBackground estimator
4. Discover the threshold pixel value based on the extracted background and background RMS images. A good rule of thumb is to follow the following estimation:

$$threshold = bkg_{image} + (2.0 * bkg_{rms})$$

5. Mask the image. All values smaller than the threshold are set to 0, all other values are set to 1.

Part 2: Source Location and Modelling

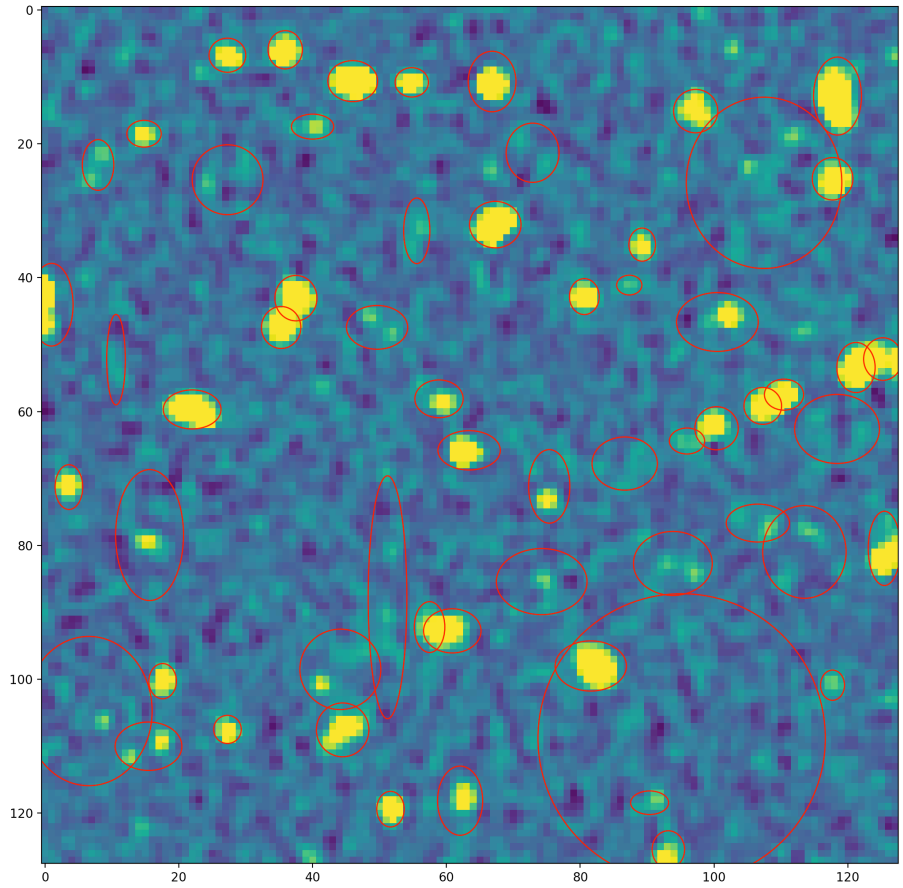
If you view the image resulting from masking, you should expect it to look similar to the below:



Utilizing tools of your choice based on the material covered in the course, your task is to characterise the various sources present in each image. Sources in a mask are represented by 1-valued pixels, whilst the background is represented by 0-valued pixels. You may assume that each 'blob' corresponds to a particular source. Based on this assumption, characterise each source based in the following manner:

1. The centroid/centremost pixel in a blob
2. The maximum length and width of each source
3. Convert this information into a red ellipse drawn on top of each source. Though not all sources are truly elliptical, find the best fit possible for an ellipse that 'outlines' each identified source.

An example (non-perfect) of the end result may look as follows:



Part 3: Report and Submission

The method and optimization to complete this task is left up to you – however a detailed report describing your technique and choices is required. In this report, you should describe:

1. A description of your solution to the problem
2. Any considerations, tweaks, optimisations that were utilised for the task
3. A methodology of experimentation performed
4. A description of any issues, shortcomings, caveats etc. of your solution

Furthermore, you are to submit your solution as a single Python file which should have, as a point of entry, a python function with the following structure, taking in a FITS file path to be processed:

```
def find_sources(fits_file):
        ...
```

A directory with 40 FITS files that you can use to develop and test your system on is found at the following shared folder:
https://drive.google.com/drive/folders/1NwJcV0y7jUtYF24YWcZhkAhYjYCb3Epo?usp=sharing

Your system will be tested independently on another set of unseen data files for assessment purposes.

Appendix

The Z-Scale algorithm is designed to display the image values near the median image value without the time-consuming process of computing a full image histogram. This is particularly useful for astronomical images which generally have a very peaked histogram corresponding to the background sky in direct imaging or the continuum in a two-dimensional spectrum. Generally, the median pixel value is estimated over a number of sample pixels along the image (computationally more efficient than calculating a median over all the pixels).

Next, all pixel intensity values in the image are ranked in order of brightness as a function I(i) – where 'i' is the rank of the pixel and 'I' is the pixel brightness. The midpoint of I(i) i.e. the median, is very near the peak of the image histogram, and a well-defined slope about the midpoint defines the width of the image histogram. The extreme points of I(j) would represent the brightest and darkest points of the pixel intensities. The slope is determined by:

$$I(i) = intercept + slope * (i - midpoint)$$

If more than half of the points are rejected (in the internal iterative process) then there is no well-defined slope and the full range of samples define the min-max normalisation range z1 and z2. Otherwise the endpoints of the function are defined as follows:

$$z1 = I(midpoint) + \left(\frac{slope}{contrast}\right) * (1 - midpoint)$$

$$z2 = I(midpoint) + \left(\frac{slope}{contrast}\right) * (npoints - midpoint)$$

The contrast parameter can be customised, but is generally kept at 0.25.