AstroIMP Documentation

Eduardo Bojórquez Martínez[1]

[1]Industrial Engineering Department, University of Sonora, Hermosillo, Sonora.

May 21, 2024

Abstract

AstroIMP is a tool for automated pre-processing astronomical images, and manipulating them since the quality of the images is usually low compared to other traditional images. Some functions are contrast enhancement, crop images, reduce noise, reduce *CCD* errors, remove the background, and stay only with the objects of interest to detect objects automatically with some algorithm for image segmentation. This document has the intention to explain how to install and use it.

1 Introduction

Since the emergence of Artificial Intelligence, commonly abbreviated as *AI*, aims to solve complex problems smartly. There are various definitions of *AI*, Ben Coppin defines it as: "The use of methods based on the intelligent behavior of humans and other animals to solve complex problems" [**?**].

One of the areas of study of *AI* is pattern recognition, which we can define as the discipline of *AI* that aims to classify a pattern into several categories or classes [**?**]; That is, how computers can observe the environment and learn to distinguish between different patterns, to make decisions based on their categories.

Object recognition is the task of identifying an object found in an image or video. Two of the most important steps in digital image processing are pre-processing and segmentation.

For the preprocessing step, everything that could generate noise is eliminated, the image is enhanced to facilitate image analysis, and there are different image enhancement techniques. In astronomical images, extended structures are normally surrounded by point objects (stars), the latter must be eliminated or cut at the maximum values, to prevent them from causing noise in the identification of emission nebulae and allow easier detection. However, when images are very noisy, it can cause the edges of surrounding objects to also be blurred. On some occasions, after carrying out the noise reduction process, they may be distorted, causing the results to be inaccurate or erroneous [?], [?], [?].

Segmentation is the process of partitioning/dividing a digital image into multiple meaningful regions or sets of pixel regions. The objective is to detect regions of interest (ROI) and simplify the representation, in such a way that it can facilitate image analysis. Image segmentation is an important technique of digital image processing, it is a difficult task to process since subsequent tasks such as object detection, feature extraction, and classification depend on the quality of the segmentation process. Currently, there is no general way to correctly segment all images, there are different types of images and each one can have a characteristic that makes them unique, which makes the problem of selecting regions of interest difficult [?].

2 Requirements

2.1 How to install anaconda and astropy

Anaconda is an open-source distribution that includes packages to process high volumes of information, perform predictive analysis, and scientific computing, is one of the easy ways to perform data science and machine learning on a single machine.

Anaconda distribution is available on the official site for Windows, Mac, and Linux:

`https://www.anaconda.com/download/success`

Download the installer for your operating system then follow the steps:

For Windows, do not install as Administrator unless admin privileges are required:

1. After the download is completed, you need to go to your Downloads folder and double-click the installer to launch.

2. Then click Next.

3. Read the licensing terms and click I Agree.

4. After that you need to select who will be installed, it is recommended that you install for Just Me, which will install Anaconda Distribution to just the current user account. Only select an install for All Users if you need to install for all users' accounts on the computer (which requires Windows Administrator privileges).

5. Click Next.

6. Select a destination folder to install Anaconda and click Next. Install Anaconda to a directory path that does not contain spaces or unicode characters.

7. Choose whether to add Anaconda to your PATH environment variable or register Anaconda as your default Python. We don't recommend adding Anaconda to your PATH environment variable, since this can interfere with other software. Unless you plan on installing and running multiple versions of Anaconda or multiple versions of Python, accept the default and leave this box checked. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.

8. Click Install. If you want to watch the packages Anaconda is installing, click Show Details.

9. Click Next.

10. Optional: To learn more about Anaconda's cloud notebook service, go to https://www.anaconda.com/code-in-the-cloud. Or click Continue to proceed.

11. After a successful installation you will see the "Thanks for installing Anaconda" dialog box.

For Mac:

1. Download the graphical (.pkg) macOS installer for your system.

2. Double-click the downloaded file and click Continue to start the installation.

3. Answer the prompts on the Introduction, Read Me, and License screens.

4. Anaconda recommends that you choose Install for all users of this computer. As of version 2024.02-1, the default location of the installer is /opt/anaconda3. To install elsewhere, select Install on a specific disk.

5. Click Install.

6. Once the installation is complete, click Continue.

7. Optional: To learn more about Anaconda's cloud notebook service, go to https://www.anaconda.com/code-in-the-cloud. Or click Continue to proceed.

8. A successful installation displays the thank you for installing Anaconda Distribution.

Astropy is a collection of software packages written in Python programming language and designed for use in astronomy. The software is a single, free, core package for astronomical utilities due to the increasingly widespread usage of Python by astronomers, and to foster interoperability between various extant Python astronomy packages.

To install Astropy you can use the following command in the Anaconda prompt:

```
conda install -c conda-forge astropy
```

2.2 How to install pre-requisites AstroIMP

2.2.1 OpenCV

The Open Source Computer Vision Library abbreviated as OpenCV, is a library of programming functions mainly for real-time computer vision. In the field of visual computing and image processing, OpenCV has established itself as a fundamental tool. This open-source library offers a wide range of functionalities that allow everything from basic image manipulation to the development of complex facial recognition and object-tracking systems. It was originally developed by Intel, the library is cross-platform and licensed as free and open-source software under Apache License 2. Starting in 2011, OpenCV features GPU acceleration for real-time operations.

To install the OpenCV library for python, you can do it with the following command:

```
pip install opencv-python
```

2.2.2 Imutils

Imutils is a library for Python with a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much easier with OpenCV.

To install the imutils library, you can do it with the following command:

```
pip install imutils
```

3 How to use AstroIMP

3.1 Specify the input and output directory

Since all the functions are required to give a directory with a set of images, it is necessary to first specify the input and output directory to use the desired function.

To specify the input directory of the images:

```
-d or --dir\_images.
```

To specify the output directory of the images:

```
-r or --dir\_results.
```

3.2 Specify crop images

Given a directory with a set of images and specified desired size to create multiple sub-images from the original image. The tool can generate sub-images with the specified size for the width for each sub-image, the height will be the same size as the specified width. The test images we used have a size of 2048 width and 4096 height in pixels, and the desired size was 512 pixels, it generated 32 sub-images with the size 512 x 512 pixels.

To crop images you can use the argument:

```
-c or --crop
```

3.3 Specify preprocess

3.3.1 Contrast enhancement

Image enhancement is an area of image processing that consists of a collection of task-dependent methods that accentuate features important for a specific goal, making human or automated analysis more effective. Astronomical images frequently contain many point objects and, at the same time, extended diffuse structures with embedded point objects; Generally, all images are affected by noise due to the acquisition process of *CCD* devices, so noise reduction is one of the necessary steps for the detection of different objects [?].

For contrast enhancement and background removal, it is required first to specify the argument -p or –preprocess.

To specify if you are going to preprocess the images:

```
-p or --preprocess
```

The ZScale algorithm was first introduced in *IRAF* software explained in [?] and updated for Python with the name PyRAF in [?]. It is made to avoid the laborious task of computing a whole image histogram by displaying image values close to the median image value. Astronomical images, typically have an extremely peaked histogram matching the continuum in a two-dimensional spectrum or the background sky in direct imaging, which can benefit greatly from this [?].

Specifies if applies contrast enhancement, using the algorithm zscale:

```
-zs or --zscale.
```

Percentile range computes the range of pixel values to use when adjusting the contrast of FIT images using a simple percentile cut. A subsample of the input image data is used for efficiency reasons [?].

Specifies if applies contrast enhancement, using the algorithm percentile range:

```
-pr or --percentile\_range.
```

Arcsinh percentile locates the lower and upper boundaries containing all visible objects using a mapping arcsinh that displays the image values.

Specifies if applies contrast enhancement, using the algorithm arcsin percentile:

```
-ap or --arcsin\_percentile.
```

3.3.2 CCD error reduction and high noise peaks reduction

In the tool we implemented the *CCD* errors reduction, the steps we apply are based on the work of [**?**] they use morphological operators to enhance the peaks that are contained in the image, and then remove them. The detailed steps that were followed are:

- Invert the image with its original dynamic range. The literature review mentions that it is a good practice in astronomy because the contrast of the image is improved, and it is easier to visualize for the human eye.

- Restore the original image with the inverted one. [**?**], carried out this procedure to enhance the inverted image and reduce the optical effects of halos around bright stars.

- Apply the mathematical morphological closure operator. By using this operator, all internal noise present in the region of the object is removed and the background is not affected. This technique also has the property of highlighting the peaks contained in the image. We are interested in those with saturated pixels, and flowering effect, among others. To achieve this, a disk-type structure with a radius $r = 8$ was used to enhance the inverted image and highlight the high peaks contained in the image.

- Once the operator closure is completed, the image is inverted again, and the result is subtracted from the original image.

- Finally, it is returned to its original dynamic range again.

To specify if you want to reduce CCD errors:

```
-ccde or --ccd_errors.
```

Due to the dynamic range of both point and extended objects, the types of noise described in the previous sections, and the errors of *CCD*, it is necessary to remove the background from the image to improve object detection extended with low emission. To achieve background elimination, a threshold is determined with the help of the *PFCM* algorithm.

3.3.3 Background removal

For background removal it uses the *PFCM* algorithm, translated to python by [?], with this algorithm we generate two groups, the objective is to divide the background of the image and the extended objects or stars that are in the image, with the result obtained, a mask is generated in which the background is assigned the value of 0, and the objects are assigned the value 1. Then a new image is generated by multiplying the values of the generated mask with the original image, in this way, we maintain the original values of the image, and the rest remains with a value of 0 (black).

To specify the background removal using the algorithm PFCM:

```
-pf or --pfcm.
```

4 Examples

5 Conclusion

Image processing in astronomy is complex, requiring several steps to be carried out beforehand to ensure that noise, image background, and stars do not interfere with the detection of extended objects. A traditional image segmentation method alone cannot be applied since the quality of the images is not the same as a traditional image.

Given the diffuse nature of the emission nebulae, and the noise produced by the *CCD*, it was best to apply the Perona Malik filter and the fuzzy possibilistic algorithm (*PFCM*) to obtain a threshold and be able to remove the background of the image, leaving only stars and extended objects.

6 Bibliography

Acknowledgments