



university of
 groningen

faculty of science
and engineering

Point Source Detection and Removal

In Astronomical Images

Master Thesis

Author:

Björn Schönrock

Primary supervisor:

Dr. M. H. F. Wilkinson

Secondary supervisor:

Prof. Dr. Kerstin Bunte

Abstract

With the increasing amount of astronomical data, analysis by hand becomes infeasible, and automated methods are necessary to perform this task. Point sources, such as stars, are a significant challenge in astronomical images, since they appear as bright spots that can hide or distort fainter structures in the image, such as galaxies or nebulae. Removing point sources can help making these structures more visible. The research on point source removal is not new, but accurate and efficient removal remains challenging. Many approaches, for example neural networks, are complex and computationally expensive, and they are often black boxes that are hard to understand. In this thesis, we aim to develop an efficient and explainable method for detecting and removing point sources in astronomical images using the Point Spread Function (PSF), which describes how a point appears in an image. The first part of the project involves building a classifier to separate point sources from other objects. In the second part, we will subtract the PSF to remove the identified point sources.

We have obtained good results on point source detection, being able to achieve high recall scores on both simulated and real data. Removal of point sources proved much more challenging, with mixed results. We were able to subtract simulated stars with a well-known PSF quite well; however, point sources in real astronomical images leave residuals after subtraction.

The source code of this project can be found on GitHub: <https://github.com/bys1/mtobjects>.

Contents

| | Page |
|---|-----------|
| 1 Introduction | 4 |
| 1.1 Thesis Outline | 4 |
| 2 Related Work | 5 |
| 3 Data | 8 |
| 4 Identification | 9 |
| 4.1 PSF fitting | 9 |
| 4.2 Simulated data | 10 |
| 4.3 Astronomical images | 12 |
| 5 Removal | 15 |
| 5.1 Simulated data | 15 |
| 5.2 Astronomical data | 16 |
| 5.3 Modelling the PSF | 16 |
| 5.4 Elliptical Moffat PSF | 17 |
| 5.5 StarNet++ results | 19 |
| 5.6 Adding simulated Moffat stars | 20 |
| 6 Discussion | 21 |
| 7 Conclusion | 22 |
| 7.1 Research Questions | 22 |
| 7.2 Future Work | 22 |
| Acknowledgements | 24 |
| Bibliography | 25 |
| Appendices | 26 |
| A NGC 4307 subtraction results | 26 |

1 Introduction

Point sources present in astronomical images can obscure extended sources, and to help reveal these sources, removing point sources is a crucial step. As the amount of astronomical images keeps growing, it becomes infeasible to perform such tasks manually, and there is an increasing need for automated methods. The detection and removal of point sources is not a trivial task. First, we need to distinguish point sources from other light sources. Then, they need to be removed, but this often leaves residuals. Removal of point sources needs to be done with care, since the goal is to discover fainter structures obscured by them. Removing too much will affect other structures as well, so ideally, the background should remain intact.

How point sources appear in images is influenced by the effects of seeing and instrumental effects[1, 2]. Telescopes are limited by diffraction, and observations of objects do not have an infinitely high resolution. Therefore, light is scattered across pixels; the image is slightly blurred. This means that the light of a point source is spread out over a larger area in the captured image. This effect can be modelled by a Point Spread Function (PSF), which is the distribution of light in an image plane when a telescope is focused on a point source of light. The PSF is represented by a kernel with which the true image is convolved to form the observed image. In this thesis, the PSF will have an important role: when finding point sources, we are effectively looking for sources that closely match the PSF. After the detection step, we will subtract a scaled version of the PSF from each point source. Therefore, having a good estimate of the PSF is crucial, so the PSF has to be either provided with or estimated from the image.

In this thesis, we aim to answer the following two research questions:

1. How can we effectively detect point sources in astronomical images?
2. Can we use the PSF to accurately subtract point sources from astronomical images?

This project builds upon MTOBJECTS, which is a tool for the automatic detection of astronomical objects in images[3, 4]. We aim to extend MTOBJECTS by classifying the objects detected by it as point sources and other objects, and then producing an image with point sources removed.

1.1 Thesis Outline

The structure of this thesis is as follows. After the introduction, Chapter 2 discusses related work on the subject, including tools and concepts that are used in this project. Next, Chapter 3 discusses the data used throughout this research. In Chapter 4, research on the detection of point sources is described, and Chapter 5 describes our experiments on the removal of point sources. Then, Chapter 6 discusses our findings and reflects on how results might be improved. Finally, Chapter 7 summarizes the findings of this project, and discusses directions for future work.

2 Related Work

Several tools have been developed to facilitate the automatic detection and extraction of astronomical objects from images. In a comparison, Haigh et al. found that SExtractor, ProFound, NoiseChisel and MTOObjects were all able to provide roughly similar performance with respect to detection completeness, but MTOObjects achieved the highest overall performance, and only NoiseChisel and MTOObjects were able to find the faint outskirts of objects [5]. SExtractor is the oldest of these tools, dating from the mid 90s [6]. Although widely used, SExtractor has drawbacks. The first step is estimating and subtracting the background from the image. SExtractor divides the image into tiles and uses an adaptive background estimate. After background subtraction, SExtractor uses a fixed threshold to identify objects, solely based on background estimates in local sections of the image. However, it ignores the object properties and shows a bias from the objects. The background estimates show correlations with large objects, leading to distortions in their shapes after subtracting the background. As a result, SExtractor does not perform well in detecting faint outskirts of objects.

MTOObjects is a tool inspired by SExtractor that aims to improve on these disadvantages [3, 4]. Rather than an adaptive background estimate, it takes the mean value of the tiles and computes a constant estimate for the entire image. In order to identify objects, it constructs a Max-Tree. A Max-Tree [7] is a tree structure in which every node represents a connected component. The root of the tree represents the entire image, and the leaves represent local maxima. An example is shown in Figure 2.1, where a simple gray scale image is thresholded at every intensity level to obtain peak components P_h^k , with h being the intensity and k an identifier to distinguish different connected components with the same intensity. Finally, a Max-Tree is obtained with a node C_h^k corresponding to each peak component. After constructing a Max-Tree, MTOObjects uses four significance tests to separate objects from noise. This works by performing a χ^2 test for every node. In a node P with area A , let $f(x)$ be the pixel value of a pixel x with the background subtracted, where the background is simply the parent node of P . If P is due to noise, the distribution of $f(x)^2/\sigma^2$ has a χ^2 distribution with one degree of freedom for any pixel x in P . The power attribute is defined as the sum of $f(x)^2$ for all pixels x in P , and then, $\text{power}(P)/\sigma^2$ follows a χ^2 distribution with A degrees of freedom. Given a significance level α , a significance test is performed, and if it is rejected, the node is marked as significant.

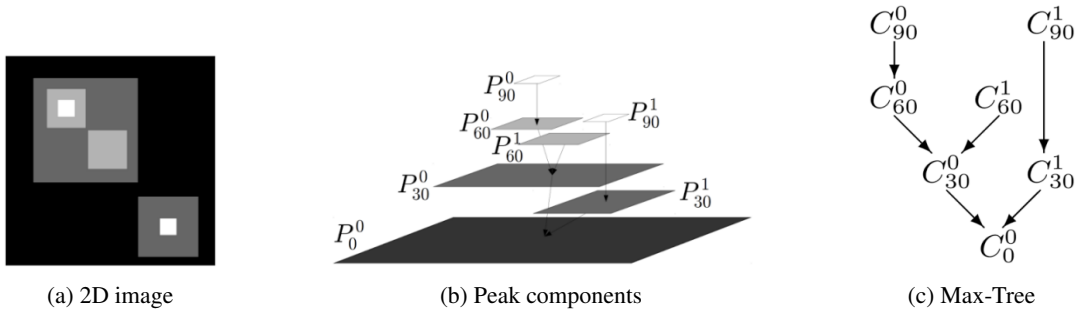


Figure 2.1: Figure taken from [3] showing how a Max-Tree can be constructed from a 2D grayscale image

When the significant nodes are identified, MTOObjects proceeds with marking objects. While several significant nodes can be part of the same objects, separate objects can also be stacked on top of one another. Separating these nested objects is called de-blending, and this is done based on the outcomes of the significance tests performed earlier. Finally, MTOObjects moves the object markers up along the tree. Nodes marked as objects have several noise pixels attached, and to eliminate these, the object markers are moved up using a parameter λ times the standard deviation of the noise. This comes with the cost of losing faint parts of extended sources, and therefore, λ has to be chosen wisely. As a compromise, Teeninga et al. suggested $\lambda = 0.5$ [3].

MTOObjects is able to detect and extract objects successfully, outputting a catalogue of objects along with several computed parameters related to their shape and light profile. However, MTOObjects does not distinguish between different types of objects. The detection and removal of point sources can be an interesting extension of MTOObjects, and the existing work can form a solid basis which this project can build upon.

The topic of point source detection has been widely explored. Lang and Hogg explained the basics of matched filtering, starting with a simple image containing a single point source, under specific, well-known conditions [8]. In this approach, the image is cross-correlated with a known PSF to emphasize point sources. Peaks in the cross-correlation indicate point source detections. Baqui et al. performed star-galaxy classification using machine learning in the miniJPAS survey [9]. In

this research, several features were explored, including morphological and shape-based features such as FWHM, brightness and ellipticity. In particular, the Full Width at Half Maximum (FWHM), which is the radius at which the intensity is half the maximum, proved to be an important feature in classification. Another approach was presented by Bonavera et al., who trained a neural network to discover point sources [10]. While their work was aimed for microwave sky simulations and not for optical imaging, they presented an effective approach to point source detection using neural networks. A disadvantage of this method is that it requires training a model before it can be used.

MOPEX is a package for both detection and removal of point sources in astronomical images [11]. It uses non-linear matched filtering for detection, and then fits the point sources using the Point Response Function (PRF), which is estimated from the input image. Finally, it removes the point sources from the image, applying active and passive de-blending to improve separation of different sources. An image with results, taken from their paper, is shown in Figure 2.2. Image *a* is an input image with the detections shown as white circles, and final extractions shown as white crosses. Image *b* is a point source probability image. Image *c* is a detection map with the detections shown as white circles. The white dashed line circles the detection that was discarded because of its small size. Image *d* is the image after subtracting point sources. The point sources seem to have been subtracted accurately, but small residuals can still be seen.

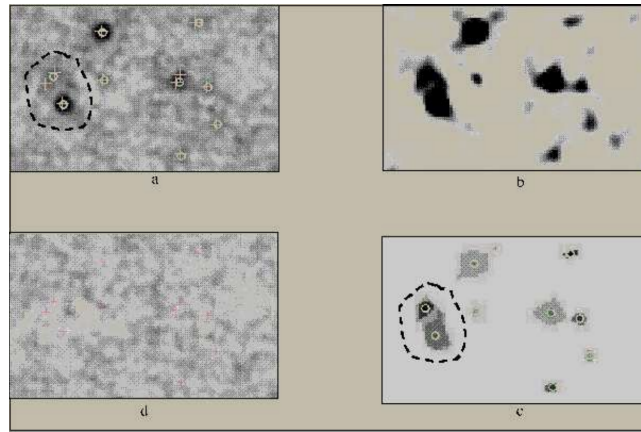


Figure 2.2: Image taken from [11] showing the results of point source detection and subtraction.

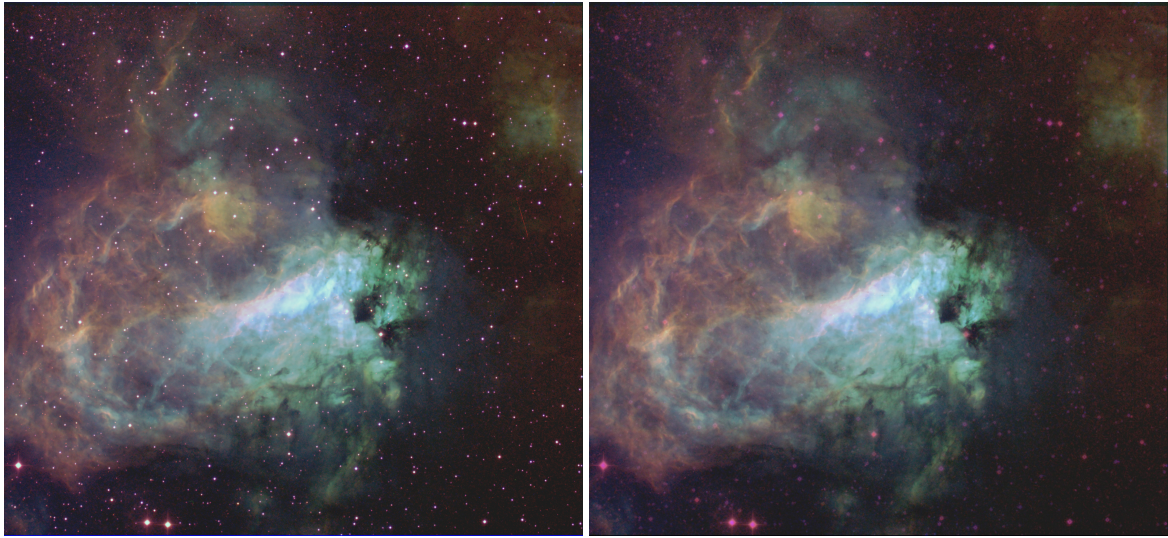
A tool for point source removal developed by Ashish Patil uses a Generative Adversarial Network (GAN) model trained with only two images¹. An example image of the Omega Nebula provided by Patil is shown in Figure 2.3. Overall, the results look good: the stars are removed, making the nebula more visible. However, the brighter stars leave clear residuals in the image.

StarNet++ is a project that uses a neural network to remove stars from images². While it produces seemingly good results, faint residuals are still visible. Moreover, a consequence of the use of neural networks is that StarNet is computationally expensive and has a long execution time as the images become larger. Neural networks are complex, and while they can perform quite well, they are essentially a black box. It is difficult to understand how and why they work. We do not know what StarNet++ subtracts, and if residuals are seen, it is hard to analyse what exactly StarNet++ has done. This is a challenge with many existing approaches, such as MOPEX and the tool by Patil: their complexity makes them both computationally expensive and less transparent.

In this project, we aim to remove point sources in an explainable way by subtracting a PSF. This approach should be much faster than neural networks, and we know exactly what is subtracted, making it easier to analyse its results. Its higher efficiency allows for better scalability, which becomes increasingly important as the amount of data keeps growing.

¹<https://github.com/code2k13/starreduction>

²<https://github.com/nekitmm/starnet>



(a) Omega Nebula with stars

(b) Omega Nebula, starless

Figure 2.3: Point source subtraction on an image of the Omega Nebula with a GAN model, by Ashish Patil.

3 Data

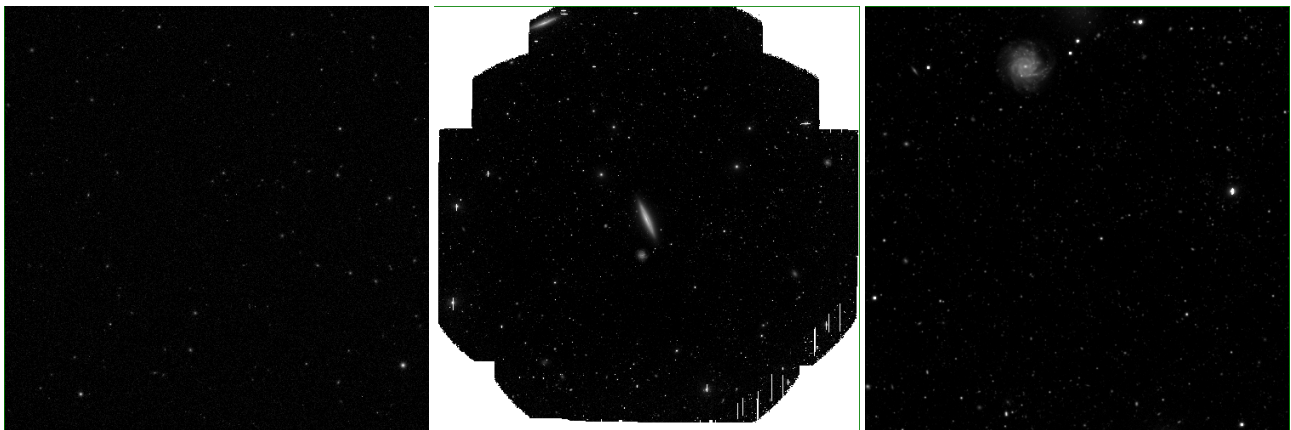
In this thesis, we used two types of images to conduct our research on: simulated and real data. First, we used code provided by Mohammad Faezi that generates images containing simulated galaxies and stars, with a known PSF. This code is based on the Fornax Deep Survey (FDS), which is an imaging survey using OmegaCAM mounted on the VLT Survey Telescope (VST)[12]. OmegaCAM is a wide-field imager that consists of 32 Charge-Coupled Devices (CCDs).

The code by Faezi first places 285 background galaxies at random locations in the image, and subsequently convolves them with an analytical PSF function, which consists of Gaussian, Moffat and exponential components and is tuned to match the OmegaCAM r-band PSF profile. Next, it generates 285 stars using the same PSF function and places them randomly across the image. Finally, Poisson and background noise is added, after which the resulting image is saved to a FITS file. For this project, we modified the code to save a catalogue of objects along with the image, containing the position and type (galaxy or star) of each object. This catalogue is later used for classification of objects.

We also use a real astronomical image, which was provided to us by Minh Ngoc Le. This is an image of NGC 4307 in the g-band from the LBT Imaging of Galactic Halos and Tidal Structures (LIGHTS) Survey, from the Large Binocular Telescope (LBT)[13]. The Large Binocular Cameras (LBC) used consist of 4 CCDs, which are linear with a residual less than 1% over the entire 16-bit range that the analogue-to-digital converter (ADC) can output[14]. The linearity of CCDs means that the recorded brightness increases proportionally with the amount of incoming light. This is important for our research, since it implies that the PSF should be the same for different magnitudes, and we can use the same PSF across the entire image, scaling it as needed. As an exception, nonlinearity can occur at very low light levels (due to noise) or very high levels (due to saturation). When a pixel receives more charge than the ADC can represent, digital saturation occurs, and the recorded intensity is clipped at the ADC's maximum value. When even more light comes in and the full-well capacity is exceeded, the physical pixel is full of electrons. This will result in blooming, where the excess light is spread across other pixels, usually along columns. In the provided image, saturated areas have been removed, by simply setting all pixels in those areas to NULL.

The image of NGC 4307 as provided to us is circular, but stored in a rectangular image of 10053x10053 pixels, with white corners due to missing (NULL) pixels. Because of this, MTOObjects was unable to load this image. As a solution, we cropped the image to 6500x7500 pixels, keeping only the usable area. This cropped image was used in our point source detection tests. For the subtraction of point sources, we were provided with a smaller crop of 2567x2567 pixels, along with the residual image obtained after processing this image with StarNet++.

Along with the image of NGC 4307 from LIGHTS, a Gaia catalogue of confirmed point sources was provided. This catalogue contains 447 point sources with their position and Gaia G-band magnitude, of which 285 fall within the 6500x7500 crop that we made of the image. Gaia is a space observatory of the European Space Agency (ESA). One of the goals of Gaia was to capture objects down to a magnitude limit of at least 20[15]. The sources in the provided catalogue have magnitudes up to 20.9. While this catalogue provides a sufficient amount of point sources to extract information from, this magnitude limit is lower than the limit in LIGHTS; therefore, the image may contain fainter point sources that do not appear in the catalogue.



(a) Simulated OmegaCAM image

(b) LIGHTS image of NGC 4307

(c) Crop used for subtraction

Figure 3.1: Simulated and real image used throughout this research.

4 Identification

The first part of this project researches the detection of point sources. This can be separated in two steps: given an input image, we first need to find significant light sources in the image. This task is performed by MTOjects, as described in Chapter 2. Then, we need to determine which of the detected objects are point sources. This process is described in the following chapter.

4.1 PSF fitting

An important measure in the detection of point sources is how well an object fits the Point Spread Function (PSF). The PSF describes how a single point appears in an image, and therefore, a point source should closely match the PSF of an image. While sometimes the PSF of an image is known, this is not always the case. In these situations, we can estimate the PSF using a list of known point sources. Since these point sources should match the PSF, we can extract cutouts from the image at their locations and combine these to find the PSF. The procedure is described in Algorithm 4.1. The input to the function is an image, a list of point sources and the size of the PSF image to extract. Then, we filter the list of stars to include only the brighter stars that do not have any other objects close to them. For each star in the filtered list, we take a cutout and normalize it by the central intensity, so that each stamp has intensity 1 at the centre. Finally, we compute the median of all stamps and return the normalized PSF. While the mean pixel value could also be used, the median is less sensitive to outliers and contamination from nearby light sources.

Algorithm 4.1 Estimate PSF from an image

```

1: procedure ESTIMATEPSF(img, star_list, PSF_SIZE)
2:   filtered_list  $\leftarrow$  filter star_list to include only bright, isolated stars
3:   stamp_size  $\leftarrow$  PSF_SIZE
4:   half_size  $\leftarrow$  stamp_size // 2
5:   stamps  $\leftarrow$  empty list
6:   for each star in filtered_list do
7:     (x,y)  $\leftarrow$  integer coordinates of star
8:     stamp  $\leftarrow$  subimage of img centered at (x,y) with size (stamp_size  $\times$  stamp_size)
9:     if stamp is near edge then
10:      continue
11:     end if
12:     Normalize stamp by dividing by its central pixel value
13:     Append stamp to stamps
14:   end for
15:   psf  $\leftarrow$  median of all stamps (pixel-wise)
16:   Normalize psf so that sum of all pixels is 1
17:   return psf
18: end procedure

```

Once we have our PSF, we can fit it to objects in the image to identify point sources. Initially, we fitted the PSF image to each object directly, by making a cutout of the image from each object and computing the squared error for each pixel. However, this method is sensitive to noise and light from nearby sources, and showed no separation between galaxies and point sources. Therefore, a different approach was chosen, which is more robust and not affected by outliers.

As a first analysis, we plotted the light profiles of objects in both classes. From a cutout identical to the PSF size, a distance map was made, containing the distance to the centre for each pixel. This map was converted to a set, allowing us to find all pixels at the same distance from the centre for an object. Then, for each object, we mapped each distance to the median value of its corresponding pixels. Finally, all values are normalized by the central intensity of the object. As a result, we have a smooth light profile for each object, going outwards from the centre. Figure 4.1 shows a plot of 285 point sources (yellow) and 100 galaxies (blue), along with the PSF (red), which has been processed in the same way as the objects. We can observe that the point sources all have very similar light profiles, while the galaxies vary heavily and no pattern can be observed in their light profiles. Furthermore, the estimated PSF follows the shape of the point sources. This implies that comparing these processed light profiles of objects with the PSF should give a much better separation than before; the point sources closely match the PSF in the plot, while the galaxies should have a much larger error. Therefore, our next step is fitting the PSF to objects using these light profiles. For each object, we define the error as the residual sum

of squares:

$$RSS = \sum_{r \in R} (I_{\text{obj}}(r) - I_{\text{PSF}}(r))^2 \quad (4.1)$$

Here, $I(r)$ is the median normalized pixel value at distance r from the centre, and R is the set of unique distances from the centre that fall within the PSF size.

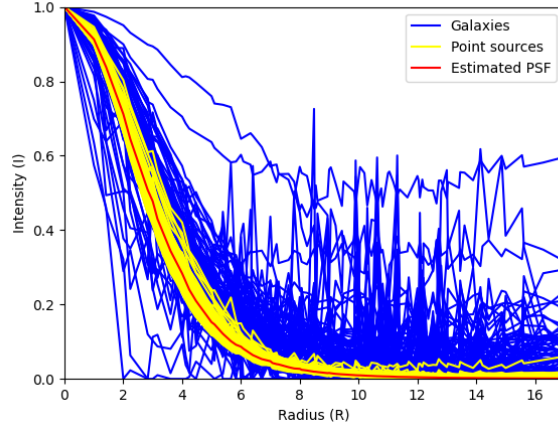


Figure 4.1: Intensity relative to centre against distance from the centre, for 285 point sources, 100 galaxies and the PSF.

4.2 Simulated data

For our first tests, we use code provided by Mohammad Faezi that generates images containing simulated galaxies and stars, as described in Chapter 3. We generated an image containing 285 point sources and the same amount of stars to perform star-galaxy classification. Along with the image, a CSV file was generated containing the location and type of each generated object. Then, we processed the image using MTOBJECTS, and read the CSV file to match detected objects with their type. This gives us a ground truth, that we can use to analyse the features of galaxies and stars and evaluate the performance of classification methods.

MTOBJECTS already calculates several parameters for each object, related to their shape and morphological profile. This is done after preprocessing the image and identifying the objects, using the pixel positions and values of each object. For example, the major and minor axes are computed using PCA on intensity-weighted pixel coordinates, which can then be used to compute the ellipticity A/B . Radii related to light distribution are computed by first sorting the list of all pixel values within an object (in descending order) and then computing their cumulative sums. Then, the first instance in the list of cumulative sums is found that exceeds the threshold of 10%, 50% or 90% within the total flux. The index of this instance is the area within which the given threshold of the total flux is contained, and this area is then converted to the radius. For the threshold of 50%, this results in the effective radius R_e . Similarly, the radius R_{fwhm} is computed by finding the first pixel value in the sorted list which is half the maximum value.

In an initial data exploration, we plotted the distributions of six parameters for stars and galaxies: The effective radius R_e , and similarly, the radius R_{10} in which 10% of the total flux is contained; the ellipticity A/B ; the log transform of the central intensity I_0 ; the radius R_{fwhm} at which the intensity is half the central intensity, or $I_0/2$; and finally, the log transform of the RSS of fitting the PSF to the object. The distributions are shown in Figure 4.2. The PSF fitting works well, with clear separation between point sources (low error) and galaxies (higher values). For other features, the separation is clearest in R_{10} , where all stars have very low values. Tight clusters are seen in R_e and $\log(I_0)$, although galaxies overlap and have both lower and higher values. R_{fwhm} and A/B have low values for stars, but also a lot of overlap and are only useful for reducing the number of point source candidates. For the radii R_e , R_{10} and R_{fwhm} , we also computed their values as fractions of the total radius of the object; however, these features also correlate with I_0 and did not improve the separation between categories.

In the next step, we tested several unsupervised classification methods: K-means, Gaussian Mixture Models (GMM) with covariance types full, spherical and tied, DBSCAN with eps values ranging from 0.1 to 0.7, and Spectral clustering. All of

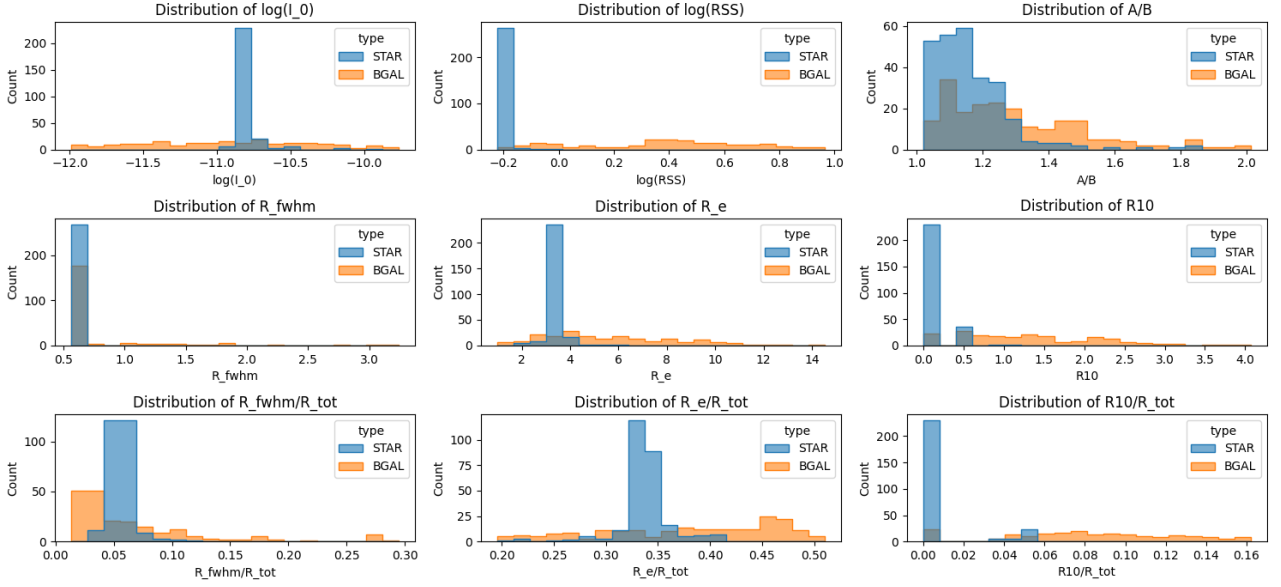


Figure 4.2: Feature distributions for 100 stars and 100 background galaxies in a simulated dataset

| Method | Features | F1 | Precision | Recall |
|-----------------------------|-----------------------------|--------|-----------|--------|
| GMM (full) | $\log(RSS)$ | 97.35% | 97.87% | 96.84% |
| DBSCAN ($\epsilon = 0.2$) | $\log(RSS), \log(I_0), R10$ | 96.99% | 97.86% | 96.14% |
| Spectral | $\log(RSS)$ | 96.71% | 95.55% | 97.89% |
| DBSCAN ($\epsilon = 0.1$) | $R_e, R10$ | 95.16% | 93.86% | 96.49% |
| GMM (full) | $\log(I_0), R_e, R10, A/B$ | 93.17% | 93.01% | 93.33% |
| K-means | $\log(RSS), A/B$ | 90.19% | 82.13% | 100% |
| K-means | $\log(I_0), R10, A/B$ | 82.25% | 69.85% | 100% |

Table 4.1: Classification results on simulated data.

these methods were tested with all possible combinations of features. Table 4.1 shows the results for some combinations. As we observed before, the RSS of the PSF fit is a good separator, and with this feature alone, we can already achieve high F1 scores: of all combinations, the best performing classifier is GMM with just $\log(RSS)$ as a feature, reaching an F1 score of 97.35%. Spherical GMM has the same performance, and spectral clustering also performs well, with an F1 score of 96.71% using just $\log(RSS)$. DBSCAN performs best with $\epsilon = 0.2$, reaching F1 scores of nearly 97% with various feature sets, all of which include both $\log(RSS)$ and $R10$. K-means performs the worst, reaching 90% F1 at most.

We have also tried to classify without RSS , only considering features that can be extracted directly from the objects. This is useful since it does not require a known PSF, or an existing catalogue of point sources necessary to estimate the PSF. The results are generally good; most methods can achieve F1 scores above 90% with several different feature combinations. K-means performs the worst, while the best performing model is DBSCAN with $\epsilon = 0.1$, reaching an F1 score of 95.16% with features R_e and $R10$. Competitive performances are reached with different feature sets and ϵ values. However, it should be noted that DBSCAN is generally unstable when used on data sets with varying sizes; as the amount of data increases, the density increases as well, which requires a smaller ϵ value in order to distinguish clusters. GMM is more robust to this, and shows a competitive performance for this dataset, reaching 93.17% F1 with features R_e , $R10$, $\log(I_0)$ and A/B .

As described above, Gaussian Mixture Models achieve good performance in classification with and without RSS . Gaussian Mixture Models (GMM) are comparable to K-means, but instead of simply assigning each data point to the nearest cluster, GMMs assume that clusters follow a Gaussian distribution. In GMMs, the Expectation-Maximization (EM) algorithm is used. In the expectation step, data points are assigned a probability of belonging to each cluster. The maximization step optimizes the parameters of the model until the best fit is found. An advantage of GMM over K-means is that it is more

flexible and can handle varying cluster shapes, sizes and densities much better.

From the simulated data, we can conclude that point sources and galaxies are well separable using clustering algorithms, both with and without PSF fitting. Gaussian Mixture Models give the best performance here, while being robust against varying dataset sizes. With $\log(RSS)$ as its only feature, it achieves an F1 score of 97.35%, and without RSS , it reaches 93.17% F1. However, it should be noted that the simulated data is much simpler than a real dataset; furthermore, large astronomical images typically contain much more objects, and the data may become too dense to be able to effectively separate data using clustering algorithms.

4.3 Astronomical images

Following the use of simulated data, we performed tests on a real image. We used an image of NGC 4307 from the LIGHTS survey, along with a catalogue of confirmed point sources, as described in Chapter 3. Processing the image in MTOBJECTS resulted in the detection of nearly 35,000 objects. The catalogue contains 285 confirmed point sources, including their location and Gaia g-band magnitude. The magnitude is a measure of how bright an object is, where lower means brighter: a difference of 5 in magnitude means that one object is a hundred times brighter than the other. A restriction on our data is that Gaia has a detection threshold around magnitude $G \approx 21$, while the LIGHTS image contains much fainter objects. This means that the image can contain point sources with a magnitude higher than 21 that cannot be seen by Gaia, and therefore do not appear in the catalogue. The catalogue should therefore be treated only as a subset of confirmed point sources.

The Gaia catalogue was provided as a FITS table, and MTOBJECTS was modified to process this table along with the image. Initially, both the image and catalogue are read from their respective FITS files. After MTOBJECTS has processed the image and created a list of objects extracted from it, the Gaia catalogue is matched with the objects found by MTOBJECTS. For each point source in the catalogue, the nearest object from MTOBJECTS is taken and labelled as a point source, given that the positions of the object and the catalogue entry are close enough. All other objects found by MTOBJECTS, which have not been matched with a point source from the catalogue, are labelled as galaxies, although some of them may be point sources that are not present in the catalogue.

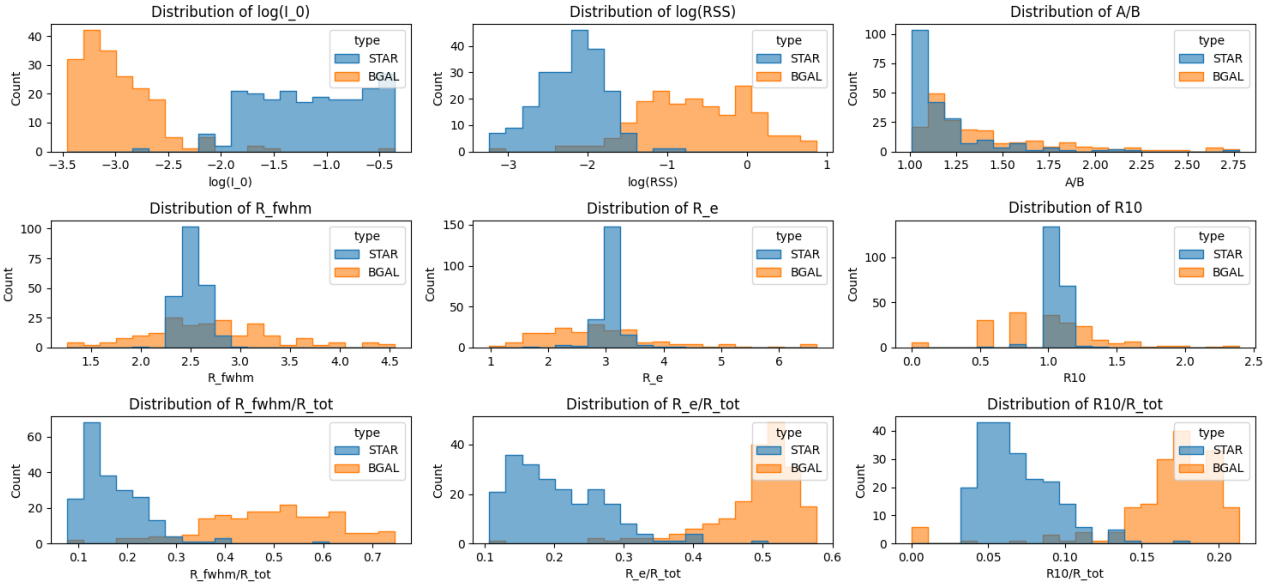


Figure 4.3: Feature distributions for 285 stars and 285 background galaxies in an image of NGC 4307.

After labelling all objects as either point source or galaxy, we can perform two-class classification. Due to the extreme imbalance in the data, we took a random sample of 285 galaxies from the data for our first tests. Plotting the distributions, as shown in Figure 4.3, we observed that they differed from the simulated data: the log transform of the maximum intensity I_0 now shows a good separation between galaxies and stars, although there is still overlap, mainly from stars with lower intensities. Furthermore, tight clusters of stars are observed in R_{fwhm} and R_e , although there is a lot of overlap with

galaxies. The $R10$ feature is less useful, and the ellipticity A/B shows a lot of overlap. Computing the radii R_{fwhm} , R_e and $R10$ as fractions of the total object radius improves the separation, because they seem to be smaller for stars; however, these values correlate with I_0 , because brighter objects have a larger area. The PSF fitting error $\log(RSS)$ also shows good separation between stars and galaxies, although there is some overlap, especially with galaxies having a low error, comparable to stars. It is likely that these galaxies are point sources that do not appear in the catalogue of confirmed point sources.

| Method | Features | F1 | Precision | Recall |
|--------------------|--|--------|-----------|--------|
| GMM (full) | $\log(I_0)$, R_{fwhm} | 98.93% | 98.30% | 99.57% |
| GMM (full) | $\log(I_0)$, $\log(RSS)$, R_{fwhm} | 98.92% | 98.71% | 99.14% |
| Spectral | $\log(RSS)$, R_{fwhm} , R_e , $R10$, A/B | 93.72% | 92.44% | 94.83% |
| DBSCAN (eps = 0.1) | R_{fwhm} , R_e , $R10$ | 92.31% | 91.53% | 93.10% |
| GMM (spherical) | $\log(RSS)$, $R10$ | 91.31% | 85.93% | 97.41% |

Table 4.2: Classification results on a sample of real data, with various methods.

Again, we tested unsupervised classification methods, starting on the balanced dataset. A subset of the results is shown in Table 4.2. The highest performance is achieved by GMM with features $\log(I_0)$ and R_{fwhm} , reaching an F1 score of almost 99%. Almost the same performance is reached with features $\log(I_0)$, R_{fwhm} and $\log(RSS)$. However, these results are misleading, since the Gaia catalogue only contains point sources with magnitudes up to 21, while fainter point sources are not in the catalogue, and therefore falsely labelled as galaxies in our data. Therefore, a better approach would be to ignore this feature. Without $\log(I_0)$, the highest F1 score is achieved by spectral clustering with parameters $\log(RSS)$, R_{fwhm} , R_e , $R10$ and A/B , achieving 93.72% F1. The best performance with GMM is achieved using spherical covariance with features $\log(RSS)$ and $R10$, which has a higher recall than spectral, but much lower precision. Excluding both $\log(I_0)$ and $\log(RSS)$, the highest performance is reached with DBSCAN (eps = 0.1) with features R_{fwhm} , R_e and $R10$.

4.3.1 Learning Vector Quantization

A problem with clustering algorithms is that they perform best on balanced datasets with clearly defined clusters. Our dataset contains around 35,000 objects in total, with no clear boundaries between clusters. Running the same classifiers on the full dataset results in F1 scores of 0%, meaning that no separation can be made.

Because of the poor performance of clustering algorithms, we attempted a different approach: supervised classification using Learning Vector Quantization (LVQ), in its basic form LVQ1[16]. More advanced adaptations of LVQ, such as LVQ2.1 or LVQ3, were not considered. While LVQ requires a training phase, it can classify big datasets effectively. LVQ works with prototypes, which serve as example data points that represent a class. Each data point is assigned the class of its nearest prototype; during training, the prototypes are moved closer to or away from each data point based on whether that point was classified correctly or not. In our approach, we start with 100 samples from each class and define two prototypes per class. They are initialized using K-means for each class, setting a prototype at the centre of each cluster. LVQ is then trained with 100 iterations and a learning rate of 0.01. These parameters were chosen as initial settings and gave decent results, so they were not tuned any further. In each iteration, for each data point the nearest prototype is selected. If it is the same class as the data point (based on its known label), it is moved closer to the data point, with the distance multiplied by the learning rate. If the prototype is not the same class, it is moved away from the data point. After training, classification is done by assigning each data point to the same class as its nearest prototype.

The process of LVQ training and classification is repeated for all possible feature combinations. The results are summarized in Table 4.3 and show good performance even on the full dataset. The F1 scores are low due to the detection of point sources that are not present in the catalogue, leading to a low precision; however, very high recall scores are reached. The highest F1 score was reached with features $\log(I_0)$, $\log(RSS)$, R_{fwhm} and A/B , with a high recall of 99.14%. An even higher recall was reached with $\log(I_0)$ and R_{fwhm} , which was also the best performing feature set in the previous experiment with GMM. However, because the $\log(I_0)$ feature is misleading due to the limited Gaia catalogue with respect to fainter point sources, we filtered out all results with this feature. The most useful feature is the PSF fitting error $\log(RSS)$, which scores the highest F1 on its own compared to other features, with a recall of 95.69%. A higher F1 score is only reached when adding $R10$, although this decreases the recall. The combination of $\log(RSS)$ and A/B ranks third place in F1, with a higher recall of 96.55%.

| Features | F1 | Precision | Recall |
|---------------------------------------|--------|-----------|--------|
| $\log(I_0), \log(RSS), R_{fwhm}, A/B$ | 31.31% | 18.59% | 99.14% |
| $\log(I_0), R_{fwhm}$ | 23.00% | 13.00% | 99.57% |
| $\log(RSS), R10$ | 11.17% | 5.93% | 94.83% |
| $\log(RSS)$ | 9.40% | 4.95% | 95.69% |
| $\log(RSS), A/B$ | 9.05% | 4.95% | 96.55% |
| $R_{fwhm}, R_e, A/B$ | 4.33% | 2.22% | 89.22% |
| $R_{fwhm}, R_e, R10$ | 2.61% | 1.32% | 99.14% |

Table 4.3: Classification results on the full dataset using LVQ.

When excluding $\log(RSS)$ as well, performance decreases. Using the same feature set that scored the highest F1 in the previous experiment using DBSCAN, a very high recall is achieved. However, the precision is significantly lower compared to feature sets with $\log(RSS)$. Although the precision and F1 scores are not very meaningful on their own, very low precision scores may indicate a higher amount of false positives. The highest F1 score without $\log(RSS)$ is obtained with R_{fwhm}, R_e and A/B , but this combination has a significant drop in recall to 89.22%.

From the results described in this section, we can conclude that the PSF fitting error $\log(RSS)$ is an important feature in classification. On the other hand, while $\log(I_0)$ appears to give good results, it is misleading due to the magnitude limit in the Gaia catalogue, and should therefore be ignored. As our final model, we chose LVQ with the features $\log(RSS)$ and A/B , because this combination resulted in a high recall with decent F1, while not relying on the misleading feature $\log(I_0)$. Still, while this feature set gives good results, it is not necessarily the optimal combination, since the limitations of the dataset make it challenging to accurately measure classification performance.

5 Removal

After identifying the point sources in an image, the next step is removing them. As described in Chapter 4, a point source should closely match the PSF, and therefore we can expect that point sources can be removed by subtracting a scaled PSF from the image. The next chapter covers our experiments and results on PSF subtraction on both simulated and real data.

5.1 Simulated data

In a first test, we use the simulated data from Section 4.2 and try to remove point sources in the image. Using Algorithm 4.1, we have obtained a normalized 25×25 PSF image. Then, for all point sources, the PSF image is scaled by the central pixel intensity and added to a new image that will contain all point sources. Finally, the point source image is subtracted from the original image, leaving us with a residual image with point sources removed. A part of the image is shown in Figure 5.1, with the original image on the left, and the residual image in the middle, which is the original image after preprocessing by MTOObjects and point source subtraction. The top images are shown in "histogram equalize" (HE) colour scale, which redistributes pixel intensities to increase the contrast, making faint structures and noise more visible. The bottom images are shown in logarithmic scale, which enhances fainter structures without making the bright parts too strong.

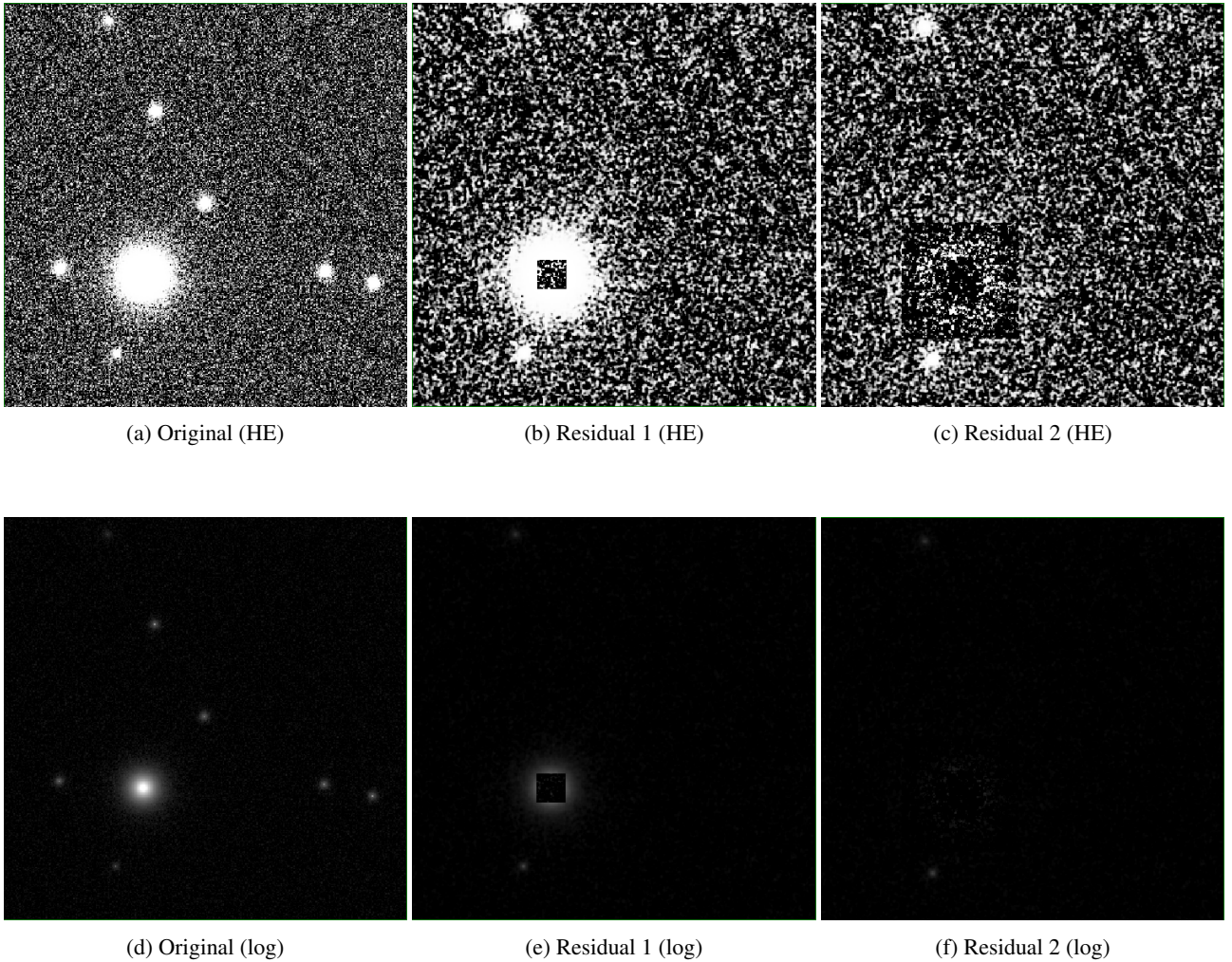


Figure 5.1: Point source subtraction on simulated data.

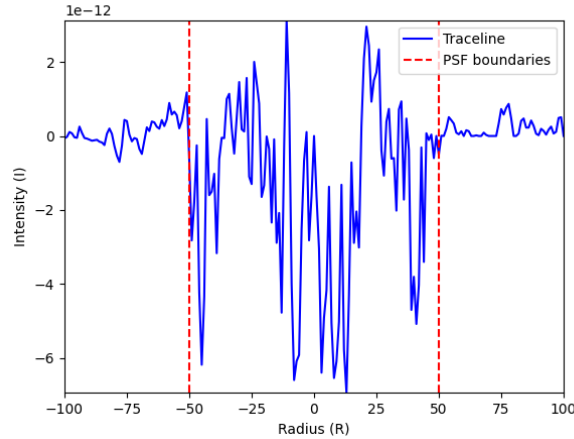


Figure 5.2: Horizontal trace line through subtracted simulated point source.

In the shown result, we can see that our first approach on point source subtraction works well on simulated data: the smaller point sources are removed entirely without visible residual. The larger star has a 25×25 square over the centre, which looks natural. Because of the limited PSF size, the outskirts of the star have not been subtracted. This has been done in a second experiment, which is shown on the right: now, the size of the estimated PSF has been increased to 100×100 , and the star is subtracted entirely. In the logarithmic image, the bright ring around the central 25×25 area has now been removed as well and no residual is visible any more. However, in the HE image, the residual for the large star is slightly darker, indicating over-subtraction in bright stars. Figure 5.2 shows the amplitude of a horizontal trace line through the residual, with higher mostly negative values in the subtracted area. A possible explanation for this could be contamination from other light sources when estimating the PSF, especially in the outskirts. However, overall, the results of our first subtraction tests are acceptable: point sources have been subtracted with no visible residual in the logarithmic image, and only very bright stars leave negative residuals, whereas smaller stars seem to be subtracted smoothly.

5.2 Astronomical data

Subsequent to the results on simulated data, we tested the same approach on a cutout of the image used in Section 4.3. However, the results were very different and are shown in Figure 5.3: a residual can be seen in the upper part of the image, while negative pixel values appear below the object centre. This suggests that the PSF might have been subtracted at the wrong location. In the simulated data, all point sources are centred exactly around one pixel. In real images however, objects may have sub-pixel offsets; in other words, the exact centre of the object may be somewhere in-between pixels. Correcting for this is non-trivial: since our PSF is a 25×25 image in the same resolution as the astronomical image, in order to shift the subtraction with sub-pixel offsets, we need to compute a new PSF image.

One way to approach this is by using interpolation. We modified the PSF estimation step to oversample the image by a factor 10 using cubic spline interpolation. For the sub-pixel offset, we use the exact object centre position as calculated by `MTOjects`: this is a weighted average of pixel values and coordinates. The oversampled PSF image is then shifted by the sub-pixel offset to align it with the object, after which it is downsampled to the original pixel scale and subtracted from the image. This approach slightly improves the results, but a residual is still visible.

5.3 Modelling the PSF

The use of a discrete PSF image, estimated from known point sources, comes with limitations. As described above, the PSF image is discrete, and interpolation is necessary to handle sub-pixel offsets. This may introduce errors and make PSF subtraction less accurate. Furthermore, the PSF image is finite, and can only be used to subtract the central region of an object. To subtract fainter outskirts, interpolation has to be used, or a larger image must be estimated from the known point sources. However, this becomes increasingly inaccurate as the distance from the centre grows, because the signal-to-noise ratio decreases and there is a higher chance of other light sources interfering. These problems could be solved by modelling a function of the PSF, which can simply calculate the intensity from a given distance. Such a function is infinite and continuous, and hence does not have the limitations of a PSF image. A widely used function to model the

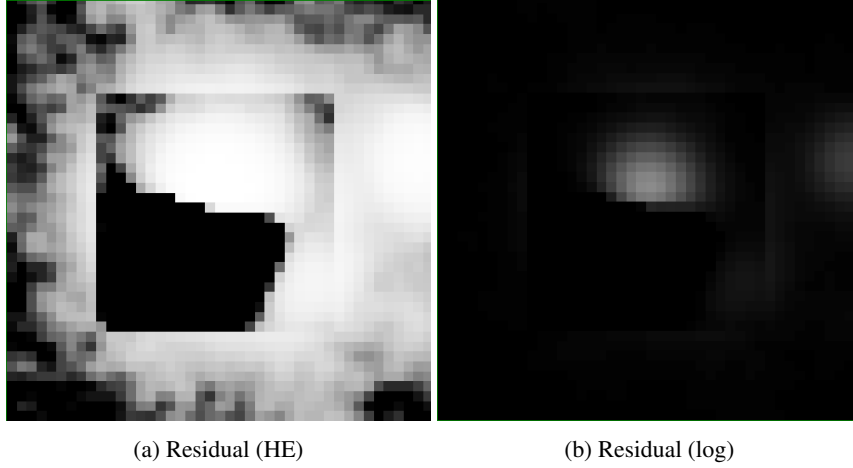


Figure 5.3: First approach on point source subtraction on real data.

PSF is the Moffat distribution, which can portray the wings of the PSF more accurately than other distributions such as Gaussian[17]. Therefore, our next step is fitting a Moffat function to point sources.

A Moffat function can be described as follows:

$$I(r) = I_0 \left[1 + \left(\frac{r}{\alpha} \right)^2 \right]^{-\beta} \quad (5.1)$$

This is a circular Moffat function, in which the intensity is computed solely with the radius r from the object centre. The intensity is scaled by the central intensity I_0 of the object, and the parameters α and β control the PSF width and steepness, respectively.

We started by fitting a Moffat function to each point source in the catalogue during the PSF estimation step and then taking the median parameters for α and β , but to improve accuracy, we also tuned the parameters further to each object during the subtraction step. To tackle the problem of incorrect positioning of the PSF on the object, we introduced a loop to correct the position based on the residual: by shifting the position of the centre towards the positive residual after subtraction, asymmetric residuals could be minimized.

This approach improved the results significantly, with no large asymmetric residuals, but a much smaller positive residual in the middle. However, the fit is still inaccurate, and some regions are over-subtracted, leading to negative residuals. The full list of results for subtraction can be found in Appendix A. The circular Moffat subtraction experiment was conducted both in the original pixel scale and on the oversampled image, with differing results: in the original pixel scale, a small vertical residual is seen in the middle, accompanied by two over-subtracted regions on both sides. In the oversampled image, only a very small positive residual is seen in the centre, but with a large black ring around it: the central pixels are under-subtracted, but the region around it is over-subtracted.

5.4 Elliptical Moffat PSF

Another approach to point source subtraction has been provided by Minh Ngoc Le, who has written code for point source subtraction using a Gaussian, Moffat or exponential PSF. While her approach is very similar to ours, the implementation is different. In this approach, an elliptical Moffat function is used:

$$I(x, y) = I_0 \left[1 + \left(\frac{x'}{\gamma_1} \right)^2 + \left(\frac{y'}{\gamma_2} \right)^2 \right]^{-\beta} \quad (5.2)$$

In this equation, the radius r has been replaced by the coordinates (x', y') , which are x and y rotated by angle θ . Correspondingly, the parameter α is replaced by two parameters γ_1 and γ_2 . This function can give a better fit if the PSF is slightly elliptical.

Apart from the elliptical formula, Le uses a different minimization approach, with no residual-based position correction, but a minimization function with all parameters (including the position) that uses the Powell method, restricted by bounds provided for each parameter. Furthermore, a smaller 10×10 cutout of the image is used to optimize the parameters. We tested the smaller focus size on our circular Moffat approach as well, however, the differences were very small. Le's approach was tested with different focus sizes on the image in original pixel scale. The results of elliptical Moffat subtraction are included in Appendix A. While the results look slightly better, there are still residuals. A small circular residual is visible in the centre, and around it are asymmetric residuals, both positive and negative.

During further experiments, a small flaw was discovered in Le's code, which affected the results significantly: the bounds provided to the minimization function prevented it from finding the minimum. For the central intensity I_0 , the maximum pixel value in the 25×25 cutout of the point source was given as upper bound. However, due to the sub-pixel offset of the point source, the real maximum intensity I_0 would be found somewhere in-between pixels, and is higher than the maximum pixel value. To address this issue, we increased the upper bound by 10% and re-ran our experiments. This changed the results, showing more positive and less negative residual. While much more positive residual is seen in the logarithmic image with focus size 10, not much residual is visible in the logarithmic image with focus size 25. Nevertheless, the fitted Moffat function is still inaccurate, and we are not satisfied with the results.

In response to our findings, Le stated that her code was intended for the removal of small point sources, and her (original) implementation with Moffat profile worked best for point sources with magnitudes between 22 and 26. For other magnitudes, she suggested the use of other profiles such as Gaussian and exponential, but a residual would still be left in the central region. For our next experiment, we searched for point sources with the highest magnitudes in the provided Gaia catalogue, and ran Le's elliptical Moffat subtraction code on a point source with magnitude 20.8. This is shown in Figure 5.4. While this result looks better than the previous ones, a residual can still be seen, with a small black ring around it. The residual is smaller, but this is logical, since the point source is also much fainter. When zooming out further in the resulting image, the residual of the subtracted point source can still clearly be spotted. Figure 5.5 shows the subtraction of an even fainter point source, which does not appear in the Gaia catalogue, but was classified as a point source by our code. In this image, the subtraction seems to succeed, and no visible residual can be seen.

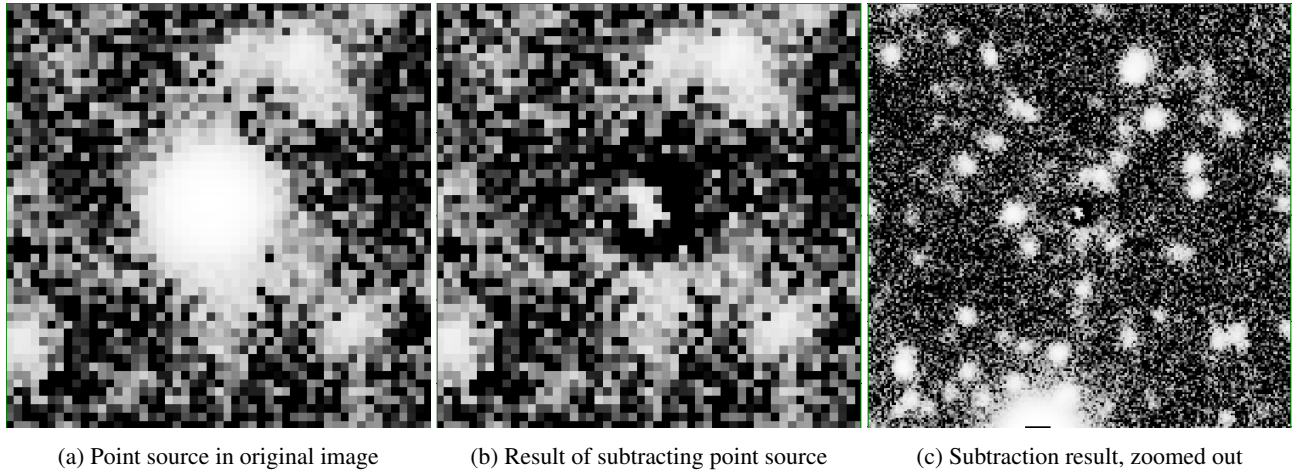


Figure 5.4: Subtraction of a fainter point source 1 with magnitude 20.8.

Similarly to the residual of our simulated star, we created trace lines through the residuals of subtracted point sources, which are shown in Figure 5.6. The figure on the left is the trace line from the brighter star on which we performed most of our experiments, the results of which are shown in Table A.1. This trace line was taken from the result of Le's original approach, with focus size 10. The second image shows the trace line of the fainter point source, with magnitude 20.8. In both lines, higher amplitudes can be seen within the subtracted area. The image on the right shows the trace line of the second faint point source we subtracted. In this point source, we saw no visible residual in the resulting image, and the trace line confirms these findings: unlike the other trace lines, no large differences in amplitude can be seen here.

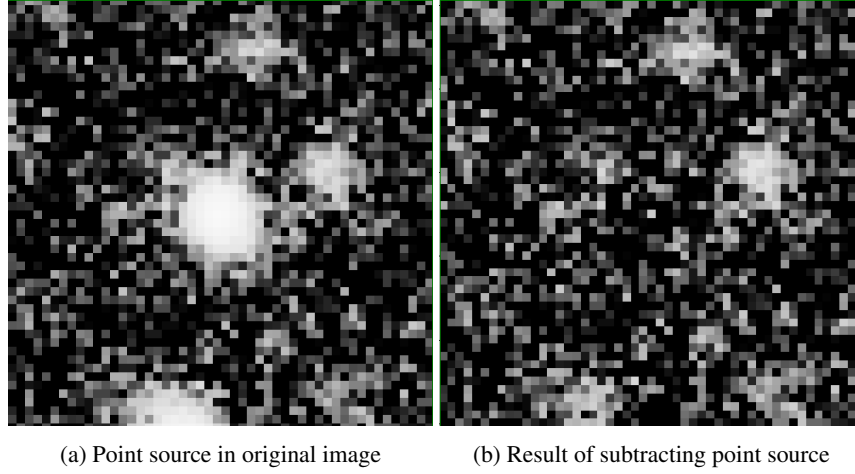


Figure 5.5: Subtraction of an even fainter point source 2.

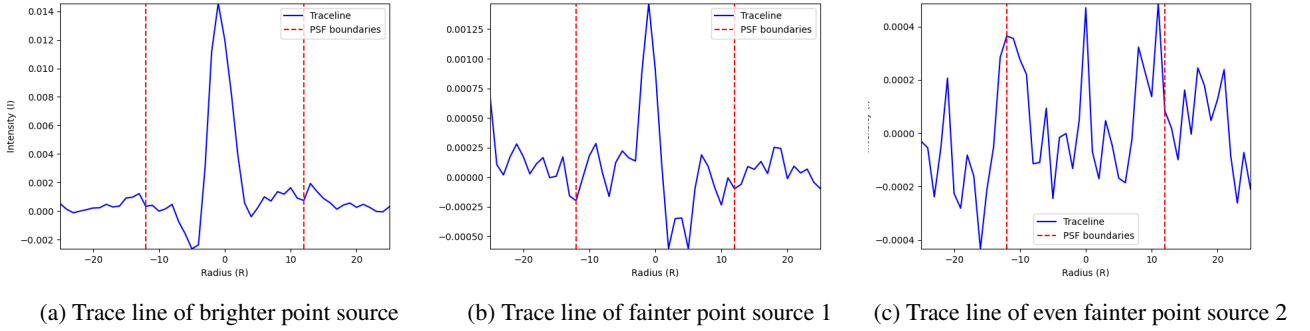


Figure 5.6: Horizontal trace lines showing residual amplitudes of subtracted point sources.

5.5 StarNet++ results

Along with the image of NGC 4307 and a smaller crop, Le also provided the result of processing the cropped image in StarNet++. This is shown in Figure 5.7. While the point sources are being subtracted smoothly, clear gray residuals can be seen with the same shape as the original point source. It seems that, although StarNet++ appears to subtract point sources quite well, it is still not perfect, and residuals remain. On the right, a trace line through the StarNet++ residual is shown. The radius of the gray residual in the image is approximately 15 pixels, and within this distance from the centre, the trace line has a slightly smaller amplitude.

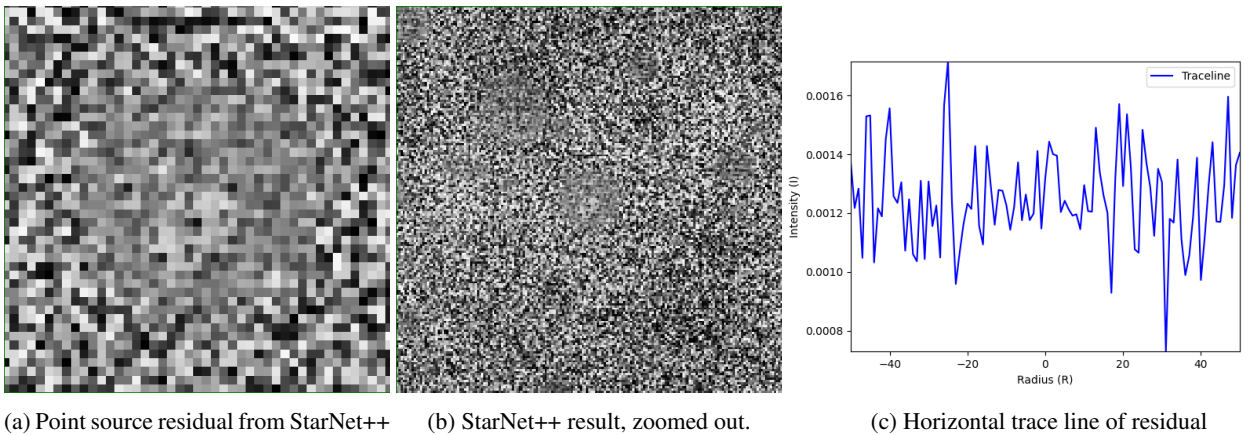
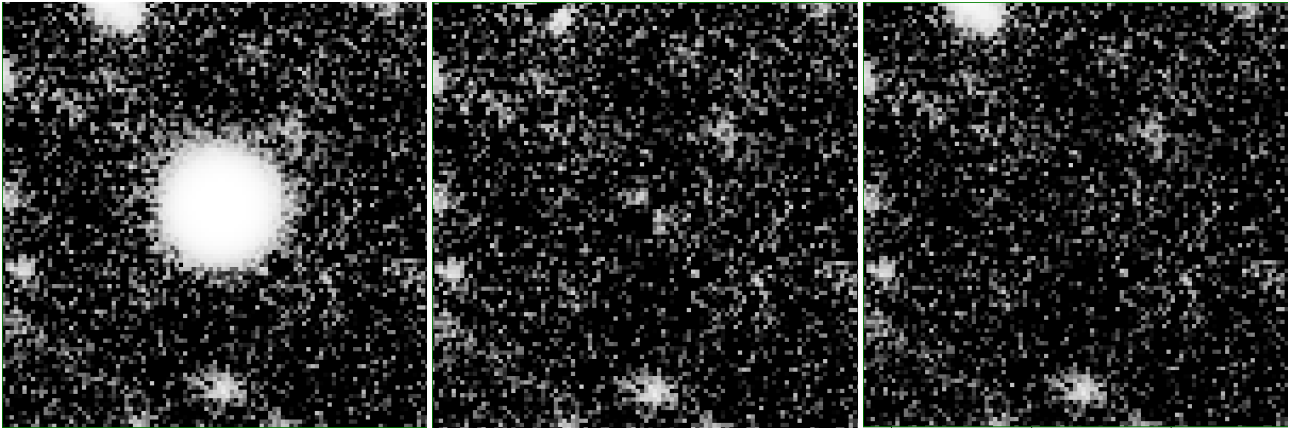


Figure 5.7: Result of processing image with StarNet++

5.6 Adding simulated Moffat stars

While we have not succeeded in accurately fitting and subtracting a PSF from real point sources, we can perform a final test. This involves manually inserting stars into our astronomical image, using a known PSF that we can use in the subtraction step. Then, we can test how well our code for elliptical Moffat subtraction can identify and subtract these point sources, and compare the residual to the original image. Although we have already performed experiments with simulated data, this experiment is different. Here, we have a real astronomical image as a starting point, only inserting a simulated star into it. Furthermore, instead of subtracting a discrete PSF image, we fit a continuous Moffat function to it and use this fitted function to subtract the point source. Another important difference is the placement of the star: when moving on from simulated to real data, sub-pixel offsets proved to be a significant challenge in fitting and subtracting the PSF. Therefore, we will not place the point source centred on a pixel, but add a sub-pixel offset to its position.

In our experiment, we add a point source at position $(1600.323, 1065.716)$, spread out in a 200×200 grid around it. The point is slightly elliptical, with $\gamma_1 = 5.1$, $\gamma_2 = 5.2$, $\beta = 3.0$, $\theta = 1.5$ and central intensity $I_0 = 0.45$. The results are shown in Figure 5.8. On the left, the modified image is shown with a point source inserted. After running Le's original code, we could not find a good fit, and a residual was seen after subtraction. Experiments with this code led us to discover the problem with too restrictive bounds described in the previous section. After correcting the bounds and re-running the code, we obtained the result shown in the middle image. For comparison, the original image before modification is shown on the right. It can be seen that the subtraction of this point source works quite well; the point source is removed with almost no visible residual. The sub-pixel offset, which was not present in the simulated image and proved difficult when moving to real data, is handled quite well in this case. In comparison with the original, the central region of the subtracted point source looks slightly different, with some very small residuals visible. However, small residuals like this are likely inevitable, and overall we believe this experiment to be successful.



(a) Inserted Moffat point source

(b) Result of subtracting Moffat point source

(c) Original image before insertion

Figure 5.8: Astronomical NGC 4307 with simulated Moffat star inserted.

6 Discussion

The first part of our project focused on the detection of point sources. Firstly, we estimated a PSF of the image based on a set of known point sources, and used this PSF to create a 1D light profile with intensity relative to distance from the centre. We used this light profile to determine how well each object matches the PSF, and used the RSS of this profile along with several other parameters, such as the effective radius and ellipticity, to create a binary classifier to separate galaxies and point sources. Due to limitations in the data, it is hard to measure the performance of these classifiers on real data; however, within these limitations, the PSF RSS seemed to separate known point sources from the rest of the data quite well, and we managed to achieve high recall scores in point source detection. Although improvement may still be possible, we are satisfied with the results and consider this part of our research successful.

In our experiments with point source subtraction, we have observed varying results. With simulated data, we achieved successful point source subtraction with a discrete PSF image that was estimated based on known point sources. However, subtracting point sources in real astronomical images proved much more challenging. In a real image, point sources may not be centred exactly on pixels, in contrast to the simulated image. These sub-pixel offsets have to be taken into account for accurate subtraction, but this is non-trivial when working with a discrete PSF image, especially if the PSF has the same pixel scale as the data. Interpolation is a possible solution and improved our results, but residuals could still be seen.

Instead of a discrete PSF image, a continuous function can be fitted to the PSF and used for point source subtraction. We used two different implementations of Moffat subtraction code, one of which was provided by Le. The results were varying, but none of the methods resulted in a perfect fit. Le stated that her implementation works accurately on fainter point sources, and suggested the use of other functions for different magnitudes. While experiments confirmed that the results of subtraction improved with fainter sources, we do not believe that the fit is actually better. As described in Chapter 3, this image was obtained with CCDs, which are almost perfectly linear. This means that the recorded brightness is proportional to the amount of incoming light, and therefore, the magnitude of a point source should not affect the shape of the PSF in any way. We should be able to find a single PSF for the entire image, which can be scaled up and down as necessary and fit well to brighter as well as fainter point sources. In fact, brighter sources have a higher signal-to-noise ratio, so we would expect to obtain better fits with brighter sources. While our experiments with two fainter point sources confirmed Le's theory that these sources can be removed more effectively with less residual, we still observed a residual in one of them. We believe that the fit is not any better for fainter point sources and a residual will always remain. However, as the point sources become fainter and the signal-to-noise ratio decreases, at some point the residual will not be visible any more. Still, the residual does exist. Experimenting with Gaussian and exponential profiles, as Le suggested for brighter sources, did not improve our results at all.

As a final experiment, we inserted a point source with elliptical Moffat PSF into our astronomical image, to test whether the fit would be better on this point source. We were able to subtract this point source successfully with no visible residual using Le's code with a small correction. Our findings suggest that it is possible to subtract point sources accurately if the PSF is well-known, but a Moffat profile is not a good fit for the image that we worked with. As described in the original paper, the Moffat function was determined empirically[17], based on a chosen set of images. While the Moffat function may be a good approximation, our experiments show that it is not a perfect fit when subtracting point sources. Therefore, we believe that the best solution to this problem would be to find a continuous function that is a better fit to the PSF of our image. The results of StarNet++ strengthen our conclusion that point source removal is a hard task; while StarNet++ uses a very different approach, this approach also leaves residuals of the point sources, with their positions and shapes clearly visible.

Whether the results by our experiments and other approaches such as StarNet++ are good enough, remains a point of discussion. While residuals are left behind after subtraction, they tend to be small, and most of the light of the point sources is subtracted successfully. The tool by Ashish Patil described in Chapter 2 also leaves residuals for brighter stars, like many other tools. However, as shown in an example image by Patil, the visibility of extended structures like nebulae is greatly improved, despite the presence of point source residuals. If the goal is to suppress most of the point source light to enhance extended structures, without requiring complete and accurate subtraction, this research makes an important contribution towards that goal.

7 Conclusion

In this thesis, we have attempted to detect and subtract point sources in astronomical images, using MTOjects. The successful subtraction of point sources can help making fainter structures more visible. To begin, we estimated the PSF from the image using a catalogue of known point sources. This estimation is good enough for detection, but could be improved for more accurate subtraction.

The first half of the project focused on the identification of point sources. Using MTOjects, we can already process an image to obtain a catalogue of light sources with relevant properties related to shape and light distribution. Initially, we researched point source classification using these properties, and later, we used the error of PSF fitting as an additional feature. Classifying objects without the PSF error worked reasonably well on simulated data, but questionably in real astronomical images. Limitations in the available data posed an additional challenge in our experiments. When using the PSF error, classification achieved good results, with high recall scores in both simulated and real data.

In the second half of the project, we researched point source subtraction using the estimated PSF. While our first experiments with simulated data gave promising results, subtracting point sources in real data is much harder. Several different approaches were used, using an oversampled PSF directly estimated from the image and with fitting a Moffat function to the point sources. However, neither of the experiments resulted in perfect subtraction. In a final experiment, we were able to successfully subtract a simulated star with Moffat profile that we inserted into the image. This implies that good subtraction of point sources should be possible, given that the PSF is well-known.

7.1 Research Questions

With the results of our experiments, we can now answer the following research questions:

7.1.1 How can we effectively detect point sources in astronomical images?

Using a catalogue of known point sources, we can estimate a PSF from the image that is good enough for detection. This can be done by extracting a stamp from each known point source, and taking the pixel-wise median as the PSF stamp. Using the estimated PSF, we can calculate the residual sum of squares (RSS) between the PSF and a cutout of a point source to see how well the point source matches the PSF. This error can then be used in classification, using an algorithm such as GMM or LVQ to classify point sources. With this, we achieved good results on the detection of point sources.

7.1.2 Can we use the PSF to accurately subtract point sources from astronomical images?

Our experiments have shown that successful subtraction can be performed if the PSF is well-known. After inserting a simulated star with Moffat profile into a real astronomical image, we used code provided by Minh Ngoc Le to fit a Moffat function to the image, and could subtract it with very little visible residual. However, when the PSF is not known, subtracting point sources is very challenging, and while the PSF can be estimated, subtraction will often leave residuals. Therefore, the answer to this question is that if the PSF is well known, successful subtraction should be possible, by using minimization algorithms to optimize the position and scale at which the PSF should be subtracted. Nevertheless, even with subtraction that is not fully accurate, it is possible to use the PSF to suppress most of the light from point sources, which can significantly enhance the visibility of extended structures.

7.2 Future Work

In this project, we have obtained good results on the detection of point sources in both real and astronomical data. Due to limitations on the data, it was hard to measure the performance of classification on real data, especially the precision; however, very high recall scores were achieved. The subtraction of point sources proved to be a much harder challenge. While the subtraction of simulated point sources with well known PSF was successful, experiments on real data resulted in visible residuals. To improve this, future work could research on improvements to PSF estimation and fitting. For instance, the Moffat function might not be good enough for images like the one used in this project, but there might be other functions that provide a better fit. Better PSF fitting could improve both the detection and subtraction of point sources. One way to approach this could be to construct a PSF based on the properties of the telescope used, although this would need to be re-done for images of other telescopes.

Another approach that could be investigated is whether the max-tree constructed by MTOjects could be used to remove point sources, since MTOjects can already separate objects using statistical tests. This is, however, not as simple as just cutting out a branch of the tree, since the background should be kept intact. Also, in the implementation used in this thesis, converting from a node in the tree to its corresponding pixels is not straightforward, since MTOjects only stores which object is associated with each pixel. In the simplest possible approach, we could simply replace all pixels belonging to a point source by the background intensity (one node up in the tree), but this would result in a flat residual that is likely still distinguishable as a subtracted point source. Furthermore, MTOjects preprocesses the image with background subtraction and smoothing, so removing point sources from the original image is more challenging.

An additional part of future work could be the automatic retrieval of point source catalogues from Gaia. By integrating automated querying of Gaia over the internet, point source catalogues do not have to be provided as FITS files, but can be downloaded for an image without any manual actions. This can improve the speed and efficiency of image processing, especially with larger datasets.

Acknowledgements

First and foremost, I would like to thank Michael Wilkinson for supervising this project. Michael has strongly supported me throughout the entire project by sharing his knowledge and providing feedback and ideas. Without his excellent guidance, this project would not have succeeded.

I am also grateful to Minh Ngoc Le for providing me with astronomical data to work on, and her code for PSF subtraction which was used as part of my research on point source removal. Her help was of great importance to this project.

Furthermore, I would like to thank Mohammad Faezi for providing me with code to generate simulated galaxies.

Lastly, I would like to thank Kerstin Bunte for co-supervising this project.

Bibliography

- [1] I. Trujillo, J. A. L. Aguerri, J. Cepa, and C. M. Gutiérrez, “The effects of seeing on Sèrsic profiles,” *Monthly Notices of the Royal Astronomical Society*, vol. 321, p. 269–276, Feb. 2001.
- [2] M. A. Alagao, M. A. Go, M. Soriano, and G. Tapang, “Improving the point spread function of an aberrated 7-mirror segmented reflecting telescope using a spatial light modulator,” in *2016 4th International Conference on Photonics, Optics and Laser Technology (PHOTOPTICS)*, pp. 1–8, 2016.
- [3] P. Teeninga, U. Moschini, S. C. Trager, and M. H. F. Wilkinson, “Statistical attribute filtering to detect faint extended astronomical sources,” *Mathematical Morphology - Theory and Applications*, vol. 1, no. 1, 2016.
- [4] P. Teeninga, U. Moschini, S. Trager, and M. Wilkinson, “Bi-variate statistical attribute filtering: A tool for robust detection of faint objects,” in *11th International Conference on Pattern Recognition and Image Analysis*, pp. 746–749, 2013.
- [5] C. Haigh, N. Chamba, A. Venhola, R. Peletier, L. Doorenbos, M. Watkins, and M. H. F. Wilkinson, “Optimising and comparing source-extraction tools using objective segmentation quality criteria,” *A&A*, vol. 645, p. A107, 2021.
- [6] E. Bertin and S. Arnouts, “SExtractor: Software for source extraction,” *Astronomy and Astrophysics, Supplement*, vol. 117, pp. 393–404, June 1996.
- [7] P. Salembier, A. Oliveras, and L. Garrido, “Antiextensive connected operators for image and sequence processing,” *IEEE Transactions on Image Processing*, vol. 7, no. 4, pp. 555–570, 1998.
- [8] D. Lang and D. W. Hogg, “Principled point-source detection in collections of astronomical images,” 2020.
- [9] P. O. Baqui, V. Marra, L. Casarini, R. Angulo, L. A. Díaz-García, C. Hernández-Monteagudo, P. A. A. Lopes, C. López-Sanjuan, D. Muniesa, V. M. Placco, M. Quartin, C. Queiroz, D. Sobral, E. Solano, E. Tempel, J. Varela, J. M. Vílchez, R. Abramo, J. Alcaniz, N. Benitez, S. Bonoli, S. Carneiro, A. J. Cenarro, D. Cristóbal-Hornillos, A. L. de Amorim, C. M. de Oliveira, R. Dupke, A. Ederoclite, R. M. González Delgado, A. Marín-Franch, M. Moles, H. Vázquez Ramió, L. Sodré, and K. Taylor, “The minijpas survey: star-galaxy classification using machine learning,” *Astronomy & Astrophysics*, vol. 645, p. A87, Jan. 2021.
- [10] Bonavera, L., Suarez Gomez, S. L., González-Nuevo, J., Cueli, M. M., Santos, J. D., Sanchez, M. L., Muñiz, R., and de Cos, F. J., “Point source detection with fully convolutional networks - performance in realistic microwave sky simulations,” *A&A*, vol. 648, p. A50, 2021.
- [11] D. Makovoz and F. Marleau, “Point-Source Extraction with MOPEX,” *Publications of the Astronomical Society of the Pacific*, vol. 117, p. 1113–1128, Oct. 2005.
- [12] R. Peletier, E. Iodice, A. Venhola, M. Capaccioli, M. Cantiello, R. D’Abrusco, J. Falcón-Barroso, A. Grado, M. Hilker, L. Limatola, S. Mieske, N. Napolitano, M. Paolillo, M. Spavone, E. Valentijn, G. van de Ven, and G. V. Kleijn, “The Fornax Deep Survey data release 1,” 2020.
- [13] Trujillo, Ignacio, D’Onofrio, Mauro, Zaritsky, Dennis, Madrigal-Aguado, Alberto, Chamba, Nushkia, Golini, Giulia, Akhlaghi, Mohammad, Sharbaf, Zahra, Infante-Sainz, Raúl, Román, Javier, Morales-Socorro, Carlos, Sand, David J., and Martin, Garreth, “Introducing the LBT Imaging of Galactic Halos and Tidal Structures (LIGHTS) survey - A preview of the low surface brightness Universe to be unveiled by LSST,” *A&A*, vol. 654, p. A40, 2021.
- [14] E. Giallongo, R. Ragazzoni, A. Grazian, A. Baruffolo, G. Beccari, C. de Santis, E. Diolaiti, A. di Paola, J. Farinato, A. Fontana, S. Gallozzi, F. Gasparo, G. Gentile, R. Green, J. Hill, O. Kuhn, F. Pasian, F. Pedichini, M. Radovich, P. Salinari, R. Smareglia, R. Speziali, V. Testa, D. Thompson, E. Vernet, and R. M. Wagner, “The performance of the blue prime focus large binocular camera at the large binocular telescope,” *Astronomy and Astrophysics*, vol. 482, pp. 349–357, Apr. 2008.
- [15] Gaia Collaboration, Prusti, T., de Bruijne, J. H. J., *et al.*, “The Gaia mission,” *A&A*, vol. 595, p. A1, 2016.
- [16] T. Kohonen, *Learning Vector Quantization*, pp. 245–261. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.
- [17] A. F. J. Moffat, “A Theoretical Investigation of Focal Stellar Images in the Photographic Emulsion and Application to Photographic Photometry,” *Astronomy and Astrophysics*, vol. 3, p. 455, Dec. 1969.

Appendices

A NGC 4307 subtraction results

This appendix shows the results of point source subtraction using several different approaches as described in Chapter 5.

Table A.1: Point source subtraction results with different methods.

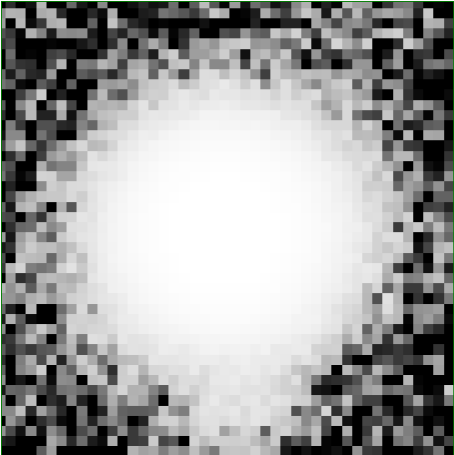
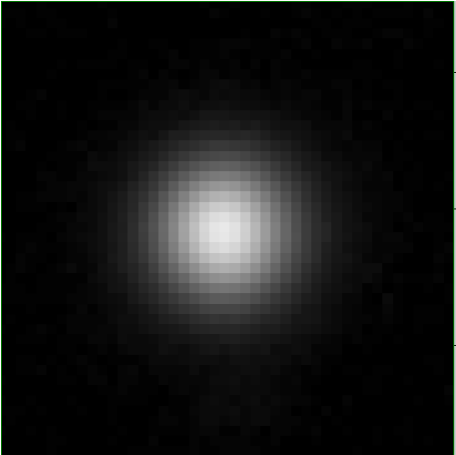
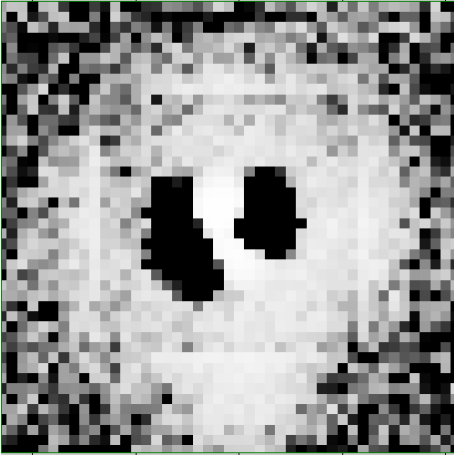
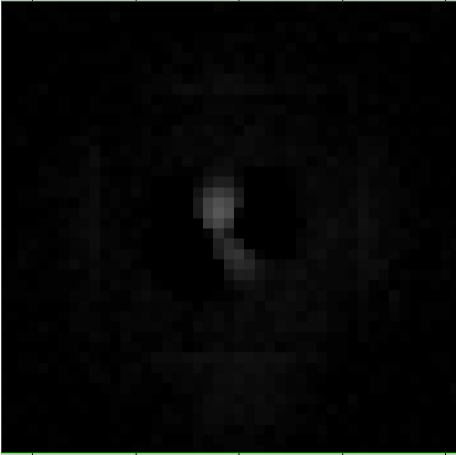
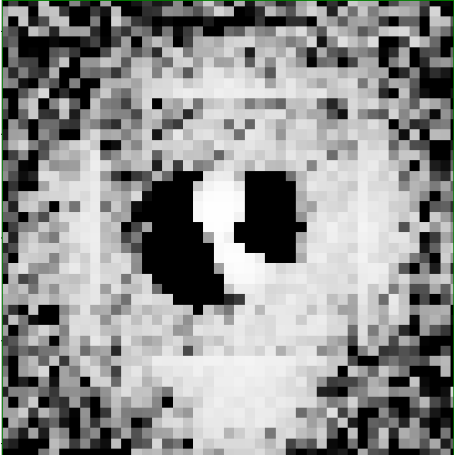
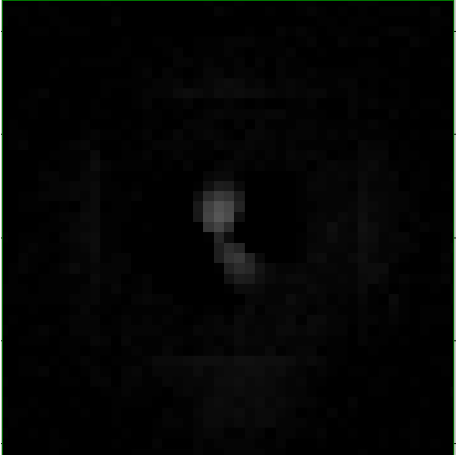
| Focus size | HE | LOG |
|------------------------------------|---|--|
| Original point source | | |
| — |  |  |
| Own approach, original pixel scale | | |
| 10 |  |  |
| 25 |  |  |

Table A.1 – continued

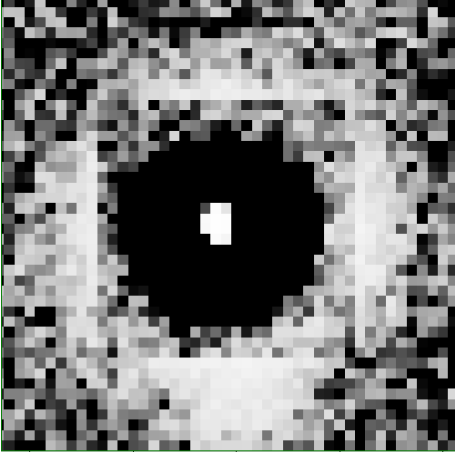
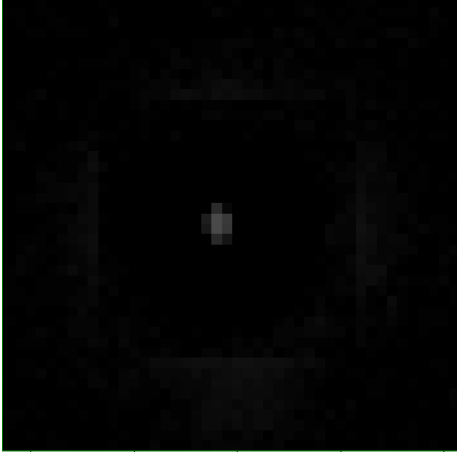
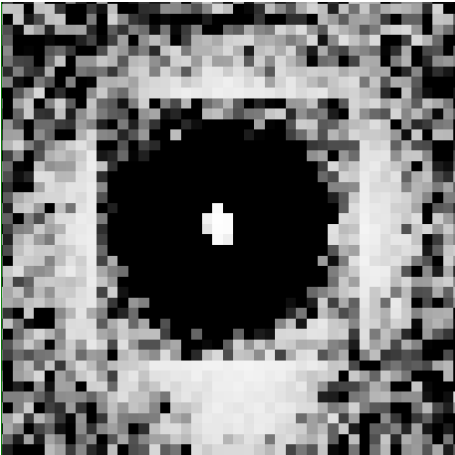
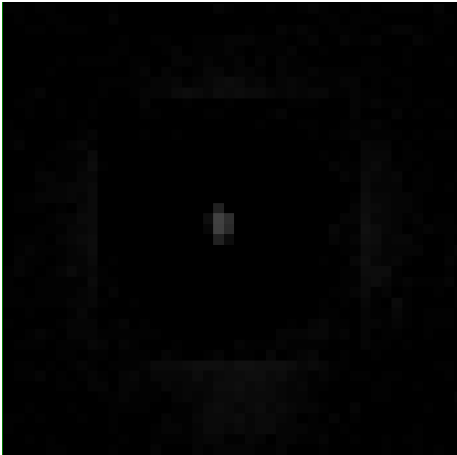
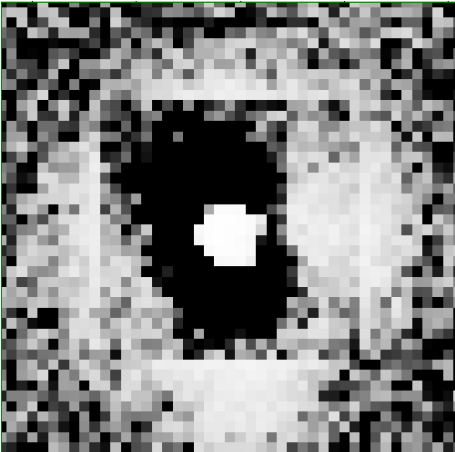
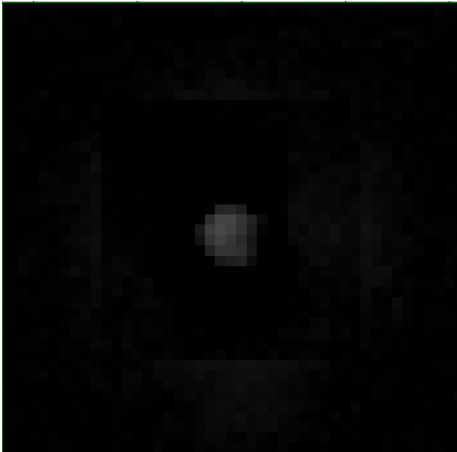
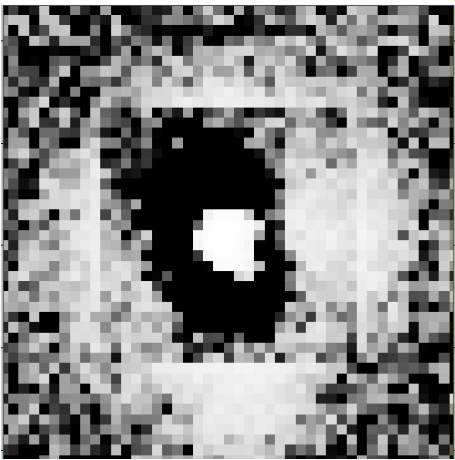
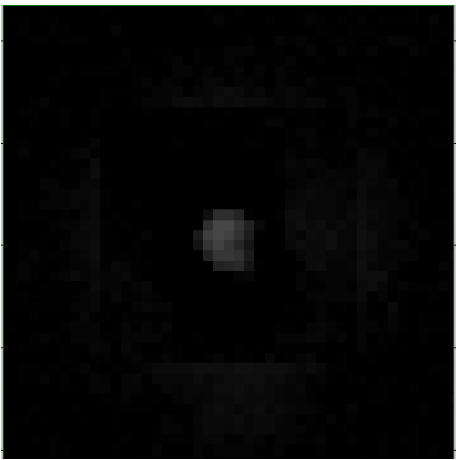
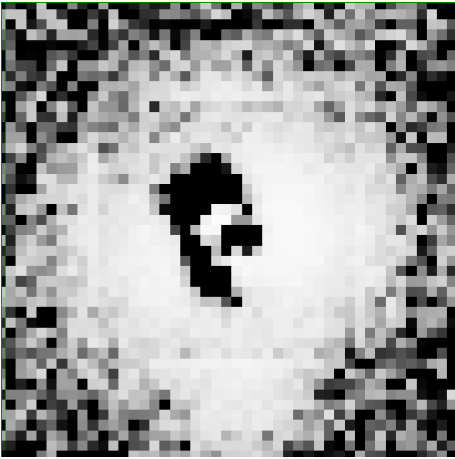
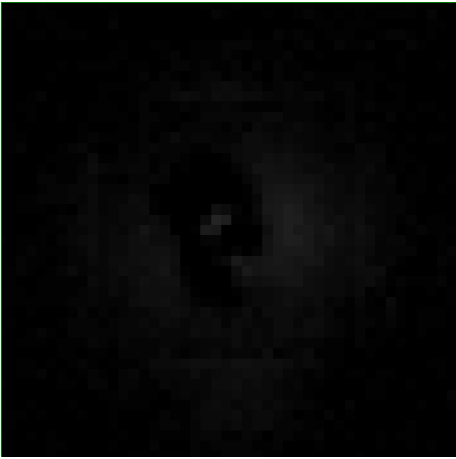
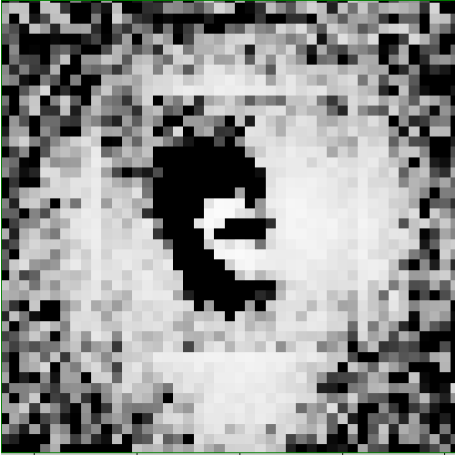
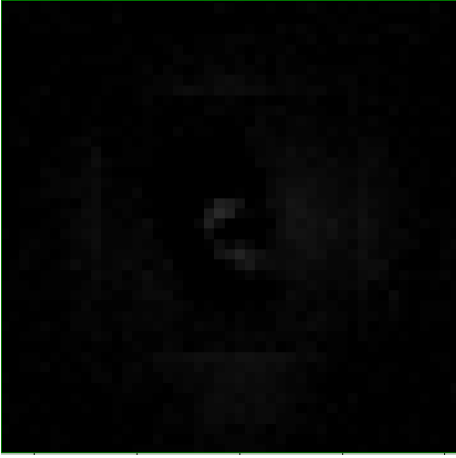
| Focus size | HE | LOG |
|------------------------------------|---|--|
| Own approach, on oversampled image | | |
| 10 |  |  |
| 25 |  |  |
| Le's approach, original code | | |
| 10 |  |  |

Table A.1 – continued

| Focus size | HE | LOG |
|---------------------------------------|---|--|
| 25 |  |  |
| Le's approach, with bounds correction | | |
| 10 |  |  |
| 25 |  |  |