Ph.D. in ASTRONOMY, ASTROPHYSICS and SPACE SCIENCE

CYCLE XXXII

Advanced Image Analysis Techniques for Extragalactic Surveys

Valerio Roscani

A.Y. 2018/2019

Supervisor : A. Fontana

Co-advisors : M. Castellano, E. Merlin

Coordinator : P. De Bernardis

Deputy Coordinator : N. Vittorio

Thesis defended on January 16, 2020
in front of a Board of Examiners composed by:

Prof. Paolo De Bernardis (chairman)
Prof. Carlo Baccigalupi
Prof. Massimo Gervasi
Prof. Pasquale Palumbo

Advanced Image Analysis Techniques for Extragalactic Surveys
Ph.D. thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: valerio.roscani@inaf.it

Abstract

Image analysis applied to astronomy is a vast field. It contains several solutions to handle the many issues inherent to the peculiar nature of astronomical images. The entire area is cross-disciplinary, and it is based on a variety of mathematical and computer science approaches, which are also at the foundation of two techniques that are essential for the analysis of astronomical extragalactic images: "denoising" and "deblending".

On the one hand, the goal of denoising algorithms is to reduce the observational noise intrinsic to the images without losing details (in our case, for example, faint distant galaxies). On the other hand, the purpose of deblending algorithms is to efficiently separate objects that appear overlapped in the image.

State-of-the-art mathematical algorithms for denoising are commonly used in several fields, but there is almost no trace of application to astronomical observations in the scientific literature, in particular concerning optical and near-IR extragalactic observations. These algorithms have the potential to enhance objects detection, granting improved statistics without requiring additional telescope time. Deblending parametric algorithms have been tested, with good, albeit not optimal, results. Many new methods, based on machine learning techniques, have been developed and are now being proposed. Improved deblending algorithms have the potential to enhance high-precision measurements at the basis of cosmological and galaxy evolution investigations.

Therefore, an in-depth study of these techniques is mandatory to assess all their possible advantages and risks quantitatively, and plan their application to forthcoming surveys where unsupervised image analysis will be unavoidable due to the massive amount of data that will be acquired. The goal of this thesis is to test new approaches to the denoising and deblending of astronomical images.

In particular, we found that a small group of denoising algorithms (ATVD, Perona-Malik, Bilateral, and TV Chambolle) enhance objects detection without altering fluxes and shapes. Whereas, tested machine learning techniques (ASTErIsM-DENCLUE, `blend2flux`, and `blend2mask2flux`) accurately separate and recover fluxes of two blended objects, more reliably than the standard approaches.

Contents

List of Figures

List of Tables

Nomenclature

ADU Analog-to-Digital Unit

AFTA Astrophysics Focused Telescope Assets

AIPS Astronomical Image Processing System

ANNs Artificial Neural Networks

BAO Baryonic Acoustic Oscillations

CASA Common Astronomy Software Applications

CCD Charge-Coupled Device

CGI Coronagraphic Instrument

CMB Cosmic Microwave Background

CNNs Convolutional Neural Networks

DE Dark Energy

DECam Dark Energy Camera

DES Dark Energy Survey

DM Dark Matter

EB Exabyte

ESO European Southern Observatory

FITS Flexible Image Transport System

FK5 Fifth Fundamental Catalog

HCIT High Contrast Imaging Testbed

HLS High-Latitude Survey

HST Hubble Space Telescope

IAU International Astronomical Union

ICRF International Celestial Reference Frame

ICRS International Celestial Reference System

IDL Interactive Data Language

IFS Integral Field Spectrograph

IFU Integral Field Unit

IRAF Image Reduction and Analysis Facility

LSST Large Synoptic Survey Telescope

LTD Long-Term Depression

LTP Long-Term Potentiation

MCL MIDAS Command Language

ML Machine Learning

MOS Metal Oxide Semiconductor

NOAO National Optical Astronomy Observatorie

NRAO National Radio Astronomy Observatory

PB Petabyte

PCA Principal Component Analysis

PDE Partial Differential Equation

QE Quantum Efficiency

RL Reinforcement Learning

ROI Region of Interest

SDSS Sloan Digital Sky Survey

SED Spectral Energy Distribution

SL Supervised Learning

STScI Space Telescope Science Institute

STSDAS Space Telescope Science Data Analysis System

TB Terabyte

UL Unsupervised Learning

VLT Very Large Telescope

WFI Wide Field Instrument

WFIRST Wide-Field InfraRed Survey Telescope

WL Weak Lensing

Introduction

Modern astronomy is a scientific field that profoundly relies on digital images. Cutting-edge telescopes and technologically advanced CCDs, allow us to produce high-quality astronomical images. Images are the primary source of information for astronomers, encoding enormous quantities of data that we analyze to understand the Cosmos. Therefore, astronomical images are of vital importance, although different factors limit the amount of information stored (e.g., noise, spatial resolution, atmosphere, exposure time, etc.). New techniques that improve the quality of the available data are always welcomed.

Next-generation extragalactic imaging surveys aim at pushing forward the limits of our knowledge about the Universe, observing the sky at different wavelengths with unprecedented depth, resolution, and/or survey area. The amount of data expected to be produced by these modern missions will officially introduce astronomy in the "Big Data" era, meaning that optimized algorithms of data "reduction" and data "mining" will be essential in the upcoming years.

Advanced image analysis techniques can be applied in astronomy to enhance the quality of the images, improving their scientific value. In this thesis, two kinds of digital image processes are investigated, "denoising" and "deblending". Several techniques, belonging to these two topics, are tested and compared against standard approaches.

Denoising algorithms aim at reducing noise in digital images. In astronomy, many natural sources of noise degrade the image quality, lowering the signal-to-noise ratio of the bright objects and suppressing the faintest ones. State-of-the-art denoising techniques have been already applied in other scientific fields, with impressive results. In this work, we investigate their application to astronomical images, comparing their performances to those of the standard approaches used in our field. In particular, we examine the advantages obtained in the detection of faint galaxies and in the preservation of the shape and flux of bright ones.

Deblending algorithms aim at separating overlapping (blended) objects enabling an accurate recovery of their fluxes and shapes. Indeed, objects blending is the result of the 3-dimensional space projection on a 2-dimensional hyperplane, which is how telescopes observe the sky. The blending of the objects negatively affects the accuracy of measurements yielding to systematic errors. Therefore, blending is not a negligible issue, especially for scientific surveys whose goal is to produce high-precision cosmological measurements. Nowadays, machine learning algorithms are gaining popularity in several scientific fields, astronomy included. Since these approaches are built to find patterns and similarities in the data, they look as optimal candidates for handling blending issues. In this work, we test two different categories of machine learning techniques and compare them to the commolny adopted multi-threshold approach implemented in the well-known code SExtractor. We test the ability of ML techniques at separating blended object and recovering their fluxes. The present thesis is organized into six chapters.

In chapter 1, we discuss digital imaging and instrumentation, highlighting the advantages and the problematics involved. We discuss digital images acquisition and the processing techniques aimed at extracting information from raw data. At the end of the chapter, we introduce the two main topics of this thesis, denoising, and deblending.

In chapter 2, we discuss upcoming extragalactic imaging surveys. The goal is to provide an overview of next-generation telescopes, discussing the "big" amount of data that they will produce, and the scientific advantages expected.

In chapter 3, different algorithms are discussed. In the first part, we introduce several techniques belonging to different families of noise reduction approaches. In the second part, machine learning is discussed, focusing on possible applications to the deblending problem.

In chapter 4, we present an in-depth analysis of different families of denoising algorithms (Anisotropic diffusion, Bilateral filtering, Total Variation, Structure-texture decomposition, etc.) and their application to extragalactic images. The analysis is based on a dataset of simulated and real optical and near-infrared images at different depths and resolutions.

In chapter 5, the impact of objects blending on scientific measurements is discussed in details, and the performances of various "deblenders" is analyzed. In the first part, we introduce an unsupervised learning algorithm (ASTErIsM-DENCLUE). We study its efficiency in separating blended objects compared to the baseline multi-threshold solution. In the second part, two convolutional neural networks (`blend2flux` and `blend2mask2flux`) are compared against the same baseline, testing their performances in the recovery of the correct fluxes of the objects.

In chapter 6, we discuss the results obtained and highlight directions for future developments.

Chapter 1

Imaging the extragalactic sky

The origin of astronomy has profound roots in the history of humanity. The two are entangled and strictly related to one another.

Over the centuries, humankind developed an ever-tighter connection with the sky. Discoveries and inventions by different civilizations mark our history, laying the foundation for a wide variety of theories about the Cosmos. A comprehensive and detailed presentation would be necessary for a complete understanding of the path that lead to the modern astronomical scenario; however, this spectacular part of our past is not the aim of this work. Nevertheless, in the following we will mention a small number of important discoveries, to provide a coarse idea of the history of astronomy progression in centuries.

With this in mind, we introduce the topic of astronomical imaging, from different points of view. In the first part of this Chapter (Sect. 1.1), we briefly discuss the afore-mentioned historical background. The goal is to provide an introduction to modern techniques, used in present-days astronomy. In the second part (Sect. 1.2 - 1.3 - 1.4), we discuss modern detectors, their calibration, and processing techniques with digital images. In the third part (Sect. 1.5 - 1.6 - 1.7), we discuss the astronomical images, the underlying mathematical formalism, and the basic techniques used to extract scientific information.

1.1 Historical Background

In the ancient times, before the invention of the telescope, the only possible observations of celestial objects were those by naked eye. The observed objects were used to perceive the flow of time, determining the seasons' periodicity and the length of the year [55]. With the advancement of human civilizations (in Mesopotamia, China, India, Greece, etc.), measurements of celestial objects position and movement played a growing historical role. For instance, Babylonians discovered the repeating cycle of the lunar eclipses (named *saros* [123]), and cuneiform inscriptions in Sumerian clay tablets (3500-3200 BC) record astrometry studies and observations (e.g., Halley comet observation in 164 BC) [8]. Greek astronomers further contributed to the advancement of astronomical discoveries. Aristarchus of Samos (c.310-c.230 BC), for instance, proposed the first heliocentric model of the Solar System [16] (echoed more than 1000 years later by Copernicus, 1543). Hipparchus (c.127 BC) determined the duration of the seasons and the length of the year, estimated the distances to the Moon and the Sun with outstanding precision, and probably invented the first astronomical tool, the astrolabe; nonetheless, the most astonishing of his discoveries is the precession of the equinoxes [128]. In 1054 AD, on the other side of the globe, Chinese astronomers recorded the explosion of the supernova that created the Crab Nebula[119].

The invention of the telescope profoundly revolutionized Astronomy, and Science in general. In the 17th century, the first low magnification refractive telescopes spread throughout Europe. Galileo Galilei built its own improved version of the telescope with a magnification factor of 20× [86]. Thanks to its invention, he discovered the four largest moons of Jupiter [165], he found that our Moon has craters, and observed the Venus phases and the sunspots, demonstrating the total incompatibility with the geocentric model [5] (introduced by Ptolemy in c. 100-170 AD [47]). Since then, improvements in the technical design and power of telescopes can be paired with impressive astronomical discoveries: for instance, the discovery of Uranus by William Herschel (1781) [117], or the invention of the prism spectroscope by Joseph Fraunhofer, which allowed the study of the Sun chemical constitution (1814)[53].

A further leap forward was the introduction of photographic plates, in the late 19th century. Even if less sensitive than the human eye, they permanently record faint objects by accumulating light, when exposed for long enough time (integrating detector), and far better precision. Using this new technique, Charles Pickering and his assistants classified and analyzed a large amount of data, putting the basis to modern quantitative astrophysics [37]. These classifications led to the understanding of the color-temperature relation of stars, and the discovery of the Cepheid variables[98].

At the beginning of the 20th century major scientific advancements completely changed our understanding of the Cosmos. In 1915, Albert Einstein, with his General Relativity theory [172], revolutionized our comprehension of gravitation and consequently our conceptions about the large-scale structure of the Universe as a whole. A few years later, in 1924, Edwin Hubble used Cepheids stars to prove that M31 (at that time dubbed *Andromeda Nebula*) is not part of our Galaxy [77], thus discovering the existence of other galaxies outside the boundaries of the Milky Way (he did so thanks to a crucial property of Cepheids, their luminosity-period relation discovered by H. S. Leawitt in 1927, which makes them ideal standard candels allowing to easily and precisely estimate their distance). In 1927 Georges Lemaître proved that our Universe is expanding, observing the linear dependence between distant galaxies velocity and their distances from us and estimating the proportionality constant [100]. In 1929, Hubble corrected the proposed value of this constant, which is today commonly known as the Hubble constant [102].

With ever more advanced technologies, modern astronomy (and in particular Cosmology) made terrific progress in the last 60 years. Thanks to radio astronomy and radio telescopes, in 1964, Arno Penzias and Robert Wilson accidentally discovered the Cosmic Microwave Background (CMB) [138], boosting the evidence of a Big Bang model to describe the origin of the Universe. Since 1965, a growing number of Space telescopes have been launched to orbit the Earth (although the very first were actually built for military reasons, and only a few for astronomical purposes). The space telescopes propelled a further leap forward, vastly enhancing the quality of the observations (in space light is not absorbed, blurred and disturbed by the Earth's atmosphere, although this advantage comes at the cost of smaller apertures). In the same years, Willard Boyle and George E. Smith invented the the Charge-Coupled Device (CCD) [155], the electronic detector still used nowadays, scoring the passage from analog to digital.

1.2 From analog to digital

The advent of digital detectors like the CCD replacing photographic plates represented a crucial improvement for the astronomical data analysis. Digital and analog detectors share some similarities: in both cases, a photon, hitting their surface, excites an electron. In the CCD, electrons are attracted by the electrodes and remains in their excited state (see Sect. 1.3.1). In a photographic plate, the electrons tends to radiate their energy, returning to their original state, unless other electrons are excited. In the latter case, they can form an excited stable state. This makes photographic plates "non-linear" detectors, whereas CCDs are classified as "linear" detectors. Linearity of the response is not the only difference. CCDs are generally smaller and more expensive than photographic plates; on the other hand, the latter have a smaller dynamic range (1:10^3 compared to 1:10^4) and a smaller sensitivity [59].

Nowadays, CCDs are the standard devices of choice for astronomical observation. Even ignoring the sensitivity and the dynamic range advantages provided by CCDs, the main reason of their popularity is the unquestionable advantage provided by using digital information. Digital data are easy to share, to reproduce without losing quality, they can be stored indefinitely, and above all they can be easily analyzed by semi-automatic tools and software for data analysis and reduction. Information can be extracted, saving a significant amount of time and resources, building chains of software, called *pipelines*. Modern statistical analysis is indeed specialized in receiving large amounts of data, and, in many cases, the information is analyzed in different facilities around the globe, now quite a common practice for large scientific collaborations.

1.3 Digital Imaging: Detectors and Calibration

The term "digital imaging" refers to the processing, compression, storage, printing, and display of images in digital format. The quality of a digital image usually depends on the sensor used to capture it. Typically, the sensor needs to be calibrated, and the image in its raw state needs to be corrected from defects. The aim of this section is to describe the detectors used to create digital images (the CCDs), and to discuss some standard corrections and calibrations, usually implemented in astronomy.

1.3.1 Charge-Coupled Devices

CCD detectors are used in many fields (e.g., medicine, security, physics, but also standard cameras in cellphones). In astronomy, all modern telescopes focus the light received from celestial sources on a CCD to capture images of the sky and store them in digital format. These devices, due to the intrinsic properties of the used materials, perform at best in the wavelength range of 300-1100 nm (today the range also includes X-rays), therefore being exceptional tools for visible and near-infrared astronomy.

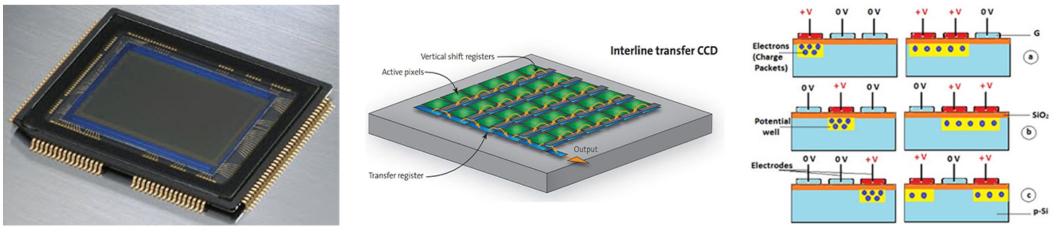

Figure 1.1. On the left: A picture of a CCD (Credit: https://www.elprocus.com); In the middle: A schematized CCD, showing the path followed by the captured electrons through the pixels array (Credit: https://www.stemmer-imaging.com); On the right: The schematized vertical section of a pixel moving the captured electrons employing potential wells (Credit: https://www.elprocus.com).

A CCD is an array of several *picture elements* (pixels) composed of metal electrodes (generally 3 per pixel), a p-silicon semiconductor layer, and a silicon dioxide layer. Pixels hit by photons produce photo-electrons within the silicon semiconductor, and holes of positive charge are formed proportionally, due to the photoelectric effect. The electrons are captured in a potential well, generated applying a voltage through the electrodes to the semiconductor and the silicon dioxide layer. These wells can store several electrons provided by their capacitance. This quantity is inversely proportional to the separation between the semiconductor and the electrodes while it is proportional to their surface.

The structure of the pixels in a CCD is called (p-doped) metal oxide semiconductor (MOS). The pixels dimensions, then, are essential to determine the capacitance of the entire detector. Larger pixels mean a larger number of electrons that can be stored before reaching saturation, but this also means a lower resolution of the image. Indeed considering a fixed area of the CCD, smaller MOS structures means a higher number of pixels (modern CCD can have 5000-7000x9000-10000 pixels 2-10 microns wide).

Once the electrons are captured, they are moved along the CCD columns, shifting the three electrodes voltage periodically. Once an entire row of captured electrons is shifted, the electronic charge is collected and is measured as a voltage. The voltage is then converted into a digital number. Generally speaking, the digital output number is in terms of digital counts or analog-to-digital units ADU. A minimum number of electrons need to be captured to produce an ADU. This factor is called *Gain* (a reasonable *Gain* value can be 10 electrons/ADU). The production of electrons is expressed in terms of *Quantum Efficiency* (QE). The ideal CCD has a QE=100%, meaning that for every striking photon, an electron is produced. Ideally, the electrons should have the same QE, but in the real case, it can slightly differ from an electron to another. Thus a technique called flat fielding is required to calibrate the pixels response. For what concerns the CCD noise, a mention of the readout noise and the dark current is needed. Due to the electronics, a CCD can be affected by readout noise, also known as read-noise.

Read noise can be produced in the conversion from analog-to-digital or by spurious electrons produced by the electronics. This kind of noise can contribute with an additive uncertainty in the value of each pixel. The other noise component is called Dark Current. It is the result of thermal electrons production. To lower this contribution to noise, the CCD needs to be cooled, working at low temperatures, using an appropriate cooling system. The signal-to-noise ratio for observations made with a CCD is provided then by the equation:

$$\frac{S}{N} = \frac{N_*}{\sqrt{N_* + n_{pix}(N_S + N_D + N_R^2)}} \tag{1.1}$$

Here, N_* is the number of photons collected (equivalent/proportional to the number of electrons produced), n_{pix} is the number of pixels, N_S is the total number of photons produced by the background per pixel, N_D is the total number of dark current electrons per pixel, and N_R is the total number of photons per pixel resulting from the read noise. Here all the sources of noise follow the Poisson distribution, except for N_R, which is shot noise. With more and more efficient CCDs, observing the sky at ideally low temperatures, the dominant source of noise is $N = N_S n_{pix} + N_*$. Since the number of received photons is typically large, the Poisson distribution is well approximated by a Gaussian distribution, with mean N and standard deviation $\sqrt{N}$.

If noise is dominated by N_* the equation 1.1 becomes:

$$\frac{S}{N} = \frac{N_*}{\sqrt{N_*}} = \sqrt{N_*} \tag{1.2}$$

As N_* is defined by $N \cdot t$, where N is the number of photons per second and t is the integration time, then $\frac{S}{N} \propto t$. It is therefore possible to solve $t(\frac{S}{N})$ to find the integration time needed to have a chosen $\frac{S}{N}$.

The current standard practice is to build 10-12 microns wide pixels to increase image resolution, with over 100 millions pixels, a read noise as low as 1 electron, and a QE close to 100% [112, 76].

1.3.2 Bias and Dark

To provide an estimate of the values produced by an unexposed CCD, a calibration of the *bias* level must be performed. An unexposed pixel can produce a small ADU distribution around zero. Bias (or zero) frames permits to measure the zero level of a CCD. The image has an exposure of zero seconds, achieved letting the shutter closed. The root mean square of the bias level is the read-noise. As well as for flat fielding, an average of 5-10 bias is normally recommended. Another estimate achieved using the closed shutter, but with an exposure time greater than zero, is the Dark current estimation. Dark current, as mentioned above, is a source of noise, generated by thermal electrons produced by the electronics at a fixed temperature. Using dark frames is possible to estimate the thermal noise for given exposure time, and it is also possible to avoid very long exposures, assuming the linear dependence with time (which is not always true). CCDs are generally cooled using liquid nitrogen (LN_2)

or liquid helium-4 (H_4) to reduce thermal contribution to practically zero [112, 76].

1.3.3 Flat fielding

CCDs are not perfect imaging devices; as pointed out above, each of the CCD pixels can have a slightly different *Gain* or QE value. To cope with this, the CCD images are typically calibrated using a *flat field* image. An ideal flat field image should homogeneously light up all the pixels, i.e. the light should be spectrally and spatially flat. Of course, this task is not always easy to accomplish, and an ideal flat field is challenging to produce. Indeed QE pixels variations are wavelength-dependent so that the illumination needs to have a very flat distribution in wavelengths, and naturally, the illumination needs to be uniform over all the device. A standard procedure is to light up the inside of the telescope dome, pointing the telescope at a bright spot of the dome, taking short exposures to avoid saturation. In the case of more filters, the flat fielding needs to be repeated for every filter, generally averaging a sample of five to ten flat field images. Another possible procedure consists in exposing the CCD to dusk/dawn sky, taking care of removing any star using a median filter. Flat fields obtained using the dome procedure are referred to as *dome flats*, while those obtained using the sky are called *sky flats*. For what concerns satellite telescopes, flat fielding is performed in the laboratory before launch, using de-focused observations of the Earth or the Moon or constant a dithered starfield [112, 76].

1.3.4 Background Subtraction

CCD pixels collect every photon produced from the objects and the sky within the telescope field of view. It is crucial then to disentangle the contribution that comes from the sky, from that coming from the objects of interest. The sky or background signal contains not only the actual photons of the sky but also the light produced by unresolved astronomical objects. The unwanted signal can be subtracted using specific software. A simple solution can be to center an annulus around an object of interest, with an area at least a few times larger than the source aperture. All the pixels, which value is below 3σ (a threshold which eliminates cosmic rays and possible contamination) in the annulus area are summed, and the sum is divided by the number of pixels, obtaining an estimated mean value of the background light around the object [76].

1.4 Digital Imaging: Processing

Important advantages of digital images are provided by image processing. A digital image is basically a multidimensional array, with array elements tracking the pixel values. Mathematical operations can therefore be easily performed. In this section, a few number of basic operations helpful to understand the following chapters are described.

1.4.1 Affine Transformations

Let us briefly and informally introduce the concept of affine space. An affine space is a geometric structure related to the Euclidean spaces, which preserves parallelism and ratio of lengths for line segments. In an affine space, there is no concept of distance. Therefore, all the Euclidean spaces are affine.

An affine transformation is a function f between affine spaces X, Y which preserves points, straight lines, and planes. Every affine transformation $f : X \to Y$ is of the form $x \to Mx + y$. M represents a linear transformation in the space X, x is a vector in X, and y is a vector in Y. So an affine transformation is composed of a translation and a linear transformation. When working with vectors, the linear transformation is represented by a matrix multiplication, whereas the translation is represented by a vector addition: $f(\vec{x}) = A\vec{x} + \vec{y}$, A is a matrix.

Applying affine transformations to digital images is quite straightforward. In this particular case, we talk about image transformations. The transformation relocates the pixels, interpolating the values of moving ones. A small sample of 3×3 transformation matrices is shown in Tab. 1.1 [176]:

Sample transformation	Description
$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$	**Identity**: Produces the same input image
$\begin{bmatrix}-1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$	**Reflection**: Reflects along the x-axis
$\begin{bmatrix}\alpha_x & 0 & 0\\0 & \alpha_y & 0\\0 & 0 & 1\end{bmatrix}$	**Scale**: Scales along the axis by α_x and α_y
$\begin{bmatrix}\cos(\theta) & -\sin(\theta) & 0\\\sin(\theta) & \cos(\theta) & 0\\0 & 0 & 1\end{bmatrix}$	**Rotate**: Rotates by an angle θ
$\begin{bmatrix}1 & \beta_x & 0\\\beta_y & 1 & 0\\0 & 0 & 1\end{bmatrix}$	**Shear**: Shears along the axis by β_x and β_y

Table 1.1. A sample of 3×3 affine transformation matrices

1.4.2 Filtering

Digital filtering is a technique used to blur, sharpen, or detect edges in a digital image. Filtering an image means that a convolution is performed. Convolution is a mathematical operation by means of which each pixel in the image is summed to its neighbors, weighted by some *kernel*, and producing a smaller size image called "feature map" [57]. Convolution is achieved sliding the kernel over the image with a defined stride, making a matrix multiplication and summing the results onto the feature map. Convolution can be expressed as follows:

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} * \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \dots & y_{mn} \end{bmatrix} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{(m-i)(n-j)} y_{(i+1)(1+j)}$$

Depending on the kernel, different results can be obtained, implying a vast range of interesting applications. As an example blurring can provide a slight noise reduction, edge detection can be used for segmentation (described in Sect. 1.4.3) or blurring plus edge detection can be coupled to reduce noise preserving the image details. A small list of kernels is shown here:

- $\begin{bmatrix} -1 & -1 & -1 \\ -1 & \alpha & -1 \\ -1 & -1 & -1 \end{bmatrix}$ Edge Detection: Enhances edges of a factor α
- $\begin{bmatrix} 0 & -1 & 0 \\ -1 & \beta & -1 \\ 0 & -1 & 0 \end{bmatrix}$ Sharpen: Sharpens of a factor β
- $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ Gaussian Blur 3×3: Blurs with a normalized Gaussian

This operation of convolution is performed directly in the spatial domain, but an alternative solution is the usage of a filter that works in the frequency domain, masking unwanted frequencies. For digital signals, this can be achieved using the Fourier Transform, while for digital images, the Wavelet Transform is used.

1.4.3 Image Segmentation

Image segmentation is a technique used to partition a digital image. It is used to locate boundaries along level curves, assigning a label to all those pixels which share similar properties, such as color or intensity. A large number of methods can be used to produce an image segmentation with different degrees of complexity.

A brief description of a small number of segmentation methods is shown here:

- **Thresholding**: Probably the simplest image segmentation method, it is based on setting a fixed threshold above which to the pixels is assigned the same integer index. Pixels below the threshold are instead set to zero and are not segmented

- **Clustering**: Many clustering algorithms can be applied to images producing a segmentation. These methods include *K-means*, DBSCAN, DENCLUE, and hierarchical clustering, which are described in Sect. 3.2.2

- **Compression**: Based on the assumption that the optimal segmentation is the one that minimizes the coding length of data [120]. The compression efficiency is related to the image regularity, while segmentation tries to find patterns

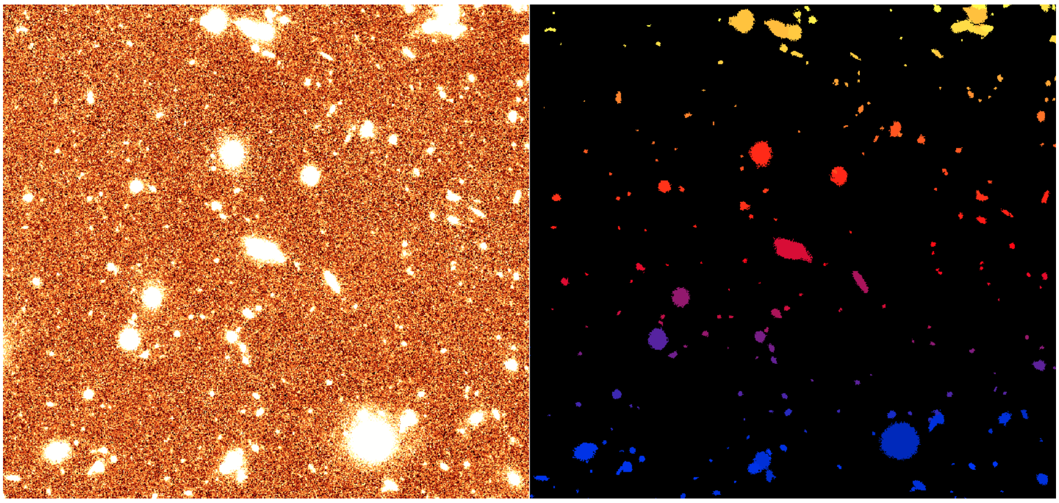

Figure 1.2. Segmentation of an HST image performed using a thresholding approach.

and similarities. The segmentation is produced using a bottom-up hierarchical clustering method. For each segment, the number of pixels that compose the texture and the boundary is used to build up a probability distribution function. The probability function is then used to select the best segmentation.

- **Histogram**: Histograms are computed using pixels in the image. Peaks and valleys are then used to locate clusters [153]. The approach is highly time-efficient, as the method requires to see pixels only once, even if it is not always easy to identify significant peaks and valleys. A refinement consists in applying the method recursively on clusters detected, to find smaller clusters in it [153] [140].

- **Region-growing**: Different kind of region-growing methods have been developed. These methods assume that neighboring pixels share similar values within a confined region.

  The seeded method requires a fixed number of seeds in input, which identify the object to be segmented. Neighboring pixels are compared, starting from the seeds. If the difference between the value is smaller than a similarity parameter, then those pixels are added to the segment.

  The unseeded method, instead, uses a region in place of seeds. In the same way as the seeded method, it compares the neighboring pixels, but in case those pixels are below a certain threshold, a new region is created, and they are attached to the new region.

  Another method called λ-connected is based on pixel intensities and linking paths between pixels. The method estimates a so-called λ connection to create segments [36]. For an overview of region-growing segmentation see [84]

- **Partial differential equation**: Based on partial differential equations (PDE), these methods are used for object extraction and tracking. Using curve propagation, which is ruled by PDEs, to find the minimum in the potential defined by a cost function. The PDE methods family includes parametric methods, level-set methods, and fast marching methods. Parametric (or Lagrangian) methods, configure contours according to sampling and evolution of each segment.

Level-set methods define a surface $s(x, y)$. The interception with the plane that represents the image, defines the contour $\Gamma = [(x, y) \mid s(x, y) = 0]$. The evolution of Γ is used to derive a similar flow for s [131].

Fast marching methods relies on two quantities: time t and speed v, defined by the evolution of a closed surface. Indeed each pixel x in the image Ω requires a certain $t(v)$ to be accepted in a segment. t is defined by the relation: $|\nabla t(x)| = 1/v$ for $x \in \Omega$. The pixels are considered far (if not yet considered for inclusion), considered (being considered for inclusion) and accepted (accepted permanently in the curve) [152].

- **Variational**: Variational methods are based on energy functional minimization to find the optimal segmentation $u*$. The functional is composed of a data-fitting term and a regularizing term. The Potts model is a variational method define by:

  $$argmin_u \gamma ||\nabla u||_0 + \int (u - f)^2 dx$$

  where f is the input image to be segmented, u is the candidate segmentation, γ is the parameter which controls trade-off between regularity and data fidelity, while $||\nabla u||_0$ is called jump penalty. $u*$ has an optimal trade-off between the squared L2 distance to the given image f ($\int (u - f)^2 dx$) and the total length of its jump set ($argmin_u \gamma ||\nabla u||_0$) [177].

  A generalization, called Mumford-Shah model, takes into account the total length of the segmentation curve K and the smoothness of the approximation of u [122].

- **Watershed**: Watershed sees images as topographic landscapes with ridges and valleys. The relative elevations are defined by values of the respective pixels or their gradient magnitude. The method decomposes an image into catchment basins (which behave like clusters). For each local minimum, a catchment basin comprises all pixels who flows downhill following the steepest descent, which ends at this minimum. The watershed then decomposes an image completely, and pixels drained to a common catchment basin form a segment [17].

- **Model-based**: Model-based techniques impose a prior which correspond to a specific model that is believed to best fit the objects in the image [158]. Searching the probabilistic model, which better represent the target objects is then the main issue, and this approach limits the segmentation only to images populated of objects which share similar features.

Image segmentation is of fundamental importance for astronomy and for all scientific fields which rely on imaging. This image processing technique is the first level of source extraction, the procedure that locates and identifies objects, such as stars and galaxies, from an astronomical image. In many cases, segmentation alone is not enough to properly identify astronomical objects. Other image processing techniques, such as Denoising and Deblending, described in Chapt. 4 and Chapt. 5, are used to enhance objects detection, extraction, and segmentation, also limiting unwanted spurious detections.

1.5 Astronomical images

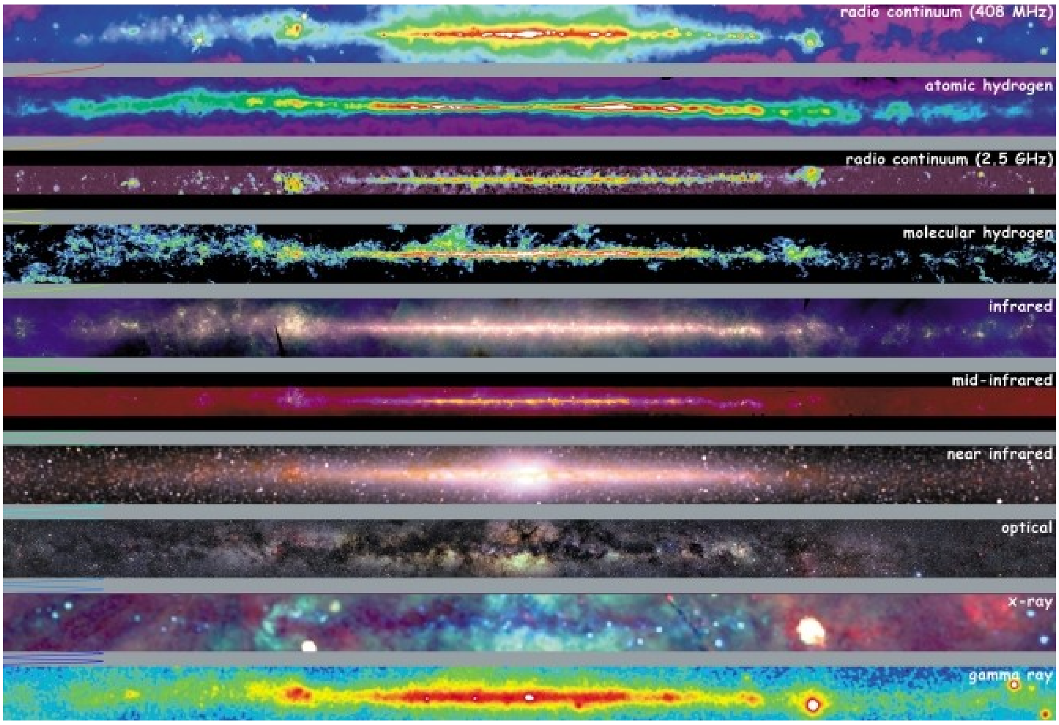

Figure 1.3. Image of the Milky Way in different wavelengths from radio to gamma-rays, showing different light profiles. Credit: http://mwmw.gsfc.nasa.gov/mmw_edu.html

The astronomical images that we obtain through CCDs, and then we analyze, are the result of what the telescopes observe. Telescopes technical features (e.g., spatial resolution, the field of view, aperture, etc.) define what kind of celestial objects are optimal to be observed. The response of the telescope observing a point source is defined by the so-called “Point Spread Function” (PSF). The PSF defines the minimum resolution that can be achieved with the related telescope. The spatial resolution is related to the diameter of the telescope and to the wavelength of the photons emitted by the observed objects. Therefore, images produced by various telescopes can be really different. The wavelength at which the observation is performed has a large impact. Astronomical objects emit light at different wavelengths in the electromagnetic spectrum, with peak emissions that depend on the underlying physical mechanisms. Some typical sources with different wavelength emission are:

- **Radio** - Supermassive Black Holes, Supernova remnants, Pulsars, Star forming regions, Quasars, Active Galactic Nuclei (AGN)

- **Microwave** - Neutral Hydrogen Clouds, AGN, Cosmic Microwave Background
- **Infrared** - Galaxies, Red-Brown Dwarfs, Cosmic Dust , Planets, AGN
- **Visible** - Main sequence Stars, Planets, Galaxies, AGN
- **Ultraviolet** - Blue-White Stars, AGN, Supernova remnants
- **X-rays** - AGN, Supernova remnants, Ionized Gas in Galaxies Clusters
- **Gamma-rays** - AGN, Gamma-ray Bursts

Our idea of the sky and of the objects that populate it completely changed with the advent of observations from space. The Earth's atmosphere is composed of molecules that absorb photons at specific wavelengths, making it opaque and impossible to observe them from the ground. Furthermore, the atmosphere limits the spatial resolution of the observation, causing an apparent blurring/distortion due to the variation of its refractive index. This blurring is referred to as "astronomical seeing".

When dealing with observations in the optical and near-infrared domains, the choice between space and ground-based telescopes is dictated by the scientific goals (high resolution, long exposure times, etc.). But for other wavelengths (Far-infrared, Microwave, X-ray, and Gamma), the presence of the atmosphere is an insuperable obstacle that can be overcome only by means of space observations (or high-altitude balloons). On the other hand, Radio telescopes have huge diameters needed for resolution purposes, which makes them (for the moment) not compatible with space missions.

Since space observations do not have to deal with the atmosphere blurring, they typically produce sharper images than ground-based, and at a wider variety of unabsorbed wavelengths. On the other hand, space telescopes are difficult/impossible to fix in case of technical problems, they are smaller, and expensive.

One last distinction is directly represented by the objects that are observed. Astronomy can be broadly divided into many scientific branches, each relying on different observables. For example, in cosmology, where the rules that govern the evolution of the Universe are investigated, the objects of interest range from microwave (with the Cosmic Microwave Background) to Weak Lensing (optical and near-infrared). Extragalactic Astronomy objects of interest range from galaxies in the optical near-infrared radiation to Ultraviolets, X-rays, and Gamma-rays, for AGNs and Quasars. In the next section, we will focus on the standard file format used for astronomical images and measurements regarding celestial objects, mainly considering the optical and infrared wavelengths.

1.5.1 Flexible Image Transport System

The standard file format within the astronomical community is the Flexible Image Transport System (FITS), developed in 1981 by Don Wells, Eric Greisen and Ron Harten [175]. This standard allows a flexible transport of data from the

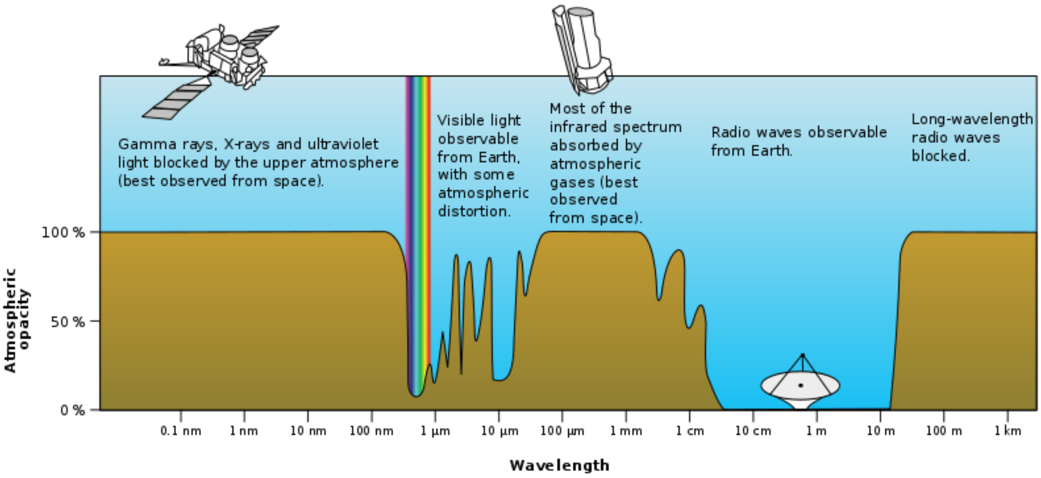

Figure 1.4. Earth's atmosphere absorption at different wavelengths. Credit: NASA, public domain via https://commons.wikimedia.org/wiki/File:Atmospheric_electromagnetic_opacity.svg

observatory to the researchers. A FITS standard image is composed by a "header" of at least 36 lines (filled with blank characters if left empty) of 80 bytes, for a minimum of 2,880 bytes. Each line contains a keyword for a maximum dimension of 8 bytes. To each keyword a value with useful information about the image is assigned. Keywords and values follow very strict rules about dimensions and names. Required keywords are listed below [175]:

- SIMPLE - has a boolean value T (true) or F (false), stating whether the image is FITS standard conform

- BITPIX - defines, with an integer value, the number of bits in the image. 8 (8-bit), 16 (16-bit) and 32 (32-bit) or -8, -16, -32, -64 for the floating-point 8-bit, 16-bit, etc.

- NAXIS - defines the data array shape. With 1 for 1-D data, such as a spectrum, with 2 for a 2-D image, such as those collected with a CCD. With 3 it defines a multi-layered cube of 2-D images. The maximum is NAXIS=999

- END - the last keyword which completes the header.

other optional keywords, which can be added above the mandatory END keyword are listed here [175]:

- BSCALE - used in combination with BZERO keyword, to retrieve the true physical values from the array pixels values, according to the equation:

$$\text{true\_value} = \text{BSCALE} \times \text{array\_value} + \text{BZERO}$$

- BZERO - used in combination with BSCALE keyword

- OBJECT - used to give a name for the main object observed

- TELESCOP - defines the telescope used to acquire the data
- INSTRUME - defines the instrument used to acquire the data
- OBSERVER - identifies who acquired the data.

A total of 53 different keywords can be used. The full list can be found at https://heasarc.gsfc.nasa.gov/docs/fcg/standard_dict.html. Following the header, the actual "data" begin: a set of arrays with dimensions and shape described by the NAXIS parameters. The FITS image is closed by the "tailer", an ASCII string of null characters.

The raw data in a FITS image is usually "reduced" to be corrected and calibrated. Following the steps described in Sect. 1.3, bias/dark subtraction and flat-fielding are performed. Different well-known tools allow FITS manipulation:

- AIPS - Astronomical Image Processing System is a free software package for radio interferometric data calibration/display/analysis designed by National Radio Astronomy Observatory (NRAO) in Charlottsville, Virginia (1978). The new version of the software is completely written in C++ and is called Common Astronomy Software Applications (CASA) (https://casa.nrao.edu/)
- IRAF - Image Reduction and Analysis Facility is a free general-purpose software for astronomical images reduction and analysis. It includes a complete programming environment and a programmable Command Language scripting facility. IRAF is supposed to outclass AIPS in terms of portability and device-independence. It is supported by National Optical Astronomy Observatories (NOAO) in Tucson, Arizona (http://ast.noao.edu/data/software)
- STSDAS - Space Telescope Science Data Analysis System is a free software layered on top of IRAF, specifically designed for HST data. The software is distributed by Space Telescope Science Institute (STScI), Baltimore, Maryland (http://www.stsci.edu/institute/software_hardware/stsdas)
- STARLINK is a free software developed in the UK in 1980, aiming to analyze astronomical images. The main objectives were to coordinate different data reduction tools through sharing and standardization, including hundreds of items and external software such as IRAF. The main support facility was closed in 2005, but support still exist at the UK Joint Astronomy Centre in Hawaii (http://starlink.eao.hawaii.edu/starlink)
- ESO-MIDAS - is a software built to integrate complex analysis with flexibility, including imaging and special reduction packages for European Southern Observatory (ESO) instrumentation at La Silla and the VLT at Paranal. Some of the STARLINK project ideas were used for the interface. It is command-line driven, and the monitor is a command interpreter. The MIDAS Command Language (MCL) is the language associated and allows to construct complex command strings. It is distributed by ESO, (https://www.eso.org/sci/software/esomidas//)

- IDL - Interactive Data Language is a commercial package, not specifically designed for astronomy, but with a more scientific purpose. Used also in scientific engineering and for medical data, it incorporates an array-oriented language as an alternative to FORTRAN and C. In astronomy IDL is used for data simulation and modeling. The software license can be purchased at https://www.harrisgeospatial.com/Software-Technology/IDL

- CFITSIO - is a library of C and Fortran subroutines for reading and writing astronomical data. It also provides many advanced features for manipulating and filtering the information in the FITS file. CFITSIO can be called from other programming languages using specific interfaces. Compatible programming languages are (C++, C#, Perl, Tcl, Python, Ruby, S-lang, MatLab, and LabVIEW) (https://heasarc.gsfc.nasa.gov/fitsio/)

- ASTROPY - is a free software package written in the Python programming language designed for astronomical analysis. The package contains classes for representing multidimensional arrays or tables, unit, and physical quantity conversions, astronomical constants, celestial coordinates. It includes the former standalone PyFITS interface to load FITS images. Astropy was written by the Astropy collaboration [38], following the increasing Python popularity among astronomers increased in the last decade (https://www.astropy.org/).

1.5.2 Astrometry

Astrometry is the precise determination of positions and motions of astronomical objects. Mapping the celestial objects provides precise information about their distribution and the local environment. This field is fundamental to understand Solar System, stellar, and galactic dynamics. With a sufficient angular resolution, it is also possible to directly observe binary systems, thus supporting the detection of exoplanets thanks to the relative motion between the planet and its host star. Astrometry is also used to estimate the distribution of DM in our galaxy and to put constraints on supernovae explosions. Defining a reference system for positions of celestial objects and measuring their distances, puts the basis of all astronomy.

A standard celestial reference system, the International Celestial Reference System (ICRS), is adopted by the International Astronomical Union (IAU). The origin at the barycenter of the Solar System. A variant of the ICRS is the International Celestial Reference Frame (ICRF), which relies on sources observed in the radio wavelength. The first version of ICRF, the ICRF1 succeeded the previous “Fifth Fundamental Catalog” (FK5), outperforming it by one order of magnitude in terms of angular noise floor and reference axis stability [79]. ICRF1 agrees with the orientation of FK5 “J2000.0” frame to within the precision of the latter, where J2000.0 is January 1, 2000, 11:58:55.816 UTC (Coordinated Universal Time) [79]. From 1998 with ICRF1 to 2009 with ICRF1 and lastly ICRF3 in 2019, the ICRF passed several revisions, incorporating an increasing number of extragalactic sources coupled with new features, such as the effect of galactocentric acceleration of the solar system [33] [127].

1.5.3 Photometry

With the term *photometry* we refer to the process of measuring the flux emitted by an astrophysical target. The term monochromatic flux F indicates the power received (W $m^{-2}Hz^{-1}$) integrating the specific intensity over the angular size of the source. A measure relative to F is the magnitude m. If we define F_0 as a reference monochromatic flux we obtain the relation between m and F:

$$m = m_0 - 2.5logF + 2.5logF_0 \tag{1.3}$$

Taking $m_0 = 0$, as the magnitude of a reference star, then the term $2.5logF_0$ defines the magnitude scale zero-point. Because the flux (and then the magnitude) is wavelength dependent, and astronomical objects are not monochromatic, two well-known magnitude systems are defined: namely, the Vega system and the AB system. The Vega system takes as reference the flux of the star Vega, setting the magnitude of the star to 0 in every bandpass. The AB (absolute) system, based on the work by Bev Oke [129], is not based on Vega, and indeed no relative reference object is used; instead, it assumes that F_0 is the same for all the wavelengths and passbands. The relation between F in frequency units and m is:

$$m_\nu = -2.5logF - 48.6 \tag{1.4}$$

while F in wavelength units and m is:

$$m_\lambda = -2.5logF - 21.1 \tag{1.5}$$

Colors

Color indices (or simply *colors*) are defined as the difference of magnitude of an object at two separate wavelengths. Taking the two filters, defined by a fixed wavelength range: $\Delta\lambda_1$ and $\Delta\lambda_2$, the magnitude of the objects measured in the different filters are $m_{\Delta\lambda_1}$ and $m_{\Delta\lambda_2}$, so the color index is defined as:

$$color\ index = m_{\Delta\lambda_1} - m_{\Delta\lambda_2}. \tag{1.6}$$

Taking, for example, a set of filters, such those in the “UBV system”[81] (U=365 nm, B=440 nm, V = 550 nm), the differences in magnitude (U-B=$m_U - m_B$, U-V=$m_U - m_V$, B-V=$m_B - m_V$) define the color index in this photometric system.

Aperture photometry

Aperture photometry is the technique of measuring the fluxes of astronomical objects fluxes summing the pixel values within a given “aperture”, i.e. a circular or elliptical area centered on the object centroid. It can be shown that for a point-like source (e.g. a star) an aperture $A = \pi r^2$ with radius $r = 3\times$ FWHM (where FWHM is the Full Width at Half Maximum of the Point Spread Function) contains $\sim 100\%$ of the object flux. The measured flux should be background subtracted to estimate the correct value of the object flux. A relation between the aperture radius and the signal-to-noise ratio (S/N) indicates that the peak of the cumulative S/N is reached with a radius equal to $r = 3 \times FWHM$. Larger apertures include a larger than required number of pixels increasing the contribution from noise sources to the error term and lowering the S/N [76].

Profile fitting

An alternative method to aperture photometry relies on modeling the object observed with analytical functions, which are fitted to the real data until a good match is achieved. These profiles are generally Gaussian for stars and point-like objects:

$$I(r) = I(0)e^{-r^2/2\sigma^2} \tag{1.7}$$

where *I(0)* is the peak intensity, r is the radial distance and σ is the Gaussian standard deviation. The real PSF can be used for fitting, and this is the preferred choice when possible. In this case, we talk about "PSF fitting".

For extended sources, the stardard choice to model the brightness as a function of the radial distance is the Sérsic profile [151]:

$$I(r) = I(r_e)exp\left\{-b_n\left[\left(\frac{r}{r_e}\right)^{1/n} - 1\right]\right\} \tag{1.8}$$

where r_e is the half-light radius, n is the Sersic index, and b_n is a constant proportional to n. n controls the profile curvature: with small n the profile is more diffuse, while larger values provide a steeper logarithmic slope, concentrating most of the brightness in a limited region. For example, setting $n = 1$ provides an exponential profile $I(r) \propto e^{br}$ which is generally used as a good spiral galaxy disk approximation. On the other hand, setting $n = 4$ gives the $I(r) \propto e^{br^{1/4}}$ which is used to model elliptical galaxies. In crowded fields, profile fitting is useful, as it weights more the pixels close to the peak, while weighting less the possibly spurious/contaminated pixels at the edges of the considered radius. Of course, profile fitting relies on the good quality of the fit, which sometimes is hard to achieve (e.g., faint, irregular or blended objects).

Template photometry

Template photometry is an advanced technique used to recover the correct photometry from objects in a low-resolution image (LRI), using information from a set of *priors* obtained from a high-resolution image (HRI). The priors are degraded to match the resolution of LRI by means of a convolution kernel and used to solve a linear minimization problem. This technique is implemented by several codes [46, 95, 20] and [115, 114], it is particularly useful when observational data in different wavelengths are combined. Even space-based and ground-based observations can be combined, using the space-based images, that plays the role of HRI, to obtain the priors for the ground-based, which in this case is the LRI. Template photometry can be used to bypass the objects blending issue in the LRI, provided that the objects are enough separated in the HRI [115, 114].

Photometric Redshift

Photometric redshift, or Photo-z, is a measure of the redshift z using multi-band photometry. This technique relies on the observation of strong enough features of the object, which must therefore be observed in different bands. The photo-z technique was initially developed in 1962 by W. A. Braum [15]. Braum observed six bright elliptical galaxies belonging to Virgo cluster using a photoelectric photometer in nine band-passes from 3730Å to 9875Å. He estimated the six galaxies average spectral energy distribution (SED) comparing it with that of 3 galaxies belonging to Abell 0801. He was able to measure the displacement of the two energy distributions resulting in a value of $z = 0.19$ compared to the known spectroscopic value of $z = 0.192$. Even if this measure was accurate, it relies on 4000Å break spectral feature, making it reliable only for evolved, e.g. early-type galaxies. Years later, other works improved the technique: using photographic plates and a theoretical no-evolution model for the galaxies [92], defining a galaxy colors-redfshift diagram of "permitted" redshift range [137], using the "ultraviolet dropout" technique, relying on the Lyman break in galaxy spectra [159][107].

Whenever possible, more accurate spectroscopic redshifts are preferable to photo-z. However, in the 1990s-2000, the technique gained popularity thanks to increasing size of surveys, in which a growing number of faint high-redshift objects have been observed, making a spectroscopic analysis unfeasible. To handle this large amount of available data, in the recent years Bayesian methods and Artificial Neural Networks have been deployed to estimate the photometric redshift. Even if these measures are the only alternative to analyze a large amount of data, their reliability is often less robust than that of spectroscopy.

1.6 Image formation model

This work aims at discussing two particular issues related to astronomical digital images: namely, *denoising* and *deblending*. To understand the algorithms used to tackle these tasks, we need to introduce a mathematical model of image formation. The model takes into account the processes involved when an image observed with a telescope is discretized on a pixels grid.

First of all, let us consider a continuous function such that the intensity of the two-dimensional image is defined as $F(x, y)$. As already discussed in Sect. 1.5, the atmosphere produces a distortion effect in the image, which can be expressed with the convolution of the Point Spread Function, $PSF(x, y)$:

$$\hat{F}(x, y) = F(x, y) * PSF(x, y) \tag{1.9}$$

Considering a discrete grid of pixels, such that (i, j) are the indices, and representing the generic pixel p_{ij}, $\hat{F}(x, y)$ is sampled at regular intervals $(\delta x, \delta y)$ along the axis of the grid. Added to the image is the noise component $\eta(i, j)$, which is a combination of different sources such as those discussed in Sect. 1.3.1. Then the final image is represented by:

$$G(i, j) = \hat{F}(i\delta x, j\delta y) + \eta(i, j) \tag{1.10}$$

where $G(i, j)$ represents the original image $F(x, y)$ on a pixels grid. All the methods discussed in Chapt. 3 assume this model and perform mathematical operations on the discrete domain defined here.

This model assumes the presence of a noise component, which naturally depends on the kind of image we are taking into consideration. Denoising methods aim at reducing this component as much as possible, to recover the correct contribution of $\hat{F}(x, y)$ to each pixel $p_{i,j}$ of the grid.

On the other hand, the PSF convolution, blurring the image, lowers the resolution, producing in some cases the blending of nearby objects. To reduce this effect, a narrow PSF is required, which can be obtained in areas of optimal astronomical seeing (e.g., dry areas), using adaptive optics, or observing from high-altitude balloons or space. Unfortunately, most of the time, the blending issue is only partially removed. Deblending algorithms aim at reducing this issue, recognizing blended objects and dividing pixels among them, or even recovering the correct shape of blended objects (through profile reconstruction techniques).

A free astronomical software, SExtractor, that handles both denoising and deblending and is widely used in the astronomical community, is discussed in the next section.

1.7 Source detection and deblending with SExtractor

In the past two decades, source detection, deblending and aperture photometry in (extra-galactic) astronomy have been typically performed by means of one of the most used free software available to the scientific community, SExtractor [19]. This software, although developed more than 20 years ago, is still widely used by astronomers. For example, it has been recently used for detection, deblending and photometry in the CANDELS survey [66, 90, 69], and it will be used for the detection in the forthcoming Euclid pipeline [97], although other solutions have been considered. It is a powerful, well tested and versatile code, and it provides a solid baseline for new software comparison. Although being suited for many applications including photometry and background subtraction, detection, deblending, in this section we focus only on detection, denoising (filtering), and deblending tasks. The information discussed here will be useful for Chapt. 4-5.

First of all, the software analyzes the set of images provided by the user. If known, the background noise model can be provided through an *rms* map, where each pixel stores the root mean square (*rms*) value of the background pixels distribution. Alternatively, the background noise model is estimated locally in each mesh of a grid that covers the whole image. The local background histogram is clipped iteratively until convergence (set at $\pm 3\sigma$) around its *median*. If σ value doesn't change more than 20% during the process, the field is considered not crowded and the *mean* of the clipped histogram is taken as a value for the background. Otherwise, the *mode* is estimated as follows: [19]:

$$mode = 2.5 * median - 1.5 * mean \tag{1.11}$$

After that, the background is subtracted. Once the subtraction is performed, the software filters the image (when a filter is provided, e.g., PSF or a Gaussian kernel), and then the image is thresholded on-the-fly. The threshold is essentially used for detection. Subsequently, the detected objects are deblended and cleaned (see Cleaning below), morphological information, and photometry are measured, and in the end, the resulting information is written in an output catalog (see SExtractor documentation at: `https://www.astromatic.net/software/sextractor`, and [75]).

The above-cited passages are analyzed below.

Filtering

With SExtractor it is possible to pre-filter the image, before detection. We have seen in Sect. 1.4.2 the possible advantages provided by filtering, depending on the kernel. Filtering can be enabled in the configuration file by setting the boolean value `TRUE` to the parameter `FILTER`, while the kernel can be chosen providing to the parameter `FILTER_NAME` the *string* name and path of the kernel to convolve with the image. In general, filtering the image enhances the quality of detection; an in-depth analysis of filtering and other denoising methods is shown in Chapt. 4.

Detection

SExtractor uses threshold parameters `DETECT_THRESH` and `DETECT_MINAREA` to determine if a generic pixel is part of an object. There are three criteria that tune the detection of objects :

- All the pixels forming the object must have values above `DETECT_THRESH` × the corresponding value of the *rms* map pixel
- All the pixels forming the object must be adjacent (including diagonals)
- The total number of pixels forming the object must be larger than (or equal to) `DETECT_MINAREA`.

In some cases it is convenient to set `DETECT_THRESH` as an absolute value (expressed in ADUs) instead of the value relative to the *rms*; this is possible setting `THRESH_TYPE` to `ABSOLUTE` instead of `RELATIVE`.

The larger is the values for `DETECT_THRESH`, and `DETECT_MINAREA`, the more significant must be the objects to be detected, performing a conservative detection. Conservative combinations restrict the probability of spurious detections at the cost of not detecting faint objects. The application of denoising algorithms allows setting less conservative combinations of these two parameters, increasing the number of faint objects detections since the spurious ones are less likely.

Deblending

Once the objects are detected, the deblending step is performed. In SExtractor, deblending separates overlapping objects using a multi-threshold approach. The software does not know a priori which of the detected objects are instead two or more objects blended. Indeed it runs the deblending algorithm to each detected object (which in some cases can produce false positives). To introduce the method, we take as an example a generic detected object. The parameter `DEBLEND_NTHRESH` defines the number of thresholds: the pixels belonging to the detected object are assigned to exponentially spaced bins between DETECT_THRESHOLD and the peak value. After that, a “tree” is built, following the flux profile of the blended objects. SExtractor divides the object into branches every time pixels above a threshold are separated by pixels below it. Two criteria define a branch:

- The number of counts in the branch is above a fraction, defined by DEBLEND_MINCONT × the total number of counts of the object
- There exists at least another branch above the same threshold and above the fraction.

DEBLEND_MINCONT represents the minimum threshold required for a branch to be separated from the three. Typical values range from 0.01 to 0.0001. Ideal values for DEBLEND_NTHRESH range from 8 to 64.

An in-depth analysis of deblending algorithms and their application is given in Chapt. 5.

Cleaning

Once detection and deblending are concluded, the clean step, which can be activated by setting to TRUE the parameter CLEAN, checks all the detections again, to see if they would have been detected without their neighboring objects around. An estimate of the neighbors’ contribution is performed using a function which approximates the PSF, the Moffat profile [121]:

$$I(r) = \frac{I(0)}{(1 + kr^2)^\beta} \tag{1.12}$$

β can be changed through CLEAN_PARAM, and ideal values range from 0.1 to 10, while 1.0 is the default value.

Outputs

SExtractor can produce an output catalog containing up to ∼400 columns (the number of parameters can be set through the .param file), storing information about all the extracted objects, such as their position in the image, their magnitudes/fluxes at different apertures, their ellipticity, etc. A small list of commonly used parameters is shown below:

- FLUX_ISO is the isophotal flux of a detected object, obtained by summing all its background-subtracted pixels within the detection footprint. MAG_ISO is the associated magnitude
- FLUX_APER is the fixed-aperture fluxes array, i.e. the fluxes within a given set circular aperture, summing all the pixels above the background, within any chosen radius. MAG_APER is the associated magnitude
- FLUX_AUTO is the automatic aperture flux estimating the total flux of a detected object, summing all the pixels within an adaptive elliptical aperture; the shape follows from [93]. MAG_AUTO is the associated magnitude

- X_IMAGE (Y_IMAGE) is the x (y) coordinate of the isophotal image centroid
- ALPHA_J2000 (DELTA_J2000) is the J2000 right ascension (declination) of the isophotal image centroid
- FWHM_IMAGE (FWHM_WORLD) is the FWHM estimate, assuming a gaussian core, expressed in pixels (deg)
- ELLIPTICITY is the ellipticity of the detected object defined as: $1 - \frac{\texttt{B\_IMAGE}}{\texttt{A\_IMAGE}}$, where A_IMAGE (B_IMAGE) is the isophotal major (minor) axis

The software can also be configured to produce the objects segmentation, their apertures, the background map, etc., setting the CHECKIMAGE_TYPE keyword value to match the desired check image output.

From the topics discussed in this chapter, it should be clear that noise and blurring (one of the leading causes of objects blending) affecting astronomical images are a crucial issue. Recovering such lost information is fundamental for imaging surveys that aim at high precision measures. Before discussing denoising and deblending algorithms, some of the next-generation surveys and their scientific goals are described in the next chapter.

Chapter 2

Next-generation surveys

2.1 Big data: imaging with next-generation telescopes

With the term *Big Data*, we refer to that field enclosing data processing software specialized in mining information out of massive datasets, that traditional methods cannot handle. Nowadays from social networks to e-commerce, from sports to science, enormous quantities of data are analyzed to find trends and patterns. Astronomy is no exception. Future advanced facilities will produce unparalleled massive data, making the upcoming datasets extremely large, far too much to be analyzed with the currently available tools and techniques. Forthcoming surveys will easily produce dozens of petabytes (PB) of data to be analyzed fastly and precisely. New approaches are needed. The term Big Data is in use since 1990s [103], and different definitions exist[96, 25, 73]. We can summarize, for astronomy, four crucial keywords:

- **Volume**: Simply the volume of data expressed in Terabytes (TB), Petabytes (PB) or Exabytes (EB)
- **Variety**: Meaning complexity, in terms of a large number of attributes per data point, different data types/formats (obtained with different telescopes)
- **Velocity**: The rate of data being produced by the telescopes and analyzed by software pipelines
- **Value**: The importance of data produced to discover new trends and patterns.

New strategies and algorithms are constantly being developed to handle this upcoming challenge (data mining processes). These new methods aim at analyzing large amounts of data to find correlations, trend, patterns, anomalies, outliers, and so on. A particularly interesting data mining field is Machine Learning (ML). A more detailed description of ML algorithms is given in Chapt. 3.2.

One of the recent surveys of the sky, and one among the first to collect enormous amount of data, is the Sloan Digital Sky Survey (SDSS) [67]. The SDSS project began in 2000 and is still producing data. SDSS-IV is the latest phase, started in 2014, and ending in 2020. By now, after more than 15 years, the SDSS collaboration has catalogued $\sim 10^8$ stars, $\sim 10^6$ galaxies, and $\sim 10^5$ quasars)[4]. SDSS is extremely important for statistical studies. A popular side-project, based on crowdsourcing

Survey	Data Volume
2MASS	3 TB [183]
SDSS	40 TB [183]
DES	∼6 PB [1]
WFIRST	∼30 PB expected [94]
Euclid	∼30 PB expected [40]
PanSTARRS	∼40 PB expected [183]
LSST	∼60 PB expected [3]
SKA	∼1 EB/day expected [14]

Table 2.1. Future and already concluded Sky Surveys with their data volumes

on SDSS images, is Galaxy Zoo [141]. Galaxy Zoo asks people to visually classify objects extracted from SDSS images, by simply reaching the site (http://zoo1.galaxyzoo.org/) and choosing among different morphological options. The project evolved in time, providing more and more options for a more detailed classification. The latest version (https://zoo4.galaxyzoo.org) combines new SDSS imaging with the HST survey CANDELS [90][66].

Next-generation surveys will produce data a factor $\sim 10^3$ larger than the data volume produced by SDSS, due to increased resolution, depth, and/or observation area. This huge quantity of raw data needs to be reduced, cleaned and analyzed, by optimized algorithms to provide high level information. A sample of surveys and related data volumes is shown in Tab. 2.1

Observables and objects for cosmological and extragalactic studies are discussed in the next section.

2.2 Cosmological and extragalactic observables

In this section, we will discuss the objects and phenomenons that will be observed and analyzed by the next-generation sky surveys to shed light on the history and evolution of the Universe. In particular, we will focus on the observables that can be extracted from extragalactic optical and infrared images: they (and their analysis) are the principal focus of this thesis.

We are interested in two categories of observables, astronomical objects, and astrophysical phenomena. Astrophysical phenomena of cosmological interest retain information about the evolution of the Universe and its composition, and are Weak Lensing (WL), Baryonic Acoustic Oscillations (BAO). Astronomical objects of cosmological and extragalactic interest are source of information for dark energy, structure evolution, reionization, star formation, etc., and are Galaxy Clusters, Supernovae Ia (SNe Ia), Galaxies, Quasars, and Active Galactic Nuclei (AGNs).

The gravitational potential field, generated by billions of galaxies and dark matter over-densities, affects the path of the photons from the source where they are emitted to the Earth. This phenomenon is called Gravitational Lensing. Gravitational lensing behaves like a piece of glass with multiple refraction indexes, that vary with position. It distorts light with variable intensity. We refer to Strong Lensing (SL) and **Weak**

Lensing (WL), to classify distortions effects. SL creates visible distortions such as rings, arcs, and multiple images. On the other hand, WL creates weak distortions of background sources, that can only be detected by analyzing statistical methods on a large number of objects to find coherent distortions.

The resulting image shows a slight distortion, and is commonly referred as *cosmic shear* (or simply *shear*). Measuring *shear* and its amplitude yields to the actual map of the gravitational field, which encodes the statistical properties of the large scale structure of the Universe. The pattern of alignments in background galaxies can be studied measuring galaxies ellipticities, but since the distortion is very faint, accurate measurements with high-resolution images are required [97].

Another way to study large scale structures is the analysis of **Galaxy Clusters**. Quantitative measures of the properties of clustering can be performed with the *correlation function*, to extract a distribution of galaxies with respect to their mutual distance [136]. Along with WL, clusters of galaxies quantify the large scale structures of the Universe. Furthermore being the largest virialized over-densities that we observe, clusters probe the growth of structures. Clusters follow the statistical distribution of the initial fluctuations and those with high masses at high redshift are extremely sensitive to possible primordial non-Gaussianity and deviation from the standard dark energy model [97].

An interesting outcome related to galaxy clusters correlation function is a preferred comoving distance of $\sim 150Mpc$. This distance provides a standard ruler which probes the expansion history of the Universe. This distance is the signature of pressure waves in photo-baryonic primordial plasma, measured in the CMB and are called **Baryonic Acoustic Oscillations** (BAO). Another way to measure distances is by using standard candles such as **Supernovae Ia** (SNe). SNe are used to establish the redshift-distance relation, which led to the discovery of dark energy [143]. Further detections can be used to measure the expansion history of the Universe while measuring their frequency allows to determine the star formation history of the Universe and the stellar evolution.

A number of issues are still under investigation [97], for example:

- The nature of dark matter and its relationship with baryonic matter
- The internal structure of dark matter halos or how the central black hole impacts its host galaxy evolution
- Where and when does star formation take place and is quenched
- The galaxy scaling relations and their dependency on the environment
- The luminosity function bright end at $z > 6.5$ and galaxy formation model
- The contribute of star-forming galaxies, quasars, and AGNs to reionization.

Future surveys will need to investigate objects at different redshift from local regions to the primordial Universe, to try and answer these questions. Collecting information in different wavelength from optical to infrared will be crucial.

In the next sections we will introduce some of the most important upcoming surveys for extra-galactic astronomy: LSST, Euclid, WFIRST, and DES (which has already concluded the period of observations). The first three represent an exhaustive sample of next-generation optical/infrared surveys. For what concerns DES, even if already concluded, its data are still not publicly available, it shares similar scientific goals with the other three, and it is supposed to be one of the external surveys (EXT) supporting the Euclid satellite observations.

2.3 Euclid

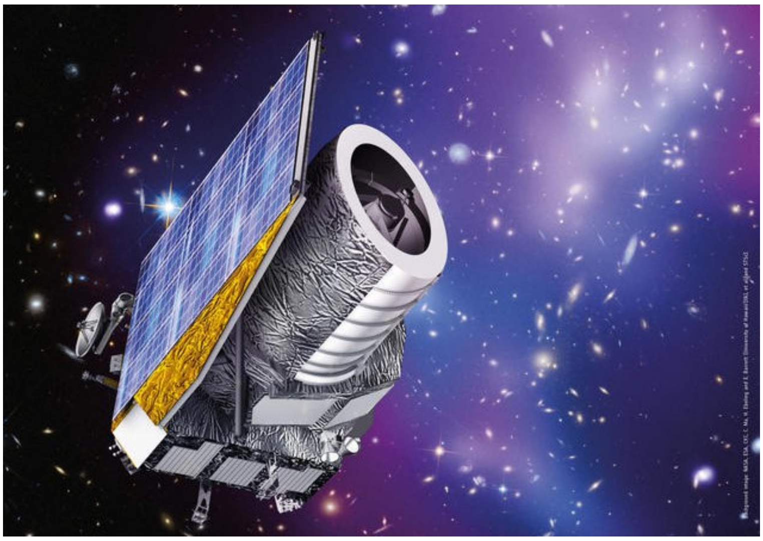
Background image: NASA, ESA, CXC, C. Ma, H. Ebeling and E. Barrett (University of Hawaii/IfA) et al. and STScI

Figure 2.1. An artist view of the Euclid Satellite. Credit: https://www.euclid-ec.org/

Euclid is a Medium Class ESA mission, which will be deployed in 2022 [2]. The main goal of the Euclid survey is to show how Dark Energy (DE) acceleration modifies the distribution of Dark Matter (DM) in the Universe. Euclid will measure over a billion of galaxy shapes and photometric redshifts, to study Weak Gravitational Lensing and Galaxies Clustering. The survey will investigate the nature of DE and DM, using so-called "cosmological probes". The Euclid mission is optimized to accurately measure two observational signatures on the geometry of the Universe: the Weak Lensing (WL) and the Baryonic Acoustic Oscillations (BAO). With WL, it is then possible to map DM, which is essential to measure the DE by quantifying the apparent distortions of the galaxies. With BAO, it is possible to derive a standard rule to measure the expansion of the Universe. Along with these two main probes, Euclid will provide several other essential data on galaxy clusters, redshift space distortions, and the integrated Sachs Wolfe effect. To achieve the level of precision needed for the WL analysis, high-quality images are required. Scales below the arcsec in the visible and infrared bands are essential to estimate galaxy shapes and photometric redshifts of the lensed galaxies out to $z \geq 2$. For what concerns BAO, accurate spectroscopic near-infrared measures of galaxies redshift out to $z \geq 0.7$ are required. Both WL and BAO require a massively large number of detected sources, resulting in a survey which will observe a major fraction of the extragalactic sky. The Euclid Wide Survey will scan 15,000 deg^2 (i.e. 36% of the whole sky). Euclid uses the WL and BAO to measure the terms of the DE equation of state, the validity of General Relativity, constraints on the sum of the neutrinos masses, deviations to a Gaussian distribution of initial perturbation, with absolute precision, making the mission able to challenge the cosmological model on many levels. As above mentioned, Euclid will produce a vast dataset of billions of galaxies and millions of spectra out to $z \geq 2$ together with morphologies, masses and star-formation rates with accuracy four times better and three near-infrared magnitudes deeper, than possible from the ground. At low redshift, it will resolve the stellar population of galaxies within $\sim$ 5 Mpc, while in combination with eROSITA, Planck and SZ telescopes it will derive the mass function of galaxy clusters, discovering over 10^5 strong lensing systems. The gravitational lensing will also be used together with the near-infrared photometry to further investigate the relationship between light, baryons, and DM between galaxies. In addition to the Wide Survey, a Deep Survey will observe an area of 40 deg^2 (composed of two 20 deg^2 regions), reaching a sensitivity two magnitude deeper than the wider. The Deep Survey will be the primary target for follow-up observations, containing objects at $z \geq 6$, and quasar candidates at $z \geq 8$. The mission lifetime is of 7 years, containing 1 month to reach the Lagrangian Point L2, 3-6 months of necessary calibrations, 6 years to complete the Wide+Deep Survey, mapping the sky in the step and stare mode and sending data to ground stations (at least one granted at time) at a rate of 850 Gbit/4 hours [97].

Instrumentation

Euclid is equipped with a 1.2m diameter Korsch reflecting telescope, appropriate for large field observations. The telescope reflects light to two instruments, the visual instrument (VIS) and the near-infrared instrument (NISP). VIS is equipped with 36 CCDs. It can observe objects with a very high resolution (better than 0.2 arcsec PSF FWHM, comparable to HST), with a pixel scale of 0.1 arcsec/pixel, in one wide band (R+I+Z) from 550 to 900 nm. The NISP instrument operates in 3 different bands (Y=920-1146 nm, J=1146-1372 nm, H=1372-2000 nm), using 16 mercury cadmium telluride (HgCdTe) near-infrared detectors with a pixel scale of 0.3 arcsec/pixel and $\sim$ 0.4 arcsec resolution. VIS and NISP operate in parallel; VIS only performs photometry measurements, while NISP also performs spectroscopy measurements. In the Wide Survey VIS will reach a sensibility of 24.5 magnitudes at 10σ for extended sources, while NISP will reach 24 magnitudes at 5σ for point sources for photometry and 3×10^{-16} $erg\ cm^{-2}\ s^{-1}$ at 3.5σ unresolved line flux for spectroscopy [97].

2.4 Large Synoptic Survey Telescope

The Large Synoptic Survey Telescope (LSST) is a 6.7 meters reflecting telescope with a field of view of 9.6 deg^2, which is in construction at the EL Peñón Peak of Cerro Pachón in Chilean Andes. LSST will carry out a 20,000 deg^2 wide survey in 10 years. The multi-purpose survey is designed to explore the Solar System beyond the Kuiper Belt, to further investigate the structure of the Milky Way together with the observation of the Local Group, and to observe billions of distant galaxies,

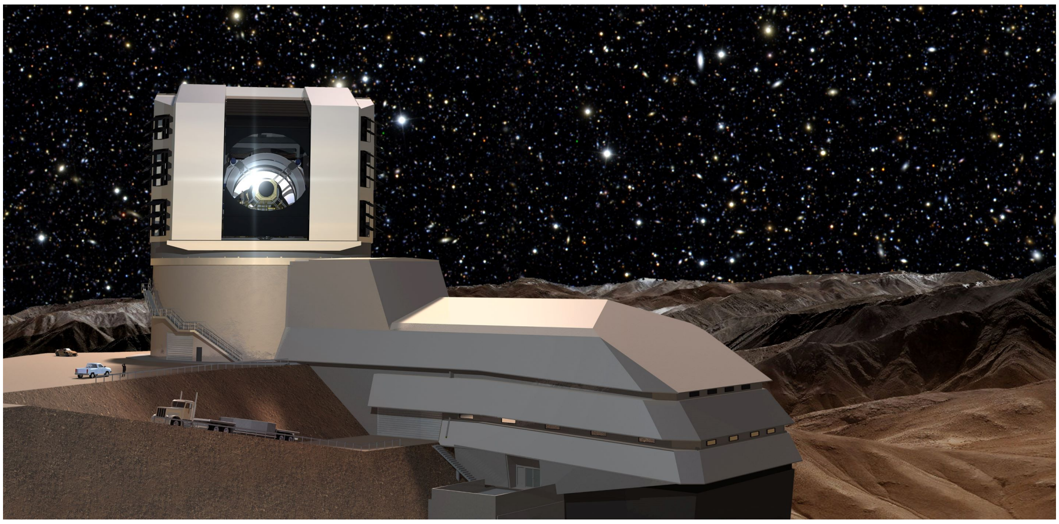

Figure 2.2. LSST facility representation at EL Peñón Peak of Cerro Pachón, Chile. Credit: https://gallery.lsst.org/

enhancing our knowledge of cosmology and galaxy evolution. The survey builds on small patches of the sky, each one to be visited 1000 times. A “visit” consists of two 15-seconds exposure per filter. The outcoming processed amount of data (32 bit) is expected to be of 30 TB/night, providing information of 20 billion objects in 10 years of activity. The wide survey will take the 90% of the observational time, leaving the remaining 10% for very deep observations (with typical single visit depth ~ 26 mag) focusing on the ecliptic, the Galactic plane and the Small and Large Magellanic Clouds. Another 1% will be used for micro-surveys, for a total of 25 nights over ten years.

LSST will observe the small bodies of the Solar System as well as the Kuiper belt region, which contains a portion of the early planet population. The encoded information will be fundamental to understanding the formation and evolution of our planetary system. Observations of the Milky Way will shed light on structures, chemical compositions, and kinematics of the stars. With LSST we will be able to understand the accretion history of the Milky Way and the fundamental properties of all the stars within 300 pc from us. The long lifetime of the wide survey will be of fundamental importance to discover and study transients and variable phenomena. With a time resolution of 15 seconds, repeated observations of the same patches of the sky and an ever-increasing limiting magnitude, the telescope will also enable the analysis of neutron stars, black holes binaries, active galactic nuclei (AGN) variability and the optical counterparts of gamma-ray burst and X-ray flashes. With the wide survey, it will be possible to better understand the evolution of galaxies, as well as galaxies morphologies with respect to their environment, providing an unprecedentedly large statistical sample. LSST will also be crucial for better constraining the cosmological model. Being a deep, wide-field, multi-color survey, with a single dataset it will provide with excellent precision four cosmological probes: WL, BAO, mass function evolution, redshift measurements using type Ia supernovae [105].

Instrumentation

The telescope is a modified Paul-Baker three-mirrors system (named M1, M2, and M3) with three refractive lenses (L1, L2 and L3) and a color filter at the focal plane, before the sensor. 189 CCDs compose the LSST camera with $10\mu m$ pixels. The pixel scale is 0.2 arcsec/pixel, which will work in the bands (u, g, r, i, z, y), covering the wavelength from 320 to 1080 nm. The median free-air seeing is 0.7 arcsec/pixel in the r band [105].

2.5 Wide-Field InfraRed Survey Telescope

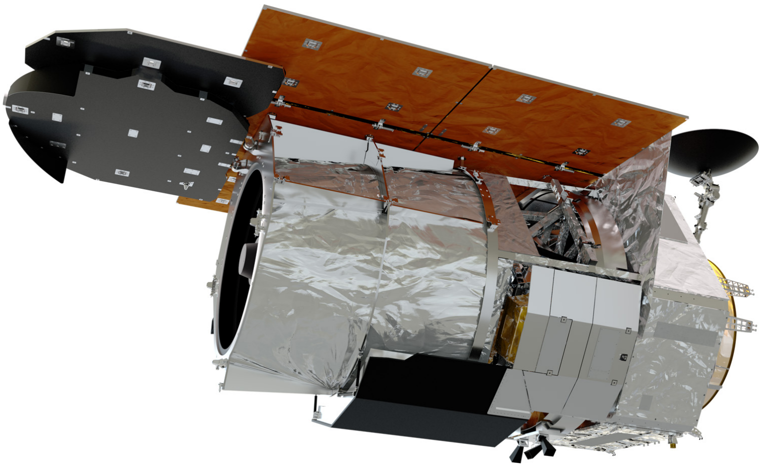

Figure 2.3. Rendered model of the WFIRST spacecraft. Credit: https://wfirst.gsfc.nasa.gov/

The Wide-Field InfraRed Survey Telescope (WFIRST) is a 2.4m telescope equipped with a Wide Field Instrument (WFI) and a Coronagraphic Instrument (CGI). WFI is composed of a wide-field camera and an Integral Field Unit (IFU), while CGI is composed by a High Contrast Imaging Testbed (HCIT) and an Integral Field Spectrograph (IFS). WIF-IFU will focus on distant supernovae Ia (SN) to study the evolution of the Universe, while CGI-IFS will be able to characterize the super-Earth and Neptune-like planets atmospheres. To WFIRST was added the acronym AFTA (Astrophysics Focused Telescope Assets) when the National Reconnaissance Office offered to donate two telescopes, with the same size of Hubble Space Telescope (HST), with shorter focal length but a wider field of view. The telescope is now known as WFIRST-AFTA. The telescope has the spatial resolution of HST-WFC3 camera but with a field of view ~ 200 times larger. Observing millions of galaxies, WFIRST-AFTA will build an accurate 3D distribution of DM across $\sim 2200\ deg^2$ of the sky. This map coupled with and the high-resolution images produced by the telescope, will be used to shed light on galaxy formation and evolution. Through the High-Latitude Imaging Survey (HLS Imaging) it will enable accurate WL shape measurements of $\sim 10^8$ galaxies, yielding precise analysis of distances, clustering, galaxy-galaxy lensing and mass profiles of galaxy clusters. With the High-Latitude Imaging Survey (HLS Spectroscopy), instead, it will measure grism spectroscopic redshift of $\sim 10^7$ galaxies. With the SN Survey, a new measure of the expansion of the Universe will be provided, thanks to the analysis of more than 2700 supernovae out to $z = 1.7$. WFIRST will also provide "Degree-Deep-Fields", ~ 100 times larger than HST deep fields. It will determine positions/motions of more than 200 million stars in the Milky Way with astonishing precision. This can be achieved collecting photons produced by sources 500 times fainter than the ones observed with Gaia telescope, providing then interesting data to trace DM distribution in our Galaxy. It will be used to observe the infrared counterpart of gravitational waves merger events, thus providing a deeper and more resolved view of the transient Universe. It will grant interesting data regarding stellar evolution and galaxy formation of nearby galaxies. Thanks to its features, WFIRST-AFTA is the optimal candidate to complement missions like Euclid and LSST.

The larger mirror of WFIRST compared to the Euclid one balances the longer wavelengths of near-infrared photons compared to the optical photons ones, thus producing images with a similar angular resolution of the Euclid-VIS image, in the near-infrared band. In the primary overlapping areas between the surveys, the higher resolution near-infrared images (by WFIRST) can be coupled with the VIS images (by Euclid), granting images of the same objects in different wavelengths at the same resolution. Moreover, even if WFIRST survey will be narrower than Euclid survey, it will be ~ 2 mag deeper.

Instrumentation

WFIRST telescope mounts a 2.4m primary mirror, associated with a secondary and a tertiary mirror, providing a three-mirror anastigmatic configuration. The orbit is geosynchronous, with a 28.5-degree inclination, motivated by the high data rate achievable. The telescope operates at a temperature of 270K, with the near-infrared focal plane at 118K. WFI is composed of 18 H4RG-10 HgCDTe detectors, distributed in a 6×3 array. The pixel scale is 0.11 arcsec (slightly variable in the different bands), with 300Mpixels covering an area of 0.281 deg^2. The camera has a filter wheel, spacing from wavelength 0.76 μm to 2.0 μm and a grism, which grants an $R = 700$ resolution. IFU has a three arcsec field of view and an $R = 75$ resolution. With the HLS survey, several depths (at 5σ) in the different bands are reached (Y = 26.7, J = 26.9, H = 26.7, F184 = 26.2) in a lifespan of 1.3 years for imaging. For spectroscopy (at 7σ) a depth of $0.5 \times 10 - 16 erg/s/cm^2 @ 1.65 \mu m$ is reached, in 0.6 years. The SN Survey, divided in 3 "micro-surveys" will reach different depths in relation to the surveys areas (Wide = 27.44 deg^2 $Y = 27.1, J = 27.5$; Medium = 8.96 deg^2 $J = 27.6, H = 28.1$; Deep=5.04 deg^2 $J = 29.3, H = 29.4$). CGI operates in the 400-1000 nm band, having a 2.5 arcsec field of view, a 10^{-9} effective contrast and a 100-200 mas inner working angle. While IFS has a $R = 70$ resolution [156].

2.6 Dark Energy Survey

The Dark Energy Survey (DES) investigates night sky with a wide-field camera mounted on the 4m Blanco Telescope at Cerro Tololo Inter-American Observatory, in Chile. The five years survey observed 5000 deg^2 of the sky in five different filters, four optical and one near-infrared (g, r, i, z, Y). DES is a collaboration between United

Figure 2.4. Blanco Telescope Dome at Cerro Tololo Inter-American Observatory, Chile. Credit: https://www.darkenergysurvey.org

States, Brazil, United Kingdom, Germany, Spain, and Switzerland. DES officially saw the light in August 2013 and completed the last session of observation in January 2019. Data will be released to the public after one year of the proprietary period. The main goal of DES is to determine, with high precision, the ratio of pressure to density (w) in the DE equation of state, and other key cosmological parameters. In addition DES is expected to catalog $\sim\ 3 \times 10^8$ galaxies with photometric redshifts, of which $\sim\ 3 \times 10^8$ will have shape measurements for WL. Part of the survey area (250 deg^2) overlaps those of the South Pole Telescope (SPT), which complements the optical techniques used by DES in finding galaxy clusters. The survey also overlaps with the Sloan Digital Sky Survey (SDSS)[67] and Vista Hemisphere Survey, to obtain a larger number of information on the galaxies observed by DES. A small area of 27 deg^2 will be used to measure supernovae Ia in the time domain [43].

Instrumentation

The wide-field camera used for DES is called Dark Energy Camera (DECam). DECam is a 570Megapixel camera, composed in an array of 62 2048×4096 back-illuminated CCDs. The CCDs are composed of high resistivity silicon with 15×15 μm pixels. The camera has a pixel scale of $\sim$ 0.26", a field of view of 3 deg^2 and an increased sensitivity, extending the wavelength range to 1050 nm. The typical single exposure is 90 seconds for *griz* and 45 seconds for Y. The survey reaches different depths related to the bands and the time exposure (g = 24.45, r = 24.3, i = 23.5, z = 22.9 and Y = 21.7; at 10σ). The estimated median seeing FWHM is $\sim$ 0.9" in filters *riz* [43]

2.7 Surveys synergies

Next-generation extragalactic surveys will produce unprecedented amounts of data. Although each will primarily focus on a few particular scientific goals, some of these goals are in common and the different approaches used to achieve them will constitute a unique opportunity: the combination of data from these surveys will produce far more complete and accurate results, surveying an extensive area of the extragalactic sky, with high resolution, a rapid cadence (from ground and space) and spanning multiple wavelength bands from optical to near-infrared.

Following [80] and [142], we will now summarize the possible synergies among LSST, WFIRST and Euclid for cosmological and extragalactic science.

As shown in the previous sections, LSST, WFIRST, and Euclid projects will provide different surveys in terms of observed sky area, wavelength, depth, and resolution. Their synergies rely on the combination of a large number of colors. Precisely, nine filters (six optical from LSST and three infrared from WFIRST/Euclid and one wide optical filter from Euclid) in the overlapping areas. WFIRST and Euclid images will be much sharper, as the PSF will be several times smaller.

Data from cosmological and extragalactic observables, discussed in Sect. 2.2, will be collected and handled with different techniques, each one having pros and cons, including systematic errors. The combination of these surveys provides the critical advantages of mitigating these errors. To name two, systematics on photo-z measures are the result of incomplete wavelength coverage, whereas low-resolution and blending create shear systematics.

Among the benefits granted it is pointed out: robust measurements of photometric redshifts, to calibrate LSST WL; analysis of the correlation between optical colors and near-infrared spectral energy distributions (SED) and a morphological study of the galaxies with high-resolution images [156].

According to [142], two possible levels of combinations can be performed: *catalog-level* and *pixel level*. The first assumes that the objects will be detected independently by the surveys and then matched, whereas the second assumes that the objects will be detected simultaneously. Various benefit about the two approaches can be considered. For example, the *catalog-level* LSST will provide confirmation of Euclid and WFIRST detection at their detection limits. While the *pixel-level* will enhance the deblenders performances, providing wavelengths and morphology information.

We now discuss the benefits from possible synergies, concerning the different observables.

Weak Lensing

The two space missions Euclid and WFIRST, with their optimal angular resolution, can provide improvements in the shear calibration for LSST. On the other hand, the ground-based telescope LSST will have a deeper limiting magnitude, collecting a large number of information on the outer parts of galaxy images.

An issue that affects ground-based telescopes such as LSST is the objects blending, discussed in Chapt. 5. Blending will affect nearly half of the observed galaxies [44][80], to reduce its impact a prior coming from Euclid (better than WFIRST, due to larger overlapping area) could be provided to the LSST deblender. On the contrary, Euclid could benefit from LSST observations at different optical bands. Indeed PSF is wavelength dependent; thus, it differs in relation to the SED. For this reason, the only wide optical band of Euclid does not permit to estimate the proper PSF for each galaxy/star, generating systematics for WL measurements.

Systematics of the three surveys are expected to be very different, cross-correlation of the shear estimates should erase the errors, leaving only WL signatures. Furthermore, the different redshift distribution of the lensed galaxies will contribute to building a complete dataset. Combination of spectroscopic and lensing data will provide a test of general relativity on cosmological scales. In LSST lensing sources span a redshift range between 0 and 3. For what concerns Euclid and WFIRST, their spectroscopic sources will span redshift from 0.4 to 3.5. With Euclid, a larger area will be covered, while with WFIRST a much higher density of sources, in a smaller area, will be provided. Furthermore, spectroscopic measures by Euclid and WFIRST can enhance LSST shear measurements. Indeed, along with CMB polarization, they are essential to calibrate its multiplicative shear bias.

Galaxy Clusters

LSST, Euclid, and WFIRST will observe different aspects of the galaxy clusters emission. LSST will provide optical colors, which will be used to measure photo-z and lensed objects time delays. Euclid and WFIRST will provide NIR photometry and spectroscopy with different areas, depths, and resolution of the images. Altogether these data/catalogs/images could provide remarkable improvements in the combined analysis. LSST is exceptionally efficient in finding clusters applying appropriate techniques for isolating overdensities (e.g., [148]), as they are often optimized for optical wavelengths. By using NIR images, cluster finding can be extended over $z \sim 1$ with Euclid and WFIRST. A joint analysis could improve a cluster sample selection, permitting to determine redshift distribution of background galaxies, that is a possible source of systematic errors. For what concern strong and weak lensing in galaxy clusters, WFIRST and Euclid can provide high-resolution images to calibrate LSST shear, while this last will observe faint arcs, produced by strong lensing, reaching a surface brightness of 28.7 mag per arcsecond.

Supernovae

WFIRST will observe thousands of SNe Ia per year from $z = 0.1$ to $z = 1.7$, while LSST will observe 10^4 SNe per year from $z = 0.03$ to $z = 1$. The combined analysis will measure distances from $z = 0.03$ to $z = 1.7$, measuring the dark energy contribute to the evolution of the universe. Combining wavelength coverage and flux calibration provided by WFIRST with the large aperture and the large number of SNe observed by LSST could lead to a better characterization of each supernova at different redshifts.

Photo-z

LSST will use photo-z to investigate Dark Energy, whereas WFIRST and Euclid rely on photo-z for mass power spectrum, faint galaxies weak lensing and for breaking spectroscopic degeneracies. Then reducing systematic error related to photo-z measurements is essential. The incomplete knowledge of the SEDs and poor spectral coverage, due to low resolution of photometry, leads to systematics. Even if LSST provides a significant advantage in estimating photo-z from $z = 0$ to $z \sim 6$, the photometric measures will be only in the optical wavelength. Combining these results with spectroscopic measures from WFIRST (which shares a comparable depth with the ten years survey), will provide complementary information, needed to reduce systematics. In the same way, Euclid can contribute with the NIR photometry in its three bands, using the large overlap of the surveys areas.

Galaxies

Combination of the three surveys will provide nine bands (from $0.3 \mu m$ to $2\ \mu m$) photometry at high signal-to-noise ratio. These combined data will allow easier identification of galaxies and quasars at high redshift. The wavelength coverage will provide SEDs of galaxies beyond $0.4\ \mu m$ to $z > 3$, allowing accurate determination of star formation rates and stellar masses.

Chapter 3

Advanced Image Processing and Computer Vision Techniques

In this chapter we discuss advanced techniques developed to extract information from images. Image processing and computer vision are interdisciplinary fields, spanning several areas of interest and performing an extensive number of tasks. Typical tasks include Image restoration, Object detection, Classification, Pattern recognition, Noise reduction, etc.

In Sect. 3.1, we focus on noise reduction techniques, aiming to provide an overview of different state-of-the-art algorithms, successfully applied in many fields. These algorithms are applied to extragalactic images in Chapt. 4 and their performances are analyzed and compared with noise reduction techniques standardly used in astronomy.

In Sect. 3.2, we focus on Machine Learning techniques and the possible advantages provided when applied on astronomical images. The application of some of these methods is shown for objects deblending in Chapt. 5.

3.1 Denoising Techniques

The focus of this section is to provide information on a large range of different image denoising families proposed in the literature. These techniques are then applied on astronomical images in Chapt. 4. An efficient denoising method is crucial to extract the information contained in the image and could be used as a preliminary step for other image processing problems, like the image segmentation and/or deblurring.

The main families of noise reduction techniques analyzed are:

- Gaussian smoothing (3.1.1)
- Anisotropic diffusion (Non-linear filtering) (3.1.2)
- Bilateral filter (3.1.3)
- Total-variation (TV) denoising (3.1.4)
- Image decomposition (3.1.5)

- Wavelets (3.1.6)
- Non-local means (3.1.7)

3.1.1 Gaussian smoothing

Let us consider the intensity function $I(x, y)$ of a noisy image, with $(x, y) \in \Omega$, where $\Omega \subset \mathbb{R}^2$ is the reconstruction domain. Let I_{clean} be the desired "clean image", i.e. an ideal, noiseless image with infinite signal-to-noise ratio. An image with a Gaussian noise component is:

$$I(x, y) = I_{clean}(x, y) + \eta \tag{3.1}$$

where $\eta \sim N(\mu, \sigma)$ is the additive noise component.
Of course, we want to reconstruct I_{clean} from I.

This filter uses a Gaussian function for calculating the transformation to apply to each pixel in the image. Mathematically, applying a Gaussian filter to a signal corresponds to convolving the signal with a Gaussian function. Since the Fourier transform of a Gaussian is still a Gaussian, applying a Gaussian smoothing has the effect of reducing the image's high-frequency components; a Gaussian filter is then *a low-pass filter*. In two dimensions, it is the product of two Gaussian functions, one in each dimension, so that the low-pass Gaussian filter is

$$G_\sigma(x, y) := \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.2}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.
As mentioned above, filtering the image $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ with a "low-pass" Gaussian filter is mathematically equivalent to process it with the heat equation [60, 101], that is solving the following linear partial differential equation

$$\begin{cases} \dfrac{\partial I}{\partial t}(x, y, t) = \nabla I(x, y, t) & \forall (x, y, t) \in \Omega \times (0, T_C]\,, \\ \dfrac{\partial I}{\partial \eta}(x, y, t) = 0\,, & \forall (x, y, t) \in \partial\Omega \times (0, T_C]\,, \\ I(x, y, 0) = I_0(x, y)\,, & \forall (x, y) \in \overline{\Omega}\,, \end{cases} \tag{3.3}$$

which has a diffusive effect on the initial datum I_0, for a small fixed time $T_C > 0$.
The relation between the Gaussian filter (3.2) and the problem (3.3) is that the solution of the heat equation is a convolution with the Gaussian filter, i.e.

$$I(x, y, t) = (G_\sigma(x, y) * I_0)(x, y) \tag{3.4}$$

with $\sigma = \sqrt{2t}$.
It is well known that applying that filter does not preserve edges. This edge blurring is due to the isotropic diffusion.

3.1.2 Anisotropic Diffusion

An improvement of the simple Gaussian filter is obtained by modifying the heat equation. Following the Perona-Malik model [139], we choose large values of $|\nabla I|$ as an indicator of the edge points of the image, in order to stop the diffusion at these points. In this way we move from isotropic to anisotropic diffusion, as follows:

$$\frac{\partial I}{\partial t} = div(\nabla I) \Rightarrow \frac{\partial I}{\partial t} = div(g(|\nabla I|)\nabla I). \tag{3.5}$$

The equation (3.5) must be complemented with suitable boundary conditions (e.g. homogeneous Neumann boundary conditions) and an initial condition. Perona and Malik pioneered the idea of anisotropic diffusion and proposed two functions for the diffusion coefficient (also called *edge-stopping functions*):

$$g_1(|\nabla I|) \quad := \frac{1}{1+\left(\frac{|\nabla I|}{K}\right)^2} \tag{3.6}$$

$$g_2(|\nabla I|) \quad := \exp\Big(-\left(\tfrac{|\nabla I|}{K}\right)^2\Big) \tag{3.7}$$

where K is the gradient magnitude threshold parameter that decides the amount of diffusion to take place.
We also consider other three edge-stopping functions that have been proposed after the original work by Perona and Malik:
In [22] is proposed an edge stopping function called Tukey's biweight function defined as:

$$g_3(|\nabla I|) := \begin{cases} \frac{1}{2}\Big[1-\left(\frac{|\nabla I|}{K\sqrt{2}}\right)^2\Big]^2 & \text{if } |\nabla I| \leq K\sqrt{2} \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

In [68] is proposed the following function:

$$g_4(|\nabla I|) := \frac{1}{1+\left(\frac{|\nabla I|}{K}\right)^{\alpha(|\nabla I|)}} \tag{3.9}$$

where

$$\alpha(|\nabla I|) := 2 - \frac{2}{1+\left(\frac{|\nabla I|}{K}\right)^2}. \tag{3.10}$$

And finally in [174] is proposed:

$$g_5(|\nabla I|) := \begin{cases} 1-\exp(-3.31488 * K^8/(|\nabla I|)^8) & \text{if } |\nabla I| \neq 0 \\ 1 & \text{otherwise.} \end{cases} \tag{3.11}$$

3.1.3 Bilateral filter

The Bilateral filter is an edge-preserving denoising algorithm that was first introduced by [161].

It is defined as (see also [12])

$$I(x) = \frac{1}{w} \sum_{x_i \in \Omega} I_0(x_i) f_r(\|I_0(x_i) - I_0(x)\|) g_s(\|x_i - x\|), \tag{3.12}$$

where

$$w := \sum_{x_i \in \Omega} f_r(\|I_0(x_i) - I_0(x)\|) g_s(\|x_i - x\|) \tag{3.13}$$

and

- I is the filtered image
- I_0 is the original input image to be filtered
- x are the coordinates of the current pixel to be filtered
- Ω is the window centered in x, so $x_i \in \Omega$ is another pixel
- f_r is the range kernel for smoothing differences in intensities (this function can be a Gaussian function)
- g_s is the spatial (or domain) kernel for smoothing differences in coordinates (this function can be a Gaussian function).

It averages pixels based on their spatial closeness and on their radiometric similarity. Spatial closeness is measured by the Gaussian function of the Euclidean distance between two pixels and a certain standard deviation ($\sigma_{spatial}$). Radiometric similarity is measured by the Gaussian function of the Euclidean distance between two color values and a certain standard deviation (σ_{color}).

3.1.4 Total Variation denoising

Total-variation denoising (also known as total-variation regularization) is based on the principle that images with excessive and possibly spurious detail have high TV, defined as

$$TV(u, \Omega) := \int_\Omega |\nabla u(x)| dx \tag{3.14}$$

for a function $u \in C^1(\Omega)$ (note that a similar definition can be given also for L^1 functions [91]). According to this principle, TV denoising tries to find an image with less TV under the constraint of being similar to the input image, which is controlled by the regularization parameter, i.e. tries to minimize $TV(I, \Omega)$. This minimization problem leads to the Euler-Lagrangian equation, which can be solved via the following evolutive problem:

$$u_t = \frac{\partial}{\partial x}\Big(\frac{u_x}{\sqrt{u_x^2 + u_y^2}}\Big) + \frac{\partial}{\partial y}\Big(\frac{u_y}{\sqrt{u_x^2 + u_y^2}}\Big) - \lambda(u - u_0), \tag{3.15}$$

for $t > 0$ and $x, y \in \Omega$, with homogeneous Neumann boundary condition and a given initial condition. TV denoising tends to produce “cartoon-like” images, that is, piecewise-constant images. The concept was pioneered by Rudin, Osher, and Fatemi in [147] and is today known as the ROF model. TV denoising is remarkably effective at simultaneously preserving edges whilst smoothing away noise in flat regions, even at low signal-to-noise ratios.

3.1.5 Structure-texture image decomposition

A general approach to the denoising problem is based on the assumption that an image I can be regarded as composed of a structural part u (i.e. the objects in the image), and a textural part v which corresponds to finest details plus the noise. Following the approach described in [9], such image decomposition technique is based on the minimization of a functional with two terms, one based on the total variation and a second one on a different norm adapted to the texture component. Given an image I defined in a set Ω, and let $BV(\Omega)$ be the space of functions with limited total variation in Ω we can decompose I into its two components by minimizing:

$$\inf \Big(\int_{\Omega} |\nabla u(x)| + \lambda \, ||v(x)||_X^p \, dx \Big) \tag{3.16}$$

where $|| \cdot ||_X^p$ denotes the norm of a given space X and the minimum is found among all functions $(u, v) \in BV(\Omega) \times X$ such that $u + v = I$. The parameter p is a natural exponent, and λ is the so-called splitting parameter which modifies the relative weights. The best decomposition is found at the λ for which the correlation between u and v reaches a minimum.

3.1.6 Wavelets

The wavelets transform is the counterpart for images of the Fourier transform and the wavelets domain, which is a sparse representation of the image that can be thought of similarly to the frequency domain of the Fourier transform [166]. Being a sparse representation means that most values are zero or near-zero and truly random noise is represented by many small values in the wavelet domain. Setting all values below some threshold to 0 reduces the noise in the image, but larger thresholds also decrease the detail present in the image.
Let us recall the relation introduced in Sect. 3.1.1

$$I = I_{clean} + \eta, \tag{3.17}$$

where η is the noise and I_{clean} is the clean image (signal). The components of η are independent and identically distributed (iid) as $\mathcal{N}(0, \sigma^2)$ and independent of I_{clean}. The goal is again to remove the noise obtaining an approximation $\widehat{I}$ of I_{clean} minimizing the mean square error (MSE)

$$MSE(\widehat{I}) := \frac{1}{N} \sum_{j=1}^{N} (\widehat{I}_j - I_j)^2, \tag{3.18}$$

where N is the number of pixels. Let us denote by $Y = \mathcal{W} I$ the matrix of wavelet coefficients of the image I where $\mathcal{W}$ is the orthogonal wavelet transform operator, similarly $F = \mathcal{W}I_{clean}$ and $E = \mathcal{W}\eta$ (see [171], [108] for more details on $\mathcal{W}$). The wavelet transform is based on the subbands (called details) at different scales usually indexed by $k \in \mathcal{K}, \mathcal{K} \subset \mathbb{N}$. The wavelet-thresholding method filters each coefficient Y_j from the detail subbands $k \in \mathcal{K}$ with a threshold function to obtain $\widehat{X}$. The denoised approximation is $\widehat{I} = \mathcal{W}^{-1}\widehat{X}$, where $\mathcal{W}^{-1}$ is the inverse wavelet transform. Two thresholding techniques are frequently used. The *soft-threshold* function

$$\varphi_T(x) := sgn(x) \max(|x| - T, 0) \tag{3.19}$$

which shrinks the argument x to 0 by the threshold T. The *hard-threshold* function

$$\psi_T(x) := x\, \mathbf{1}_{\{|x|>T\}} \tag{3.20}$$

which sets the input to 0 if is below (or equal) the threshold T. Note that the threshold procedure removes noise by thresholding *only* the wavelet coefficients of the corresponding subbands, while keeping the low resolution coefficients unaltered.

3.1.7 Non-local means

The non-local means algorithm averages the value of a given pixel with values of other pixels in a limited proximity, under the condition that the patches centered on the other pixels are similar enough to the patch centered on the pixel of interest. This algorithm is defined by the formula [27]

$$NL[I_0](x) = \frac{1}{C(x)} \int_{\Omega} \exp\left(-g^h_\sigma(x)\right) I_0(y)\, dy \tag{3.21}$$

where

$$g^h_\sigma(x) := \frac{G_\sigma * |I_0(x + .) - I_0(y + .)|^2)(0)}{h^2}, \tag{3.22}$$

I_0 is the original image, $x \in \Omega$, $C(x)$ is a normalizing constant, G_σ is a Gaussian kernel with σ denoting the standard deviation, and h acts as a filtering parameter. The algorithm has been found to have excellent performances when used to denoise images with specific textures[1].

We define by $size_I$ the image size in pixels, by $size_p$ the size of the patch in pixels, by d_p the maximal distance in pixels where to search patches, by n the image number of dimensions ($n = 2, 3$ depends if we consider 2D or 3D images). In its original version the computational complexity of the algorithm is proportional to: $size_I * (size_p * d_p)^n$ [27]. A new “fast” version is now preferentially used since its actual complexity is proportional to: $size_I * d_p^n$ [42].

Compared to the classic algorithm, in the fast mode the distances are computed in a coarser way, indeed all pixels of a patch contribute to the distance to another patch with the same weight, no matter their distance to the center of the patch. This approach can result in a slightly poorer denoising performance.

When the standard deviation σ is given, the method gives a more robust compu-

[1] https://scikit-image.org/docs/dev/api/skimage.restoration.html#skimage.restoration.denoise_nl_means

tation of patch weights. A moderate improvement to denoising performance can be obtained subtracting the known noise variance from the computed patch distances, that improves the estimates of patch similarity [28].

3.2 Machine Learning and Neural Networks

Machine Learning (ML) is a subset of artificial intelligence field, a science that researches methods for machines to acquire new knowledge, relying on patterns and inference. The goal is to make machines capable of solving tasks nowadays handled by human beings. The term ML appears for the first time in literature in a paper of Arthur Samuel in 1959 [149]. The precise definition of machine learning is:

Def: *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E* [118].

ML algorithms are deployed to perform a large variety of tasks: data mining, image segmentation, spam detection, market analysis, robotics, etc. Three are the factors that made ML so popular today [65]:

- Larger than ever amount of available data easily sharable via Internet
- Computing power: in terms of CPU performances, storage capability, and available memory
- Improved algorithms to be deployed on large datasets

Some example possible application for ML algorithms, include:

- Classification: The algorithm is capable of assigning data from a dataset to a certain category, having learned which are the main features (attributes) of the objects populating that category through a training set.
- Clustering: The algorithm is capable of recognizing patterns in the dataset, making it able to distinguish between different categories (named clusters) of objects without a training set
- Prediction: The algorithm is capable of making previsions using historical data as a training set to forecast specific events.

Generally speaking, a large number of features are available for each object in a dataset, and not all of them are useful to a specific task. Indeed a large number of unuseful attributes provide a noise component, which worsens the performances. For this reason, a dimension reduction is often performed. A popular solution is the Principal Component Analysis (PCA) [134]. PCA converts possibly correlated features in the dataset into a set linearly independent attributes, called principal components.

ML algorithms can be divided into 4 categories: supervised Learning (SL), unsupervised learning (UL) semi-supervised and reinforcement learning (RL) [65]. In the following sections, we will focus on supervised and unsupervised learning as of main interest for this thesis field of research.

3.2.1 Supervised Learning

Starting from a training set, which is a dataset containing labeled data, the SL algorithms learn the main features of the dataset building up an optimal model defined by a *cost function.* Datasets are data structures containing m number of rows. Each row has n columns, containing the n features. The training set must be representative such that the parameter space covered by the input features must span all possible objects for which the SL algorithm is to be used. SL solves two kinds of problems, classification, and regression. For example, a trained algorithm can distinguish between dogs and cats if in the training set, where it learned, there are images of both, dogs and cats, with the correct label. A possible real-life application is for driver-less cars, as the algorithm recognizes obstacles like pedestrians, other cars, etc. and also traffic lights road lines, etc. While if provided as training set the stock market data, including not only stock prices but also an event history of the prices, it will be able to predict price fluctuations.

How an SL algorithm learns from data is called regression. There are two kind regressions:

- Linear regression: when the algorithm learns to predict the value of a continuous variable
- Logistic regression: when instead learns to predict if the outcome will be achieved

Let's define y the continuous variable we want to predict independently by regression type. $x_1, x_2, .., x_n$ are the "features" (x_i), or input variables, characterizing the dataset. $w_1, w_2, .., w_n$ are the multiplicative coefficients that the algorithm assigns to x_i, they are called "weights" (w_i), and the additive constant called "bias" (w_0). y is defined by a linear combination of x_i:

$$y = w_0 + w_1 x_1 + w_2 x_2 + .. + w_n x_n = w_0 + \sum_{i=1}^{n} w_i x_i \tag{3.23}$$

Eq. 3.23 is called *hypothesis function.* The aim is to find the optimal set of w_i, which permits to best characterize the contribution of the different features. From now on, the predicted outcome will be y_p.

Cost Function minimization

Usually, weights are set to have a random initial value, which is close to zero. A trial and error approach is deployed to set weights to their optimal value. A so-called *cost function* takes into account the difference between y_p^j (predicted value) and y^j (target value reported in the label). Some of the weights that improve the error for a given row could deteriorate the error for another row. The average error across all the rows is considered, to avoid the issue. All the errors are, first of all, squared and then are summed, to avoid positive and negative errors to eliminate each other. The *cost function* assumes the form:

$$J(w_i) = \frac{1}{2m} \sum_{j=1}^{m} (y_p^j - y^j)^2 \tag{3.24}$$

Since $J(w_i)$ is quadratic, the shape of the curve ha only one minimum and further is from the minimum higher is the absolute value of the slope. Then to locate the minimum the *Gradient Descent* method is deployed. Gradient descent determines the slope of J in respect to every w_i using *partial derivatives.* Then:

$$\frac{\partial J}{\partial w_i} = \frac{1}{m}\sum_{j=1}^{m}(y_p^j - y^j)\cdot x_i^j \tag{3.25}$$

then weights are corrected following the relation:

$$w_i^{new} = w_i^{old} - \alpha\frac{\partial J}{\partial w_i} \tag{3.26}$$

α is the *learning rate*, which controls the learning speed. The choice of α impacts not only on learning speed but directly on convergence! Too large values make the algorithm to diverge, moving away from the minimum, too small values instead make convergence time endlessly long. There are today computationally more efficient optimizers, which deploy an adaptive learning rate such as AdaGrad [88], ADAM [87], and more. Correction of the weights goes on until a threshold on J is reached. SL algorithms suffer a problem called overfitting [52](or overtraining). Indeed, excessive training can build an extremely well-fitted model to the training data, lowering the overall performances when the same model is used to predict data that has never been seen from the algorithm. Some techniques can be deployed to lessen the chance of overfitting (e.g., Bayesian priors, Regularization [167], Dropout [157] etc.). The output of the whole algorithm is y_p, a continuous variable, which predicts the expected value of a generic y given a generic set of x_i. This technique is called *Linear Regression* if a set of parameters is used to predict a linear outcome, providing them with the linear model built. An alternative to linear regression is the *Logistic Regression.* Logistic regression is when the outcome is a boolean variable, 0 or 1. For this case, the function that parametrizes the hypothesis is called *Sigmoid function*:

$$y_P(z) = \frac{1}{1+e^{-z}} \tag{3.27}$$

where $z = w_0 + \sum_{i=1}^{n} w_i x_i$. y here assumes values between 0 and 1. Indeed y represents the probability for the outcome to be 1. The cost function for logistic regression is changed due to the Sigmoid non-linearity:

$$J(w_i) = -\frac{1}{m}\left[\sum_{j=1}^{m} y^j log(y_P^j) + (1-y^j)log(1-y_P^j)\right] \tag{3.28}$$

y^j is the $j-th$ label (containing a boolean value), after this point, the procedures are the same described for linear regression. The Sigmoid function used in this way assumes the role of an activation function. The role of the activation function is fundamental for neural networks and will be discussed later.

Logistic regression is the simplest SL approach. It can be considered as a binary Classification. Indeed Classification is a generalized logistic regression, where instead of having only 2 classes (1 and 0) we have N classes (e.g. 'star', 'galaxy' and 'quasar' labeled as the vectors [1,0,0], [0,1,0], and [0,0,1], respectively [11]) and the probability of the generic object to be classified is provided for each class. Typical classifiers are k-Nearest Neighbors, Naive Bayes, Random Forests, Support Vector Machines, Neural Networks, etc. The classification has an important role in astronomy, as it can be used to distinguish stars from galaxies (e.g., [170]) or to classify galaxies in their respective morphological classes (e.g.[62]), to estimate photometric redshift, etc.

3.2.2 Unsupervised Learning

UL provides a different approach to SL. The main difference with SL is the unlabelled data provided. Indeed UL does not need a training set, as it aims to find patterns in the dataset and then creates clusters composed of the data points. Clustering is used in astronomy to produce segmentation maps [162], which is fundamental for object detection, and it provides an attractive solution for the "blending" problem described in Chapt. 5.

Many clustering algorithms have been developed, K-means [106], DBSCAN [50], Mixtures [51], OPTICS [7] are just few examples. A visual result of their application on 6 datasets are shown in Fig. 3.1. In the next subsection the clustering algorithms, K-means, DBSCAN and DENCLUE are considered.

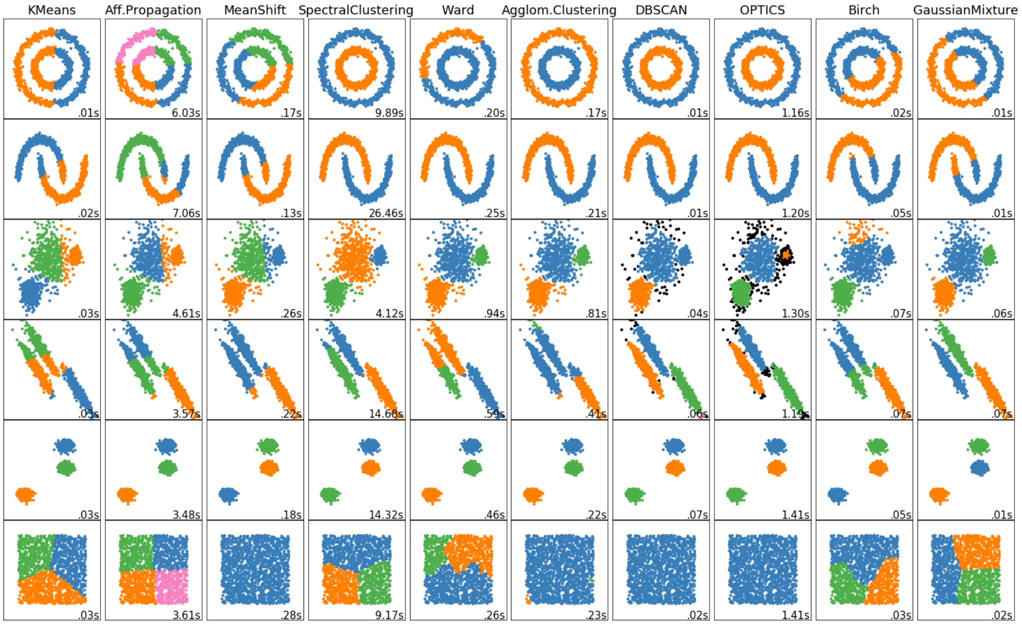

Figure 3.1. Visual comparison between clustering algorithms applied on different datasets [135]. Every row is a dataset and every column is the result provided by the algorithm. The computation time for the algorithms applied on the datasets is recorded in the bottom-right corner of every scatter plot. In order: K-means ($K = 3$) [106], Affinity Propagation [58], Mean Shift [39], Spectral Clustering [125], Ward [144], Agglomerative Clustering [144], DBSCAN [50], OPTICS [7], Birch [181][182], Gaussian Mixture [51]

K-means

Given a predefined number of clusters K, *K-means* [106] assigns each point to one of them. As the data points are unlabelled, the clusters use mean value of it components as representative for the cluster. Given a dataset of m elements, which is represented by x^i with $i = 1, 2, .., m$ each element is composed by n features x_j with $j = 1, 2, .., n$, the aim is to assign each x^i to a cluster C^k where ν_k is the cluster center, calculated by averaging all the positions of the data points assigned to C^k. The distance of x^i by each center is defined:

$$d = \sqrt{||x^i - \nu_k||^2} \tag{3.29}$$

Even in this case, a *cost function* minimization is required to find the optimal solution. The cost function is defined as the sum of all the distances between each data point and the cluster center assigned to that point:

$$J = \frac{1}{m} \sum_{i=1}^{m} \sqrt{||x^i - \nu_k^*||^2} \tag{3.30}$$

ν_k^* is the K-th cluster at which x^i has been assigned. According to J the algorithm updates C^k to contain the data points which minimize the overall distance, then ν_k are updated. These steps are repeated until a threshold on J is reached.

The choice of cluster's initial position is random. K-means is performed several times with different initial positions. After that, the convergence is reached, the algorithm with an initial position which minimizes the cost function is chosen. The only limitation to K-means is that the number of clusters K needs to be decided a priori. A great way to find the ideal K is to perform the algorithm several times, with a different number of K clusters. After that, plotting J versus K produces the $K - J$ curve. If the curve shows an "elbow" $K = K_{elbow}$ is the optimal number of clusters [160]. Indeed $K > K_{elbow}$ does not contribute significantly to J reduction, while $K < K_{elbow}$ gives too large J values. In some cases, $K - J$ curve may be too smooth to find an elbow. In such cases, the optimal K cannot be found in this way.

DBSCAN

The density-based spatial clustering of applications with noise (DBSCAN) is another clustering algorithm [50], which does not require a fixed number of clusters in the input. Furthermore, DBSCAN is able to find clusters of any shape, it has a definition of outliers, and it only relies on two parameters. Considering m data points to be clustered, a data point p can be: *core*, *directly reachable*, *reachable*, *outlier*. ϵ is the radius of a circle used to evaluate p neighborhood, while min_p is a minimum number of points. These two parameters are used in the following definitions:

- **Core**: if the number of points, including p within ϵ, is greater then a minimum number min_p, then p is a core point
- **Directly reachable**: if another point q is within ϵ from p, q is directly reachable

- **Reachable**: if a path of directly reachable core points connects a point q to the core point p, then q is reachable from p
- **Outliers**: all not reachable points are outliers

All the core and reachable points form a cluster. The non-core points form the edge. Since reachability is not symmetric, another definition, density connection, is required. Two points p and q are density connected if both p and q are reachable from a third point r. The density connection is symmetric. A cluster defined in this way satisfies two properties:

- All points in the cluster a density connected
- A point density-reachable from a point in the cluster is part of the cluster as well.

DENCLUE

A method based on density clustering is called DENCLUE (DENsity-based CLUstEring) [6][74], more suitable to handle large amount of noise and for clustering datasets with high-dimensional features. Differently from DBSCAN, the algorithm is not dependent on a radius parameter but uses a kernel density estimation. This estimation is an indicator of the probability density in the surrounding region. With the latest version of the software, a fast hill-climbing method coupled with Gaussian kernels are used for density estimation. The kernel density approximation of the probability density function is:

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_1}{h}) \tag{3.31}$$

where x_i is the independent and identically distributed sample of a random variable f, K is the kernel function and h is the bandwidth, which defines the smoothing. K is a gaussian with μ=0 and σ=1.

DENCLUE uses the kernel function to find points of density attraction, called attractors. An attractor is a local maximum of the estimated density function. An attractor x^* is validated using a threshold ξ such that: $f(x^*) \geq \xi$.

Each data point is associated with the clusters using the attractors, using the fast hill-climbing procedure:

$$x_{j+1} + \delta \frac{\nabla f(x_j)}{|\nabla f(x_j)|} \tag{3.32}$$

where x_j and x_{j+1} are the coordinates of the generic point x at the iteration j and $j + 1$ respectively, δ is the speed of convergence. A cluster is defined by a set of attractors and input points. Each input point is assigned to an attractor.

3.2.3 Neural Networks

Neural Networks or Artificial Neural Networks (ANNs) were developed with the idea of reproducing a human brain. Indeed some key features of ANNs directly come from our knowledge of how the human brain works. In our brain, neural networks are composed of Neurons, in the same way, ANNs are composed of artificial neurons. Without going into many details, a neuron is an electrically excitable cell [133], which receives multiple inputs by branched protoplasmic extensions called Dendrites [89] producing a single output which transferred to other neurons with structures called Synapses. [56]. Synapses transfer information from a neuron to the dendrites of the following. A neuron can indeed receive electrical inputs from vital organs or other neurons. The strength of each synapse can vary with time as the brain learns and is "trained" it increases or decreases the strength of the various synapses. These processes in neuroscience are called Long-Term Potentiation (LTP) [23] and Long-Term Depression (LTD) [111]. In an ANN, artificial neurons (or also called nodes) assign real values weights to their connections with the following neurons, resembling the strength of the synapses. The ANN has several initial neurons which fits the input data (the number of columns of an input dataset, where are stored the features of each data point), and a fixed number of neuron in output, according to the linear regression/ logistic regression/classification problem. The ANNs have also an arbitrary number of intermediate layers (called hidden layers), each populated by neurons. The neurons in the hidden layer can vary with the layer. Also, connectivity between layers can vary. The neurons from 2 adjacent layers can be all connected (in this case, we talk about fully connected or dense layers) or only partially connected. Several design decisions must be taken to build up a neural network[65]:

1. The number of hidden layers
2. The number of neurons for each hidden layer
3. Connectivity between nodes of adjacent layers (dense or not)
4. The activation function used for each layer
5. Cost function used for minimization
6. The optimizer used for the optimization

Like for linear/logistic regression, for neural network, a cost function dictates how weights need to be updated. The optimizer has the role of providing the slope of the cost function toward the local minimum, and the activation function transforms the output of neurons in comparable quantities. For simplicity, let us consider a neural network, with one single hidden layer composed of 1 neuron. This network is provided with a two-dimension dataset with m elements. x_1 and x_2 are the features of the $i-th$ data point in the dataset, w_{11} (w_{12}) is the weight that connects x_1 (x_2) to the first neuron, while w_{21} is the weight that connects the first (and only) neuron in the hidden layer to the output. b_1 and b_2 are the biases of the neuron in the hidden layer and the one in the output layer, respectively. Through w_{11} and w_{12}, the first neuron makes a weighted sum (z_1) of x_1 and x_2:

$$z_1 = x_1 w_{11} + x_2 w_{12} + b_1 \tag{3.33}$$

as seen in the logistic regression, the output of the neuron can be modified by an activation function f. Let's call the output of the activation function from the first neuron as a_1 and from the second neuron a_2. a_2 is the predicted value, provided x_1 and x_2 from this sample neural network.

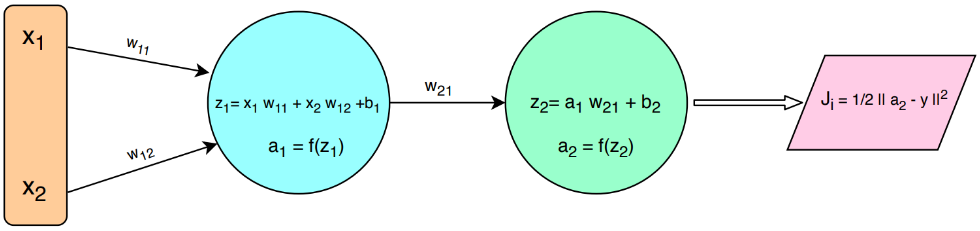
x_1
x_2
w_{11}
w_{12}
$z_1 = x_1\, w_{11} + x_2\, w_{12} + b_1$
$a_1 = f(z_1)$
w_{21}
$z_2 = a_1\, w_{21} + b_2$
$a_2 = f(z_2)$
$J_i = 1/2 \,||\, a_2 - y \,||^2$

Figure 3.2. A visual representation of the sample neural network defined in Sect. 3.2.3

Cost Function minimization

As for regression, the cost function J is calculated:

$$J_i = \frac{1}{2}||a_2 - y||^2 \tag{3.34}$$

a_2, y and J_i are in referred to the $i-th$ data point, the output of the neural network, the label and the contribute to the cost function respectively. The total error (or total cost function) is expressed as:

$$J = \sum_{i=1}^{m} \frac{1}{2m}||a_2 - y||^2 \tag{3.35}$$

To update weights and biases the Eq. 3.26 is required:

$$\frac{\partial J}{\partial w_{lk}} = \frac{\partial J}{\partial a_l}\frac{\partial a_l}{\partial z_l}\frac{\partial z_l}{\partial w_{lk}} \tag{3.36}$$

$$\frac{\partial J}{\partial b_l} = \frac{\partial J}{\partial a_l}\frac{\partial a_l}{\partial z_l}\frac{\partial z_l}{\partial b_l} \tag{3.37}$$

which for w_{21} is:

$$\frac{\partial J}{\partial w_{21}} = \frac{\partial J}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial w_{21}} \tag{3.38}$$

and for b_2 is:

$$\frac{\partial J}{\partial b_2} = \frac{\partial J}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial b_2} \tag{3.39}$$

To update weights in previous layers, the number of partial derivatives increases, as an example to update $w_1 1$:

$$\frac{\partial J}{\partial w_{11}} = \frac{\partial J}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_{11}} \tag{3.40}$$

the complexity of the update obviously increases with the increasing number of hidden layers. It is possible to notice that if the Sigmoid described in Eq. 3.27 is used as activation function, all the a_l will be between 0 and 1, so with the increasing number of partial derivatives the weights in the first layers will "learn" at a slower rate then those in the last layers. To prevent this issue, called the "Vanishing Gradient" [178], another activation function is usually preferred, the Rectified Linear

Unit (ReLU) function[70]:

$$f(x) = max(0, x) \tag{3.41}$$

using ReLU, the output of the neurons in between 0 and ∞, and the vanishing gradient problem is solved. As the output of the neural network is required to classify data points, the ReLU activation function cannot be used. In his place the Softmax function [21] is generally used:

$$a_l = \frac{e^{z_l}}{\sum_{p=1}^{P} e^{z_l}} \tag{3.42}$$

Where p represents the generic "class" reported in the label and P the total number of classes. Softmax provides a probability for a data point to belong to the class p

3.2.4 Convolutional Neural Network

In Sect. 1.4.2 we introduced convolution. Convolution plays an important role in a specific category of neural networks. This particular kind of network is called Convolutional Neural Networks (CNNs). The kind of input data is assumed to be an image. Convolution is used to discover patterns, using different trainable filters (kernels). CNNs can be employed for image classification, object detection, and other computer vision applications.

CNNs architecture

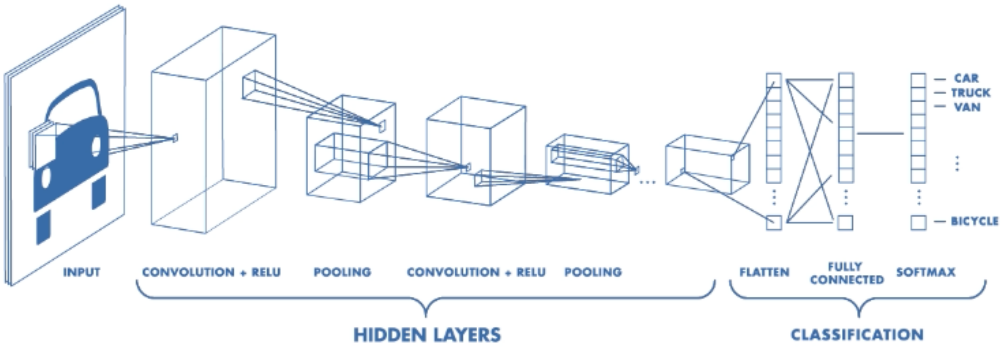

Figure 3.3. A typical CNN architecture for classification [57]

The first layer of a CNN is a convolutional layer. Here the input image (grayscale or RGB) is filtered by an arbitrary number of kernels with arbitrary shape (kernels must only match the depth of the image, 1 for grayscale and 3 for RGB). The number of filters K, their shape F, and the stride S are called hyperparameters and are chosen when the CNN before learning, then their values are fixed. An Edge detection filter can be used to pre-process the images to make them sparse matrices, which reduces computational time [164][65]. The convolutional layer output is several convolved images (sometimes called channels) equal to the number of filters. As for the other ANNs and activation function is coupled to each convolutional layer, and is generally used the ReLU function for the same reasons cited in Sect. 3.2.3.

After a convolutional layer, a pooling layer is often added. A pooling layer has the task of reducing the data size through post-processing. An average or a maximum pooling layer is generally used. The average pooling filters the image with kernels of an arbitrary shape F and a stride $S \geq 2$, averaging the pixels and creating a feature map of dimensions S times smaller. The maximum pooling (max pooling) instead takes the maximum. Pooling is needed to shorten training time and controlling overfitting [57]. The max-pooling is generally preferred.

After a pooling layer, several other convolutional layers can be attached, followed by another pooling layer, and so on. The convolutional part is then concluded and now begins the classification part. After several convolutional+pooling layers, the initial image has been transformed into a data cube. The shape of this data cube is dependent on the hyperparameters K, F, and S is chosen for each layer. In the end, the data cube is flattened. Each element of the cube is considered the input of a fully connected classical ANN. While a classifier characterizes the output layer. Usually the Softmax activation function [57] is used.

Segmentation with CNNs

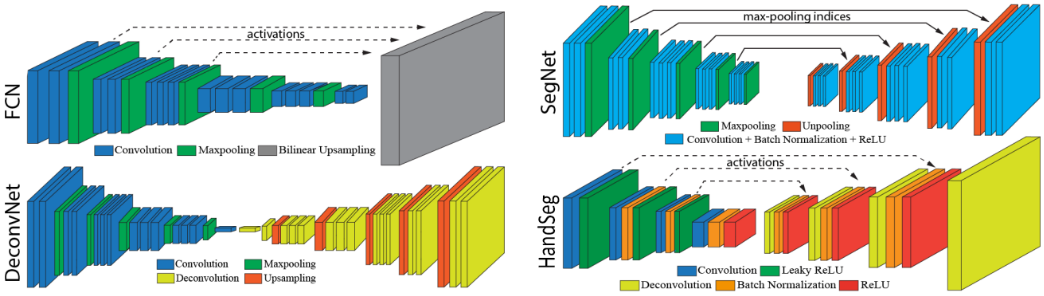

Figure 3.4. Some Segmentation CNNs architectures [24]

Along with classification problem, CNNs can be deployed to produce segmentation maps. Segmentation maps are particularly useful in astronomy for object detection, and CNNs provide a new solution to the blending problem (described in Chapt. 5). The aim is to obtain a segmentation map of each input image, performing an up-sampling at the end of the already discussed convolutional network, called encoder.

The simplest implementation to produce segmentation maps with CNNs is to remove the flatten layer and convert the fully connected layer to a 1×1 fully connected convolutional layer, with the number of filters (channels) K corresponding to the number of classes desired. Then an up-sampling layer with a fractional stride S^* interpolates the 1×1 convolutional layer outputs. The FCN model [104] implements this strategy. FCN model up-samples this output by a factor ×32, this architecture is called FCN-32s. Unfortunately, up-sampling with a large factor deteriorates the quality of the label map. Indeed going deeper in terms of convolutional/pooling layers provide deep features, but also spatial location information is lost [163]. To handle this issue, FCN connects distant layers bypassing the adjacent ones by adding, pooling layers a different depths (each different encoding degrees of depth/spatial information), and then transferring the information. This connection is called a skip connection. The architectures obtained in this way are FCN-16s and FCN-8s, up-sampling the label map respectively ×16 and ×8. The results make clear that FCN-8s provides the best segmentation [163][104].

Another strategy adopted is to attach a fully up-sampling convolutional network, to the encoder, taking the role of the decoder.

SegNet [10] and DeconvNet [126] follow the encoder-decoder architecture. Rather than relying on interpolation, both architectures employ un-pooling operations. Similarly to DeconvNet, SegNet up-samples the feature maps via memorized max-pooling indices in the corresponding encoder layer. SegNet also employs a series of convolutional layers, while DeconvNet employs deconvolutional layers. Deconvolution is the transpose of a convolution, represented by the gradient of a convolutional layer, making DeconvNet more computationally intensive, but providing similar results in terms of accuracy [10].

U-Net [145] model employs architecture similar to SegNet and DeconvNet. Down-sampling using max-pooling and up-sampling using up-convolutions. Similarly to FCN, skip connections are performed through a concatenation operator (instead of a sum) between layers from the decoder to the corresponding encoder layer. At the final layer, a pixel-wise convolution is used to build the label map.

HandSeg [24] models a hybrid encoder-decoder employing a hierarchy of deconvolution layers without employing max-pooling and un-pooling layers. Similarly to FCN and U-Net, the information in the encoder is kept and provided to the decoder. Downsampling and up-sampling are achieved by $S = S^* = 2$ in convolutional/deconvolutional layers.

All the CNNs mentioned above employ a final 1×1 fully connected convolutional layer to produce the segmentation map, with the number of classes (filters) required.

In Chapt. 5, a method based on CNNs, which handles separation between blended objects, is described.

Chapter 4

Denoising Comparison for Extragalactic Imaging Surveys

Measuring the amount of photons that we receive from astronomical sources over a given range of wavelengths is the primary way to gather information about the Universe. From the advent of digital photography in the 1980's, charge-coupled device (CCD) imaging is one of the primary ways by which we do so. Currently, CCDs can reach 100 million pixels, with read noise as low as one electron, almost 100% quantum efficiency, and sensitivity from the X-rays to the near infrared. Before being ready for the extraction of meaningful scientific content, astronomical images must be processed to, for instance, combine different observations into a single mosaic, correct for flat-field, transients, artifacts, and defects, subtract a global or local background, etc. Once these preparatory steps are completed, the quality of the image mainly depends on its resolution capability (which is usually proportional to λ/D, the ratio between the observed wavelength and the diameter of the telescope, in the case of diffraction-limited instruments, e.g. space observatories; or from the atmospheric seeing for ground-based facilities), and on its depth (i.e. magnitude at a given reference signal-to-noise ratio), which mainly depends on the duration of the observations (exposure time). Since increasing the latter is often unfeasible or too demanding, searching for alternative methods to increase the signal-to-noise ratio (SNR) is important. A possible solution can be the application of noise reduction ("denoising") techniques.

Typically, extragalactic images are convolved with a PSF shaped kernel to enhance source detection [an application of the lemma by 124]; this is the most standard example of a denoising algorithm, since filtering reduces the noise variance, allowing real sources to raise above the background. In many familiar cases, the typical PSFs of telescopes are quite similar to 2D Gaussians, making the PSF filtering basically indistinguishable from a Gaussian filtering. However, in many non-astronomical applications of image analysis this approach is often outclassed by other, more refined methods, designed to be more efficient and to better preserve the borders and edges of the sources. The goal of this chapter is to compare several classes of denoising techniques, in order to find which ones yield the best improvements in source detection. To this aim, we have performed an extended set of tests. We considered many different noise reduction algorithms, roughly belonging to the following families: Perona-Malik (PM) filtering, Bilateral filter, Total Variation (TV) denoising, Structure-texture image decomposition, Non-local means, and Wavelets. Note that the numerical methods employed range from variational methods to PDEs-based techniques, also including some statistical methods. We tested them using two different datasets. First, we focused on simulated images, created by state-of-the-art codes and prescriptions in order to mimic different realistic cases. This simplified environment has the advantage to allow a detailed analysis of the results, since the "truth" is perfectly known. For real images, we applied the algorithms giving the best results obtained on the simulated dataset to check if the improvement is confirmed. To our knowledge, this is the first attempt to extensively compare a large number of denoising algorithms in an astrophysical context. In general, the performance of any of these methods depends on the kind of noise that affects the image. Here we are mainly interested in extragalactic imaging, and in particular we focus on the next-generation of optical - near-infrared instruments and surveys such as Euclid [97], LSST [105], DES [43], and WFIRST [156]. Throughout this chapter we adopt the AB magnitude system [130] and a ΛCDM cosmology with $\Omega_m = 0.3$, $\Omega_\Lambda = 0.7$, $H_0 = 70\ Kms^{-1}Mpc^{-1}$. The content of this chapter is part of a scientific paper (submitted to *Astronomy & Astrophysics*) at the time of this thesis submission, and it is currently under the referee revision.

4.1 Methods

In this section we present how the methods already described in in Sect. 3.1 have been implemented for the analysis. All the algorithms shown here are analyzed in Sect. 4.3.

Gaussian Filter

We have used a simple Gaussian smoothing using a kernel that approximates a PSF of known FWHM, referring to it as "PSF". While with "Gaussian" we refer to the Gaussian filter with internal parameter σ.

We made use of the `gaussian_filter` routine implemented in the Python package Scipy[1] [82], with $\sigma \approx \frac{FWHM_{pixel}}{2.355}$, easily obtained defining the Gaussian kernel radius $r = x^2 + y^2$, where the kernel maximum is at $r = 0$ then $FWHM = 2\sqrt{2\ln 2}\sigma$, see also https://brainder.org/2011/08/20/gaussian-kernels-convert-fwhm-to-sigma/ for further details.

Perona-Malik filter

This method has been developed by us in C++ and it is available at: https://github.com/valerioroscani/perona-malik.git.

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.gaussian_filter.html

Bilateral filter

We used the Python routine `denoise_bilateral` available in the Python package SCIKIT-IMAGE[2]. We noticed that using our dataset, variations of the `sigma_spatial` were less effective than variations of `sigma_color` (see Sect. 3.1.3 for the parameters definitions). We decided to set `sigma_spatial` = 3 since it provides the best results.

Total Variation denoising

We test the ROF method that was proposed by Chambolle in [34] and the TV denoising using split-Bregman optimization [64, 63, 30]. For the implementation of the two aforementioned methods we have used the Python routines `denoise_tvchambolle` and `denoise_tv_bregman` belonging to the Python package SCIKIT-IMAGE [2] [168].

Structure-texture decomposition

In [32], the authors proposed a C++ code named Astro-Total Variation Denoiser (ATVD), which implements three versions of the technique, based respectively on the TV-L^2 ($X = L^2(\Omega)$), TV-L^1 ($X = L^1(\Omega)$) and TVG (X being a Banach space as defined in [9]) norms. Two thresholds are defined and used in the stopping criteria of the algorithms, called ϵ_{corr} and ϵ_{sol}.
ϵ_{corr} defines the correlation algorithm precision, whereas ϵ_{sol} defines the method precision (e.g. TVL2, TVG, TVL1). For all our tests, we will use $\epsilon_{corr} = 10^{-4}$ and $\epsilon_{sol} = 10^{-3}$, as suggested by the authors in [32].

Wavelets

We consider the two thresholding methods defined in the Python routine `denoise_wavelet` [2] [35, 49], the first applies BayesShrink, which is an adaptive thresholding method that computes separate thresholds for each wavelet subband as described in [35], the second is “VisuShrink”, in which a single “universal threshold” is applied to all wavelet detail coefficients as described in [49]. This threshold is designed to remove all Gaussian noise at a given σ with high probability, but tends to produce images that appear overly smooth.
In this work we decided to apply the Meyer wavelet described in [116] with VisuShrink thresholding method since, analyzing the application on our dataset, we found that it provides the best performances based on the analysis described in Sect. 4.3. The list from which we took the Meyer wavelet can be found in [99].

4.2 The dataset

We first test the denoising algorithms on five different simulated images (Tab. 4.1), chosen as to reproduce the properties of a wide range of typical cases in terms of resolution, depth, pixel scale and wavelength:

- VIS: Euclid satellite visual band (wavelength: 550-900 nm)

[2] https://scikit-image.org/docs/dev/api/skimage.restoration.html

- NIR H: Euclid satellite near infrared H band (wavelength: 1372-2000 nm)
- EXT G: ground-based optical filter
- H160: Hubble Space Telescope (HST) near infrared F160W band [e.g. CANDELS-wide 69]
- IRAC: Irac-Spitzer 3.6μm channel.

From now on, we refer to the simulated images, provided as input to the algorithms, as "Original", while we refer to the simulated images representing the true sky, without noise included, as "Noiseless".

Filter	PSF-FWHM	Pixel Scale	Mag Lim$^{(a)}$
	arcsec	arcsec	
VIS	0.2	0.1	25.25
NIR H	0.3	0.3	24.01
EXT G	0.8	0.2	25.93
H160	0.15	0.06	27.23
IRAC	1.6	0.6	25.40
HUDF (H160)	0.15	0.06	28.16/29.74$^{(b)}$
Ks (HAWK-I)	0.4	0.1	24.45/26.3$^{(c)}$

Table 4.1. $^{(a)}$: SNR=5;$^{(b)}$: limiting magnitude at the CANDELS and at the full HUDF depth, respectively; $^{(c)}$: images from [31], and from the HUGS survey [54], respectively.

VIS and NIR H reproduce the expected features of the visual and near-infrared bands in the forthcoming ESA satellite Euclid [97], and EXT G simulates a typical ground-based complementary optical observation for the Euclid Wide Survey. H160 is modeled after the detection band in recent deep surveys such as CANDELS [66, 90] and 3D-HST [154], whereas IRAC simulates the features of the *Spitzer* Channel 1 band in the CANDELS GOODS-South field [69].

The images have been simulated with *SkyMaker* [18] on the basis of source catalogs generated by the Empirical Galaxy Generator (EGG) [150] and they have been perturbed by Gaussian noise in order to reach the limiting magnitudes reported in Tab. 4.1. All the PSFs are Gaussian except for the IRAC case where a real IRAC 3.6μm channel PSF has been used. The H160 and HAWK-I images are real observations whose tests are described in Sect. 4.5.1-4.5.2.

We can sort the simulated images in several different ways:

- Depth, from the deepest to the shallowest: $H160 > EXT\ G > IRAC > VIS > NIR\ H$
- PSF, from the sharpest to the coarsest: $H160 > VIS > NIR\ H > EXT\ G > IRAC$

- Pixscale, from the smallest to the largest: $H160 > VIS > EXT\ G > NIR\ H > IRAC$.

For each simulated image, we cut three independent areas of the sky, which are the same for every band but differ in dimensions due to the different pixel scale. The regions are listed below:

- BG: centered on a big elliptical galaxy (see Fig. 4.1)
- CL: centered on a cluster of galaxies (see Fig. 4.2)
- CM: an average portion of the sky (see Fig. 4.3).

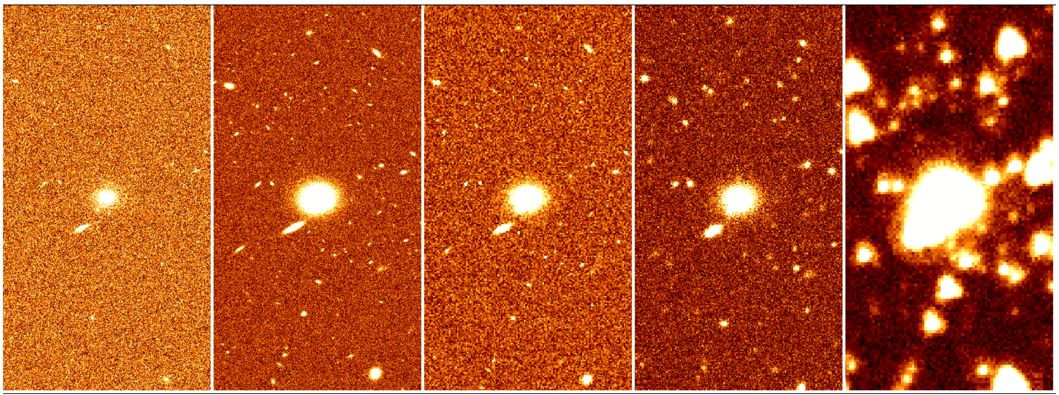

Figure 4.1. From left to right: Crops of the BG (Big Galaxy) image central area for VIS, H160, NIR H, EXT G and IRAC

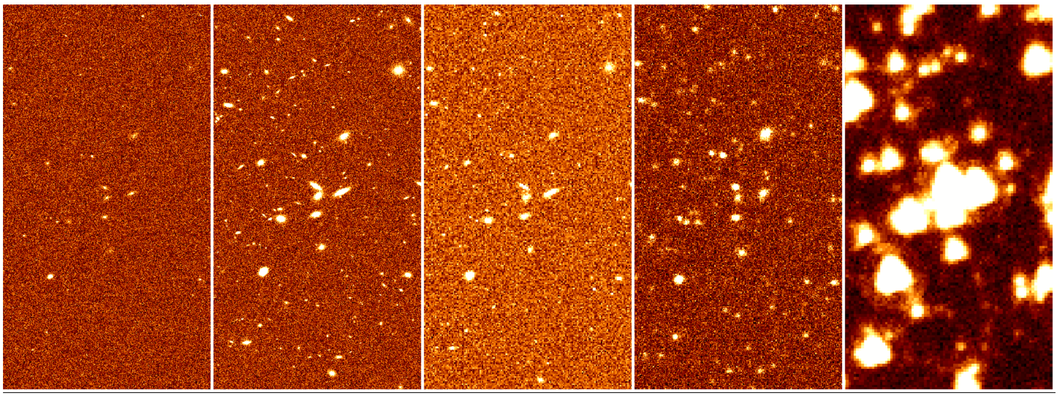

Figure 4.2. From left to right: Crops of the CL (Cluster) image central area for VIS, H160, NIR H, EXT G and IRAC

The three regions have a dimension of:

- VIS: 1000x1000 pixels
- NIR_H: 333x333 pixels
- EXT_G: 500x500 pixels

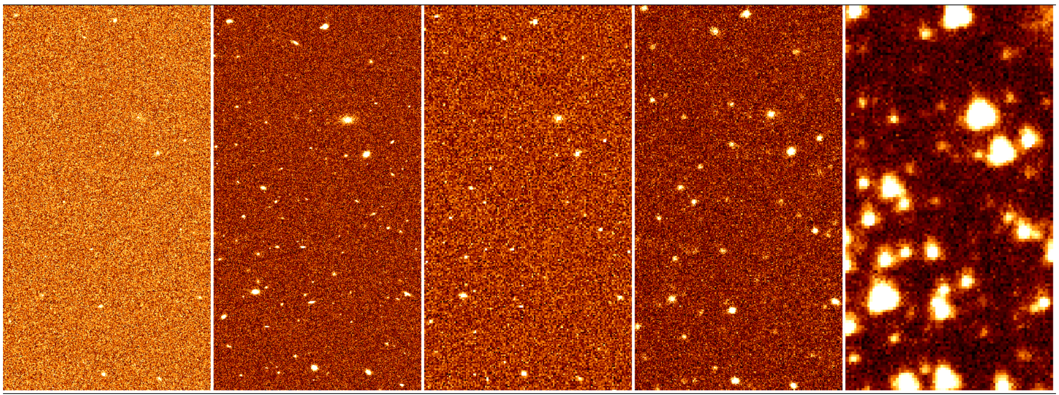

Figure 4.3. From left to right: Crops of the CM (Average field) image central area for VIS, H160, NIR H, EXT G and IRAC

- H160: 1666x1666 pixels
- IRAC1: 166x166 pixels.

After the analysis described in Sect. 4.3, commenting the results obtained in Sect. 4.4, additional tests on real images (see Tab. 4.1 for details) ground-based (HAWK-I) and from space (HST) are reported and analyzed in Sect. 4.5.

4.3 Quality tests

The idea at the basis of the analysis is to first evaluate the algorithms through different tests, in order to apply only the most promising ones (with their best configurations) on real images. We organize our analysis on the five simulated images in different levels of testing. A brief description of each step is given below:

1. As a first step we compare the algorithms performances through three parameters: *mean square error* (*MSE*), *structural similarity* [SSIM 173] and *CPU time*. The *MSE* is defined as:

$$MSE := \frac{\sum_{i=1}^{N}(x_i - \widehat{x_i})^2}{N} \tag{4.1}$$

where x_i is the i-th pixel in the denoised image and $\widehat{x_i}$ is the i-th pixel in the original image (without noise). The SSIM is defined as:

$$SSIM := \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.2}$$

where μ_x is the average of x, μ_y is the average of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, σ_{xy} is the covariance of x and y, c_1 and c_2 are constants proportional to the dynamic range of the pixel values. The *CPU time* is the computational time required by the algorithms to filter the image. Through these tests we identify the main internal parameters of each algorithm and their ideal values.

2. We test the performances of the algorithms selected in the previous step as a function of the FWHM of the PSF and as a function of the background noise level.

3. We test the stability of the algorithms selected in the previous steps against variations of the main internal parameter value (identified in Step 1), measuring how the MSE varies as a function of the parameter values.

4. We test how the PSF FWHM is affected by the selected denoising algorithms checking if they preserve the FWHM of point-like objects and the galaxies profiles.

5. We test the selected algorithms, studying two parameters, *completeness* and *purity*, which provide a quality estimate of the catalog produced after an ideal source detection, exploring a combination of detection parameters.

6. As last step, we test if the denoised images can be used also for photometry measurements, analyzing if the object fluxes are preserved after denoising.

Finally, we apply the best performing algorithms of our selection on real images acquired from space and ground-based telescope, as described in Sect. 4.5.

4.3.1 Implementation details

We compare the different images, following always the same procedure here described:

- The Original image is scaled to the range $[0, 1]$

- The Original image is filtered by the denoising algorithm providing the *denoised* image

- The *denoised* image is scaled back to [$Original_{min}$,$Original_{max}$], where $Original_{min}$ and $Original_{max}$ are the maximum and minimum values in the Original image, using the following equation:

$$x^i_{Original} = (Original_{max} - Original_{min}) * x^i_{[0,1]} + Original_{min} \tag{4.3}$$

  where $x^i_{Original}$ is the $i-th$ pixel in the Original image and $x^i_{[0,1]}$ is the $i-th$ pixel in the *denoised* image scaled to [0,1]

- *MSE* and *SSIM* are computed by comparing the *denoised* image to the noiseless one.

In order to choose the best internal parameter for each denoising algorithm (a list of these parameters is in Sect. 4.4.1), we used different stopping criteria:

- **ATVD**: In ATVD is already implemented a stopping rule, through a minimization problem, as described in Sect. 3.1.5

- **Perona-Malik**: In PM code we have a stopping rule composed by 3 conditions: in the first one we compare at each time step MSE_n with MSE_{n-1} where MSE_n is the MSE at the current time step, whereas MSE_{n-1} is the MSE at the previous time step. The code continues running as $MSE_{n-1} - MSE_n > 0$. The second condition concerns the number of iterations n: the code continues running until the number of iterations does not exceeds the maximum number of iterations NMAX, which is set to NMAX=500. The third condition is $\mid \frac{MSE_{n-1}-MSE_n}{MSE_{n-1}} \mid \leq \epsilon$, with $\epsilon = 10^{-10}$
- **Other denoising algorithms**: For all the other denoising algorithms we choose the main parameter(s) in the parameter space by an iterative process with an adaptive step, computing the MSE at each iteration. The stopping rule is reached when the step is smaller than an ϵ, set to 10^{-10}, and the number of iterations is lower than the maximum number of iterations NMAX, which is in this case set to NMAX=100.

4.4 Results

In this section we analyze and comment in a separate and sequential way the results related to the quality tests, following the same order of the steps used in Sect. 4.3.

4.4.1 Ranking with *MSE* and SSIM

In this test we use three parameters to constrain the performances of denoising methods: *MSE*, *SSIM* and *CPU time* (Sect. 4.3). We give priority to those algorithms that are able to minimize as much as possible the *MSE*, preferring the fastest method (in terms of *CPU time*) and the highest *SSIM* in case of comparable *MSE*. Following this criterion, in this step we identify the best configuration and the main parameters for every algorithm. These results are taken into account separately for all the simulated images presented in Sect. 4.2.

The main internal parameters identified for the different algorithms are listed below:

- Wavelets: `sigma` - The noise standard deviation used for compute threshold(s)
- NL means: `h` - Cut-off distance in grey levels
- TV Bregman: `weight` - Denoising weight, efficiency of denoising
- TV Chambolle: `weight` - Denoising weight, efficiency of denoising
- Gaussian: `sigma` - Standard deviation for Gaussian kernel
- Bilateral: `sigma_color` - Standard deviation for grey value distance
- Perona-Malik: `T` - Number of iterations of the anisotropic diffusion
- ATVD (TVL1,TVL2,TVG): λ - Structural-Texture splitting parameter.

Further details for the algorithms implemented in Python and the measurement of *MSE* and *SSIM* can be found in the SCIKIT-IMAGE documentation [3]. The method used to identify the best internal parameter for each algorithm is described in Sect. 4.3.1 In Appendix A we show the best *MSE* and CPU time values of every algorithm, for the different crops. The tables are organized to record the best *MSE* and CPU time values obtained with the algorithms. The columns represent the different image simulated filters and the value indicated in bold is the lowest of the column. Tab. A1-A2-A3 contain the *MSE* values for the crops BG, CM and CL respectively. While Tab. A4 contains the CPU time values for the crop CM. We remind that in the following “PSF filtering” amounts to filtering with a Gaussian whose FWHM is the same as the PSF-FWHM, while in the case of the “Gaussian filtering” the σ (and thus the FWHM) is a free internal parameter. We shortly summarize here the main results:

- TVL2, PM, NLmeans slow and TV Chambolle always yield good performance, typically providing the lowest values of *MSE*
- TVL2, PM, NLmeans slow, TV Chambolle always perform better than Gaussian filtering, with the only exception of the IRAC image (we discuss the IRAC situation below in Sect. 4.4.2)
- the *MSE* of all the methods is proportional to the pixel scale of the image, so that low sampling implies worse results
- in most cases (with the exception of IRAC, which we discuss below), the PSF filtering provides a larger (i.e. worse) value of the *MSE* compared to the one provided by Gaussian filtering.
- in some cases, the *MSE* of the denoised image is larger (i.e. worse) than the one measured without denoising the image at all. Indeed some algorithms in the situations listed below tend to over-smooth the image, providing a worse MSE. This event occurs:
  1. in VIS (CM) image, in the case of the PSF filtering
  2. in all the H160 images for both the PSF filtering and TV Bregman
  3. 2-4 times in NIR H images for methods NLmeans fast, wavelets, TV Bregman and PSF filtering
  4. only once in EXT G (CM), for the PSF filtering
  5. 4 to 5 times in IRAC images, for NLmeans slow, NLmeans fast, TV Bregman, Wavelets and PSF filtering.
- the *SSIM* ranking typically reflects the *MSE* ranking, pointing out the same group of best algorithms found in the *MSE* ranking; even if some positions are swapped in few cases, the *SSIM* values provided by the best algorithms are comparable ($\Delta SSIM < 10^{-4}$)

If we focus on the algorithms belonging to the same classes of methods, we can note that:

[3] https://scikit-image.org/docs/dev/api/skimage.restoration.html

1. TVL2 performs clearly better than TVG and TVL1, e.g. in BG, $1 - \frac{mse}{mse_{Original}}$ value is always within 5% from the value provided by the original image (no noise), with the exception of IRAC, where it drops to 0.2, which is still greater than the values provided by TVG and TVL1, as shown in Fig. A2

2. all the PM methods yield similar performances, (see Fig A1), and therefore we choose to only keep $g = g_1$ with the parameter k set to $k = 1e - 3$ in the following steps

3. NLmeans slow performs slightly better than NLmeans fast for H160,VIS and EXT G, ($1 - \frac{mse}{mse_{Original}}$ differences are within 5% in favor of NLmeans slow) and much better for NIR H and IRAC (where NLmeans fast performs worse than Original)

4. TV Chambolle performs better than TV Bregman, in H160, NIR H and IRAC, TV Bregman performs worse than Original, whereas for VIS and EXT G it performs (14% and 3% worse than TV Chambolle, relatively) (see Fig A3)

5. Bilateral is always within the best performing techniques (see Fig A3)

We nevertheless keep Wavelet, Gaussian and PSF filtering for reference, since they are widely used. Hence, at the end of this first step we are left with 8 methods: PM with edge-stopping function g_1 and $k = 1e - 3$, TVL2, Gaussian, PSF, NL means slow, Bilateral, TV Chambolle and Wavelets. Following our experiments analysis we decide to discard 8 algorithms: 4 PM methods, TVG and TVL1, NL-means fast and TV Bregman.

4.4.2 The IRAC results

We note that the IRAC images do not follow the same trends as the other bands. While for all the other images there is always a small group of algorithms which perform better than all the others, for IRAC nearly all the denoising algorithms tend to have similar performances. We investigated the possibility that the number of pixels were not enough (166x166 pixels) to extract significant conclusions from these images and we tested the algorithms on an IRAC (CM) simulation with pixel scale 0.1 arcsec and size of 1000 × 1000 pixels. We noticed that, TV Chambolle, NL means slow and Gaussian provide the best performances (MSE $\propto 10^{-9}$), followed by TV Bregman, TVL2, PM (g=g1 k=0.01) (MSE $\propto 1 \times 10^{-8}$), PSF (MSE $\propto 2 \times 10^{-8}$) and then Bilateral, PM (g=g1 k=0.001) and Wavelets (MSE $> 2 \times 10^{-8}$). After this small test we point out that again the IRAC band doesn't follow the trend defined in the other bands (even if the *MSE* decreases for all the methods and the Original image), but with the increased number of pixels TV Chambolle, NL means slow and Gaussian are the algorithms which provide the best performances. The low resolution of IRAC here plays a fundamental role, impacting on most of the algorithms performances. This aspect of the algorithms will be described in the next Sect. 4.4.3.

4.4.3 Stability against FWHM and depth variations

In this second test we compare the performances of the 8 algorithms with respect to the variation of the FWHM and depth of the images. We consider two cases:

- A 1000x1000 pixels crop of the simulated VIS image convolved with different kernels, to degrade the resolution increasing the FWHM without changing the depth of the image (we considered the cases FWHM = 0.5, 1, 1.5 and 2.0", with the original FWHM being 0.2")
- We decreased the depth of a 1000x1000 crop of the simulated H160 image without changing the FWHM, by adding Gaussian noise with increasing standard deviation σ $(\times 1, 10, 20, 30$ and 40 times the original one) to the Noiseless image.

The plots summarizing the results are shown in Figs. B1- B6, in Appendix B. We can note that:

1. The MSE calculated on the original image alone decreases at increasing FWHM due to the loss of information (i.e. small objects and details). All algorithms follow this trend while lowering the *MSE* even more due to the effect of filtering (see Fig. B1)
2. The ratio between the MSE obtained by each algorithm and the MSE computed on the original image $(\frac{mse}{mse_{Original}})$ increases at increasing FWHM, with the only exception of the Gaussian filtering which instead follows the opposite trend (see Fig. B2)
3. $\frac{mse}{mse_{PSF}}$ is weakly affected by variations of the FWHM for most of the denoising methods, with the exception of Gaussian (see Fig. B3)
4. as expected, the MSE increases at increasing background level (due to the increasing of σ for the Gaussian noise) both in the original image and in the output denoised images for all the algorithms (see Fig. B4)
5. $\frac{mse}{mse_{Original}}$ decreases at increasing background level for all the methods(see Fig. B6)
6. $\frac{mse}{mse_{PSF}}$ increases at increasing background level for all the methods (see Fig. B5).

Summarizing, we conclude that the best performances by any denoising algorithm are obtained on images with low SNR and high resolution (narrow FWHM). The best performances with respect to the PSF method are obtained by applying the denoising methods on image with high SNR, regardless of the PSF FWHM. These results can be used to estimate the efficiency of the denoising algorithms in different situations, underlining that when applied to high resolution images they provide the best improvements, whereas if applied on low SNR images (where there is the peak of performances), the improvements compared to the PSF are slightly less significant. From these results, it would be very interesting to apply these methods, as an alternative of the PSF filtering, on images with high resolution and high SNR.

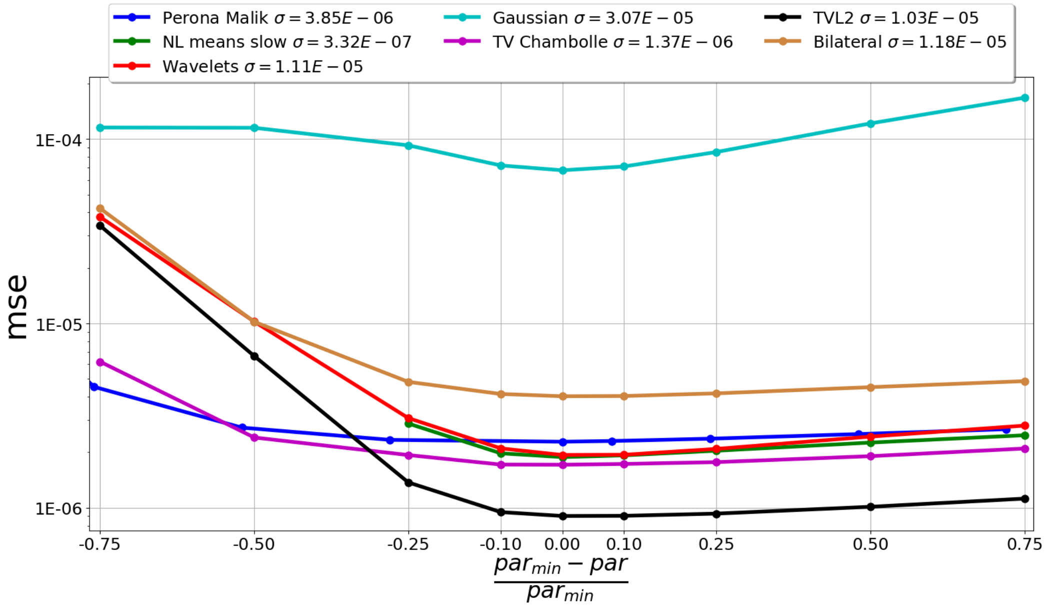

Figure 4.4. Step 3: Stability against variations of the parameters. Each curve corresponds to a denoising algorithm. We plot the *MSE* against the relative variation of the parameters, $\frac{par_{min}-par}{par_{min}}$. Obviously the absolute minimum of the curves is reached in 0 on the x-axis, corresponding to the ideal value of the parameter. In the upper panel we report the standard deviations of the $mse_{mean} - mse$ distributions.

4.4.4 Stability against variations of the parameters

In this test we analyze the selected methods by varying the values of those internal parameters that had been kept fixed to the optimal ones in the previous tests. The goal is to understand whether the performances are stable against sub-optimal parameter settings. We exclude the PSF filtering from this analysis as it is just a particular case of the Gaussian filtering method. We perform the test on the VIS (CM) image and we change the main parameter value of each technique by $\pm 10\%, \pm 25\%, \pm 50\%$ and $\pm 75\%$ with respect to the value used for the *MSE* analysis (see Subsect. 4.4.1). The results are shown in Fig 4.4. We notice that most of the techniques tend to have similar performances when over-estimating the parameters, remaining relatively stable; on the contrary, under-estimating it significantly worsens the performance. However, all the algorithms have a lower dispersion in *MSE* compared to the Gaussian filtering (this is not evident in the plot because of the logarithmic y-axis scale, but we verified it numerically and we give the values in the upper panel of the plot), meaning that they are generally more stable against the variation of the parameters. In addition they yield a ~ 1 order of magnitude lower mean when the parameters are below the optimal value, and by ~ 2 order of magnitudes when they are above it.

4.4.5 Conservation of the FWHM

The optimal denoising approach should not significantly alter size and shape of the detected sources so to enable a meaningful scientific analysis. We thus tested the selected methods by measuring the FWHM of the detected sources with SExtractor, and comparing the measured values to the ones obtained on original, unfiltered images. We perform this test on the simulated VIS image described before, which is mainly populated by galaxies, and on a specific rendition of the simulated VIS image populated by stars distributed on a grid. The results are shown in Figs. 4.5- 4.6.

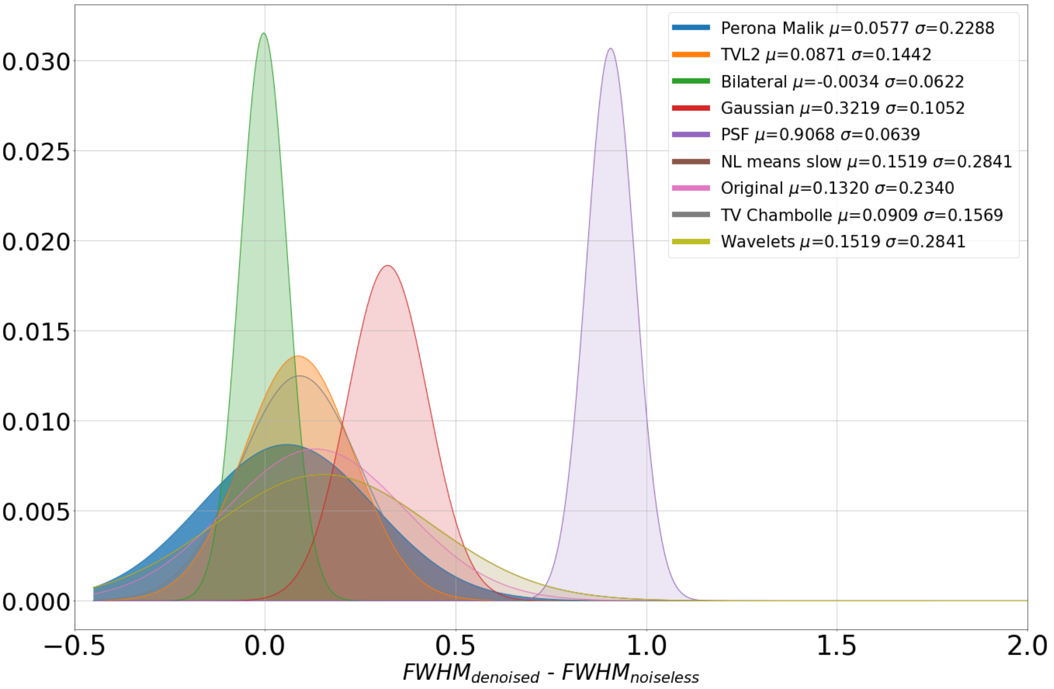

Figure 4.5. Step 4: FWHM conservation test on stars. On the x-axis we plot the $FWHM_{denoised} - FWHM_{noiseless}$, where $FWHM_{noiseless}$ is the FHWM of the objects measured on the Noiseless image. μ and σ are the mean and the standard deviation of the distribution of $FWHM_{denoised} - FWHM_{noiseless}$.

While for the stars in Fig. 4.5 the PSF filtering tends to smooth all the detected object as much as of $\sim$ 50% of the FWHM, most of the other algorithms, and in particular Bilateral, Perona-Malik, TVL2 and TV Chambolle, have a much lower impact (the FWHM is degraded by less than 20% of the original value). Similarly, for the galaxies in Fig. 4.6, the PSF filtering causes again a small offset, whereas all the other methods tend to better preserve the FWHM. From this test, we can conclude that all the tested algorithms preserve the shape of the sources better than the PSF filtering.

4.4.6 Completeness and Purity

In this test - perhaps the crucial one - we checked the quality of the catalogs of sources extracted from the denoised images. We analyze two quantities, both relevant to assess the performance of the detection process: namely, the *completeness* and the *purity* as defined below. We extract the catalogs running SExtractor in *dual* image mode using a denoised image as detection band and the original image

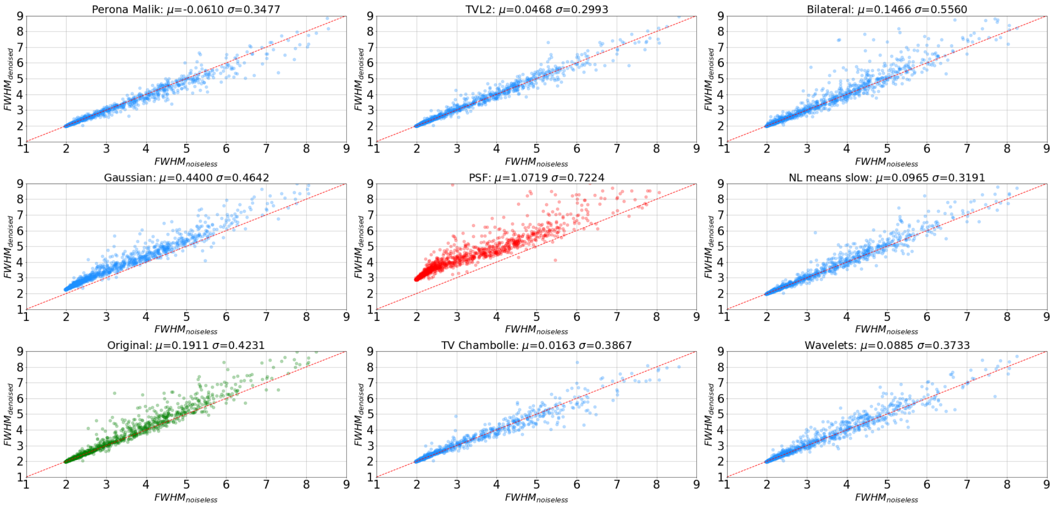

Figure 4.6. Step 4: FWHM conservation test on galaxies. On the x-axis we plot the FWHM of the objects measured on Noiseless image $FWHM_{noiseless}$, whereas on the y-axis we plot the FWHM measured on the Original image after the application of the denoising algorithms $FWHM_{denoised}$. μ and σ are the mean and the standard deviation of the distribution of $FWHM_{denoised} - FWHM_{noiseless}$.

as measurement band so to perform a cross-correlation between the extracted and the true catalogs of sources both in terms of position and flux.

We used the simulated VIS 5000x5000 pixels image, searching for the best SExtractor parameter configurations for every denoised image. We have thus tested a large number of possible combinations of the two parameters which control the detection, i.e. `DETECT_THRESH` (from a minimum value of 0.2 to a maximum of 6.0, with steps of 0.1) and `DETECT_MINAREA` (with values: 3,6,9,12,15,30), considering only the combinations for which the quantity $\texttt{DETECT\_THRESH} * \sqrt{\texttt{DETECT\_MINAREA}} > 1$, which provides a selection of objects with a global significance of at least 1-σ. The number of detection parameters combinations which fulfil this requirement is ~ 350. We point out that the best algorithms configurations used for this test and obtained by *MSE* minimization, do not differ significantly from the best configurations found in the VIS (CM) image (Sect. 4.3). We introduce some notations:

- $n_{detected}$ is the total number of detected objects, which includes both real and spurious detections indiscriminately
- $n_{simulated}$ is the number of simulated objects in the image
- $n_{spurious}$ is the number of spurious detections, as identified by the *spurious sources identification approach* described in the following.

The *spurious sources identification approach* that we define for this work is related to the SExtractor cross-correlation, when an association catalog is provided: we denote by $C_{R_{assoc}}$ the circle centered on the simulated object original position with radius R_{assoc}, which is the maximal distance allowed for the association made by SExtractor. We set it to 6 pixels (i.e. $3 \times FWHM$). Then, we tag an object as spurious if one of the following two conditions holds:

- is outside $C_{R_{assoc}}$
- (is inside $C_{R_{assoc}}$) AND ($|mag_{measured} - mag_{simulated}| > 1.0$) AND ($mag_{aperture} - mag_{simulated}| > 1.0$),

where $mag_{measured}$ is SExtractor `MAG_AUTO` (an estimation of the total magnitude of the source), $mag_{simulated}$ is the true magnitude of the simulated object, and $mag_{aperture}$ is SExtractor `MAG_APER` corresponding to the magnitude within a circular aperture with diameter of 6 pixels.

Finally, we can now define the two quantities:

$$completeness := \frac{n_{detected} - n_{spurious}}{n_{simulated}} \tag{4.4}$$

$$purity := 1 - \frac{n_{spurious}}{n_{simulated}} \tag{4.5}$$

where $purity = purity_{assoc}$, determined by the association approach defined above. We measure completeness and purity in 0.2 magnitudes bins. In Fig. 4.7 we plot the magnitudes at which the completeness drops below 50% against the one at which the purity drops below 90%. Every symbol corresponds to a different denoising technique, and repetitions of the same symbols corresponds to different combination of the detection parameters for the same algorithms. For readability, only the best 5 combinations per algorithm are shown.

We note that all the methods are improving the detection, the best performance are reached by TVL2, Perona-Malik, TV Chambolle and Bilateral. Indeed, they reach the completeness threshold 0.6 mag deeper, and the purity threshold 0.8 magnitude deeper than the non-denoised run. Moreover, they improve the detection compared to the PSF smoothing, reaching 0.2 magnitudes deeper in both completeness and purity.

It is tempting to consider the *MSE* and *SSIM* measured on the VIS images used for the completeness and purity analysis, searching for a possible correlation between the parameters. In Fig. 4.8 the plots are produced using the results shown in Fig. 4.7. We find no or weak correlation between *MSE* (*SSIM*) and purity, whereas a stronger correlation exists between *MSE* (*SSIM*) and completeness.

We show the snapshots of a sample of objects detected by the different methods in the VIS image in Appendix C. These snapshots give a visual match of objects detected in the denoised images. We only show the best performing algorithms results compared to the Original, PSF-filtered and the Noiseless images. For VIS the algorithms are: PM, TVL2, Bilateral and TV Chambolle. We point out that these objects are undetected with the best detection parameters configuration in the image filtered with the PSF.

4.4.7 Conservation of the flux

In this final test, we compare the total fluxes (SExtractor `MAG_AUTO`) measured on the simulated denoised images, for objects with magnitude within 19 and 23, to the true input fluxes. The results are shown in Figs. 4.9-4.10.

The standard deviation of the difference between measured and true magnitudes is ~0.13 for PSF-filtered images. All denoising methods show similar performances

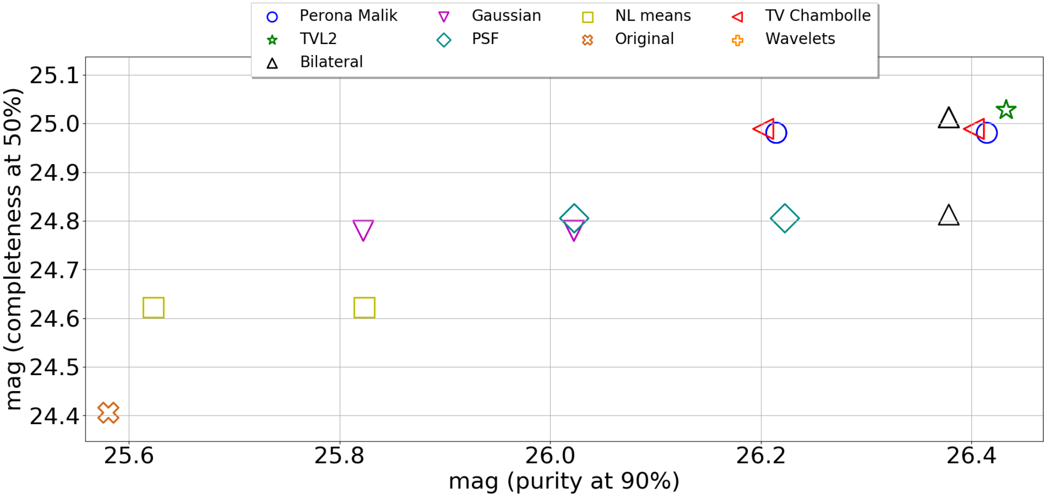

Figure 4.7. Step 5: Completeness and purity test. We extracted catalogs on the VIS simulated image processed with the denoising algorithms, using different configurations of SExtractor. We plot the magnitude at which the completeness drops below 50% against the magnitude at which the purity drops below 90%. Each symbol corresponds to a different denoising method, which can be present multiple times in the plot due to different combinations of detection parameters. The positions of the symbols are slightly randomized to improve readability.

with the exception of the Wavelet one (σ_{mag}=0.35). We conclude that denoising algorithms preserve the overall calibration of the input images and they enable a photometric accuracy comparable to the one usually achieved on images filtered with the PSF.

4.5 Test on real images

After having analyzed the performance of denoising techniques on a series of simulated images, we test the algorithms on real images, using the HST $H160$ observations of the GOODS-South Field and a crop of the HAWK-I survey.

4.5.1 Space telescope images

We use two images of the area of the Hubble Ultra Deep Field: one at the full depth released with the official CANDELS mosaics that includes all WFC3 observations of that region (HUDF09, reaching H160=29.74 at SNR=5), the second, shallower one at the depth obtained with WFC3 observations of the CANDELS Treasury Program alone (GSDEEP, H160=28.16 at SNR=5) [90, 66]. We will use the former, deeper image as "true sky", against which we will compare the performance of denosing techniques on the shallower image. Using an analysis similar to that in Sect. 4.4.6, we take as reference catalog the one obtained running SExtractor on HUDF09 with conservative detection parameters. The goal is again to check completeness and purity. We use again an association radius R_{assoc} of $3 \times FWHM$, now corresponding to 7.5 pixels. We identify an object as spurious using the same criteria used in Sect. 4.4.6 with a $mag_{aperture}$ within a circular aperture with diameter

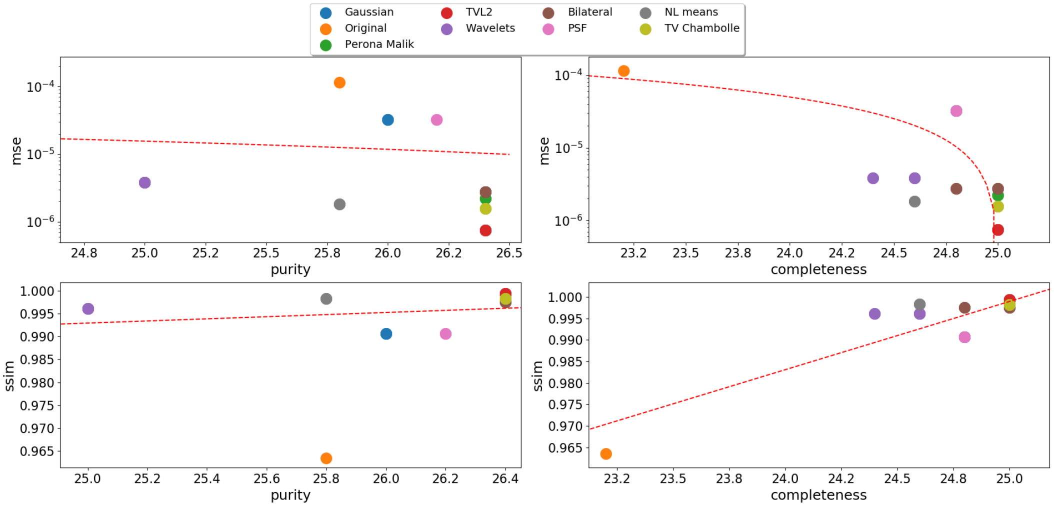

Figure 4.8. Correlation between *MSE* or *SSIM* and purity or completeness. On the x-axis we plot the magnitudes at which completeness (purity) reaches 50% (90%), whereas on the y-axis we plot the parameters *MSE* or *SSIM*. Dashed lines are the linear best-fitting.

of 7.5 pixels. The resulting plot, visible in Fig. 4.11, is similar to the one obtained on the simulated image (Fig. 4.7). Clearly, TVL2 outperforms all the other algorithms outperforming the PSF by 0.2 mag in completeness and purity (or alternatively 0.4 magnitudes more complete and 0.2 magnitudes less pure); Bilateral performs better than the PSF filtering, by an amount of 0.2 magnitudes in completeness. Perona-Malik can provide a 0.2 mag more complete and 0.2 less pure alternative to the PSF filtering.

Like in Sect. 4.7, we show the snapshots of a sample of objects detected by the different methods in the GSDEEP image in Appendix D. For GSDEEP the best algorithms are: PM, TVL2, Bilateral and NL means. We point out that these objects are undetected with the best detection parameters configuration in the image filtered with the PSF.

4.5.2 Ground-based images

We repeat the same tests described above on two Ks-band observations of the Goods-South field acquired with the HAWK-I imager at the VLT: a shallower observation of the field presented in [31] and the ~2 magnitude deeper observation released by the HUGS Survey [54, see Table 4.1]. As above, we use the deepest image as “true sky” and we apply the algorithms to reduce the noise on the shallow image. We use again the association radius R_{assoc} of $3 \times FWHM$ corresponding to 11.25 pixels, with the relative $mag_{aperture}$ (11.25 pixels diameter), identifying an object as spurious using again the same criteria already used in Sect. 4.4.6 and Sect. 4.5.1. The resulting plot (see Fig. 4.12) shows that, while these algorithms improve the image detection compared to not making denoising at all (same result obtained in Sect. 4.4.6 and Sect. 4.5.1) they do not provide significant improvements compared to the PSF. Indeed, only PM creates a catalog of 0.2 magnitudes more pure at the same completeness. These results are in agreement with those in Sect. 4.4.3, where

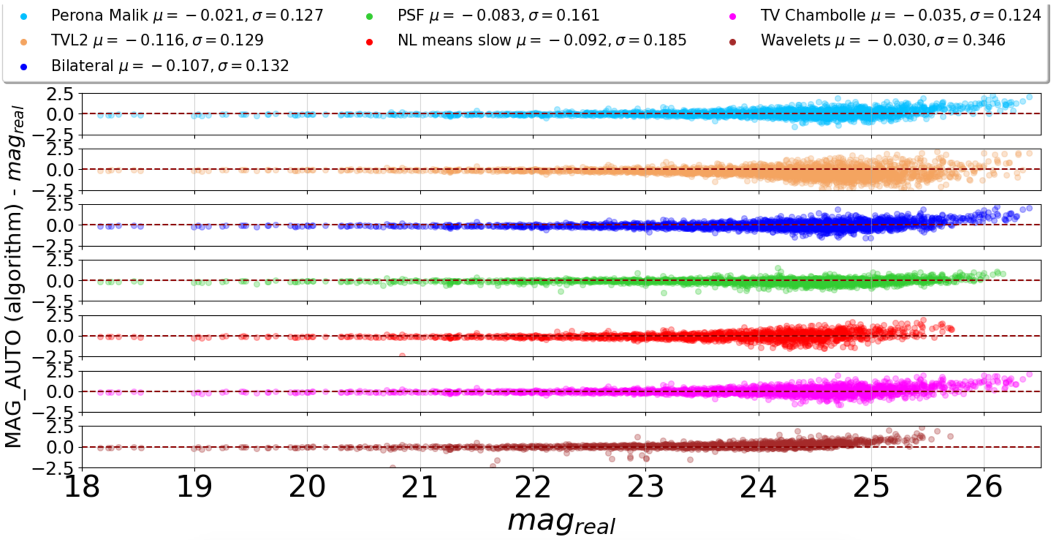

Figure 4.9. Flux conservation distribution for objects with magnitude within 19 and 23. On the x-axis the real objects magnitude mag_{real}, On the y-axis, the difference between the magnitude measured `MAG_AUTO` and mag_{real}. Only the detected objects within the purity and completeness thresholds (Sect. 4.4.6) are considered. μ and σ are the distribution mean and the standard deviation values.

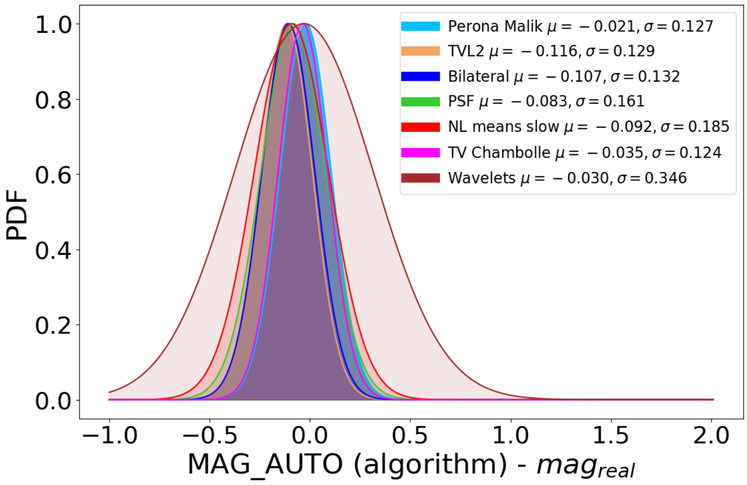

Figure 4.10. Flux conservation distribution for objects with magnitude within 19 and 23. On the x-axis, the difference between the magnitude measured `MAG_AUTO` and the real objects magnitude from the catalog (mag_{real}). On the y-axis the `MAG_AUTO` - mag_{real} probability distribution function. Only the detected objects within the purity and completeness thresholds (Sect. 4.4.6) are considered. μ and σ are the distribution mean and the standard deviation values.

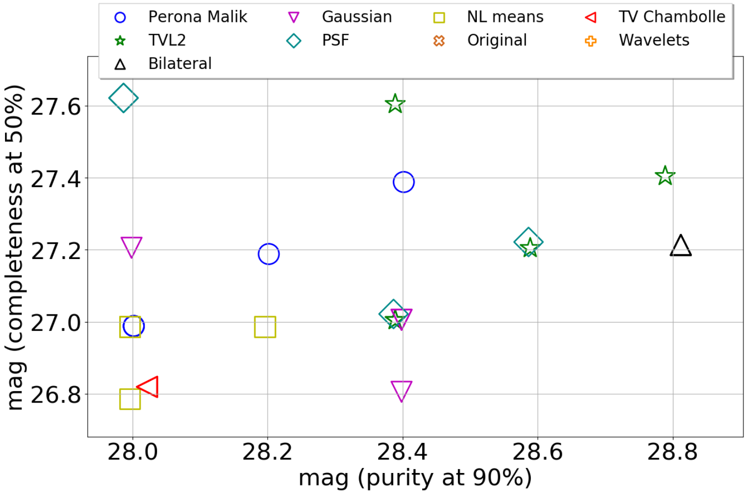

Figure 4.11. Space telescope real Images Completeness & Purity (GSDEEP and HUDF09). On the x-axis the magnitude at which the purity drops below 90%, on the y-axis the magnitude at which the completeness drop below 50%. Each symbol is referred to a different denoising method, which can be present multiple times in the plot due to different combinations of detection parameters, see text for details. The points are randomized around their actual position within a circle of radius 0.025 magnitudes to improve visibility.

we noticed that all these methods give the best with high resolution images (see Fig. B2), such as VIS and H160. Indeed the lower resolution of the ground-based images impacts the algorithms performances. In the same way the methods, and mainly the PSF, perform better for lower SNR images (e.g. HAWK-I compared to VIS and H160), as shown in Figs. B5-B6, resulting in less significant improvements from the methods compared to the PSF.

4.6 Summary and Conclusions

The goal of this work is to make an extensive comparison between a number of denoising algorithms, aimed at identifying the best choice to improve the detection of faint objects in astronomical extragalactic images (e.g. considering the typical cases of HST and Euclid). To this purpose we performed a large set of tests, on simulated images. We also tested the methods on real images: from space and ground-based, collecting really interesting hints on many situations.

We chose to test the smallest set of essential denoising algorithms, leaving a more complete comparison for future works. In particular, we point out that ATVD-TVL2, Bilateral, Perona-Malik and TV Chambolle are the most interesting to use among all, as they provide good performances in the different tests proposed. Even if most of these methods are quite unusual for the astronomical community, they are very well-known in many other fields. They are known to outperform a straightforward PSF/Gaussian filtering, which is the standard choice in astronomy. We therefore considered these techniques as the reference ones, against which we tested all the other methods.

As a first test, we considered the two parameters *MSE* and *SSIM* (defined in

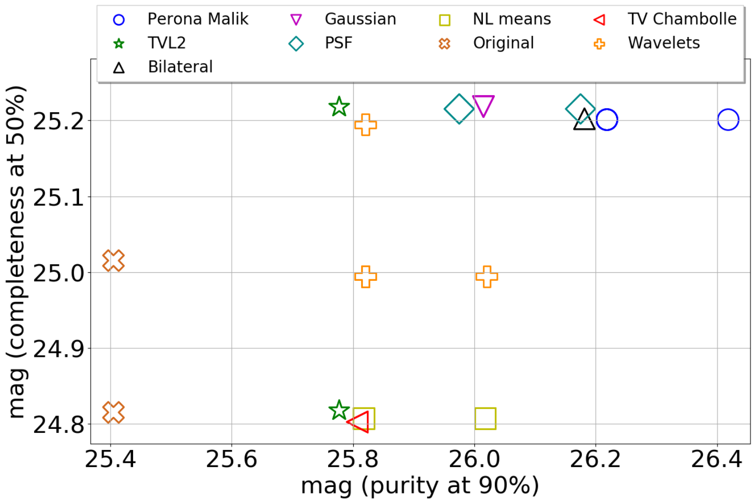

Figure 4.12. Ground-based real Images Completeness & Purity (HAWK-I and HAWK-I UDS). On the x-axis the magnitude at which the purity drops below 90%, on the y-axis the magnitude at which the completeness drop below 50%. Each symbol is referred to a different denoising method, which can be present multiple times in the plot due to different combinations of detection parameters, see text for details. The positions of the symbols are slightly randomized to improve readability.

Sect. 4.4) and checked which methods yield the best performances with respect to them. We compared their performances again through *MSE* and *SSIM* in relation to depth, resolution and type of image. We tested the algorithms ability to preserve the FWHM, in order to understand if they can preserve the shape of the objects, useful in case photometric measurements on the denoised image are needed. We tested their stability using the *MSE*, varying the ideal parameter of a fixed percentage, with the goal of having a hint of their reliability, in case the best parameter is chosen wrongly. We have also tested possible detection improvements through two parameters *completeness* and *purity* which are used to measure the fraction of real detections on the total number of objects in the image. Finally, we applied these methods on real images (CANDELS-GS-deep and a crop of HAWK-I). We summarize below the key points of the analysis performed:

- From *MSE* and *SSIM* we noticed that 8 algorithms are always on top of rankings, especially for VIS and H160 images, which are of main interest for the detection in future surveys. These algorithms are: PM, TVL2, Gaussian, PSF, NL means slow, Bilateral, TV Chambolle and Wavelets

- From the PSF and Depth variation test we noticed that all the methods are performing better with narrower FWHM. Whereas all the methods perform better with the noise level increasing (in terms of Gaussian noise standard deviation) but their improvements is more significant, compared to the PSF, with high SNR images

- From the FWHM conservation test, we noticed that most of the algorithms tend to not smooth the image, in terms of FWHM increments. On the contrary, the PSF smoothing provides an offset in the FWHM measurement, for both, stars-only image and galaxies-image

- From the completeness and purity test we found a small number of algorithms which provide 0.2 magnitudes more pure and complete catalogs than the PSF filtering, these are TVL2, Perona-Malik, TV Chambolle and Bilateral
- From Flux conservation test we found that all algorithms, with the exception of Wavelets, have similar performance to the PSF filtering, preserving the overall calibration of the input images
- From real image test (H160) we found that TVL2 outperforms all the other algorithms, and it is the only one that performs better than the PSF of 0.2 magnitudes in completeness and purity, while Bilateral produces only a 0.2 more pure catalog
- From real image test (HAWK-I) we found that only Perona-Malik outperforms the PSF filtering, by 0.2 magnitudes in purity, the other methods performs worse/similarly to the PSF.

The results we obtained demonstrate that denoising algorithms should be considered valuable tools for optical and near-IR extragalactic surveys, as they clearly improve the detection of faint objects. In particular, structure-texture image decomposition, Total Variation denoising, Perona-Malik and Bilateral filtering are of particular interest. While further specific tests are needed to define for each survey the optimal approach, and its parameter set, among the above mentioned ones, our investigation on a small but reasoned reference set of simulated and real extragalactic images shows that the scientific return of ongoing and future surveys can be significantly enhanced by the adoption of these denoising methods in place of standard filtering approaches.

Chapter 5

Deblending with Machine Learning Techniques on Euclid-VIS Images

In this chapter, we introduce the blending of astronomical objects typically found in extragalactic surveys and its impact on scientific measures. The aim is to introduce various deblending approaches, discussing their performances, in handling blending issues. All the techniques shown in this chapter can be applied to several situations, while alternative solutions are adopted [83, 113, 29] to optimally work on multi-bands surveys.

We discuss three techniques and their application on a Euclid-like environment [97], considering a single high-resolution VIS band with PSF-FWHM of 0.2 arcsec, pixel scale of 0.1 arcsec, and limiting magnitude of 25.25 at SNR=5. The datasets consider a straightforward blending situation with only 2 objects blended. This decision is justified by examination of the HST catalogs, from which $\sim 75\%$ of the blendings is composed of only 2 objects [45]. The first technique is the already famous code, SExtractor [19]. Its deblending strategy is based on a multi-thresholds approach, moreover it is used as a baseline. The second technique is based on an algorithm called DENCLUE (defined in Sect. 3.2.2) implemented in a Python package called ASTErIsM [162], which is currently under testing (Tramacere et al. in prep.). The third technique is actually a set of two Convolutional Neural Networks `blend2flux` and `blend2mask2flux` [26].

After the introduction the chapter is divided in two parts, each introducing the different methods. The comparison is performed using SExtractor as common baseline. The datasets used differ for the two works along with the comparison metrics. These two parts are the result of two scientific papers. The first, still under preparation, discusses about theASTErIsM performances, while the second, already accepted by *Monthly Notices of the Royal Astronomical Society* [26], discusses about `blend2flux` and `blend2mask2flux`.

5.1 The Blending issue in extragalactic images

Future extragalactic surveys (some of them seen in Chapt. 2) will push the limits in size and depth to a new level, diving in the world of big data. With such an amount of data, photometry and weak lensing measurements will reach unparalleled high precision. The high precision required can only be achieved, limiting as far as possible the major sources of systematic errors. One of these is the objects blending. Objects blending (or simply blending) is the result of overlapping objects due to the projection of the 3D sky on a 2D image. In a crowded field, even objects with a large relative distance can be overlapping from our telescopes perspective, forming a single blended profile. Blending is a function of the surface number density of objects in the field and the telescope PSF (in terms of FWHM), which means that objects with a large limiting magnitude and a broad PSF are mostly affected. It is estimated that for an ultra-deep optical ground-based survey, such as LSST $\sim 45\% - 55\%$ of the objects will be blended, producing systematics in several scientific measures [44]. On the other hand, surveys from space, with a narrower PSF, are advantaged, being less affected by blending issues. Nevertheless, high precision required for these surveys scientific goals still implies that the blending issue must be handled.

A straightforward classification of blending is defined in [44]. The definition assumes that the objects follow the PSF profile, parameterized by a Gaussian:

- **Catastrophic blends**: When 2 or more objects are blended such that the relative distance between their peaks is within the sum of the 1σ distances, and are normally considered a single object by detection algorithms
- **Conspicuous blends**: When 2 or more objects are blended such that the relative distance between their peaks is within the sum of the 2σ distances, and are detected as blended
- **Innocuous blends**: When 2 or more objects overlap by a negligible factor, being the distance between their peaks larger than the sum of the 2σ distances.

Before introducing the methods a brief description about the impacts of blending on the scientific measurements is shown.

5.1.1 Blending impact on Photo-z

Blending impacts photo-z measurements, providing false flux information due to contamination which imply a redshift wrongly estimated [44]. To define the contamination amount, it is needed a quantity called pixel purity [41]:

$$P_i = \frac{signal\ of\ object\ i}{total\ signal\ of\ blended\ objects} \tag{5.1}$$

P_i is the pixel purity of the object i, which is blended with other n objects. From P_i, purity per pixel P_{ij} is defined such that $\sum_{j=1}^{N} P_{ij} = 1$, N is the total number of pixels of the blended objects. Thus, contamination for the object i is [44]:

$$C_i = 1 - \frac{\sum_{j=1}^{N} S_j P_{ij}}{\sum_{j=1}^{N} S_j} \tag{5.2}$$

S_j is the pixel value (contributing to the flux). Considering a case where in a blending, the first object is very bright in respect to the other objects, than C_i is close to 0, whereas for the other fainter objects C_i is close to 1. Flux contamination is not the issue to concern if the blended objects are at different redshifts the photo-z estimation is less reliable [44]. The redshift difference between two blended objects is defined as:

$$D = 1 - \frac{\int_0^\infty \frac{1}{\sigma_1\sqrt{2\pi}} e^{\frac{-(z-z_1)^2}{2\sigma_1^2}} \frac{1}{\sigma_2\sqrt{2\pi}} e^{\frac{-(z-z_2)^2}{2\sigma_2^2}} dz}{(\frac{1}{\sigma_{min}\sqrt{2\pi}})^2 \int_0^\infty e^{\frac{-(z-z_{min})^2}{2\sigma_{min}^2}} dz} \tag{5.3}$$

D is calculated assuming that the photo-z uncertainty of the two objects is represented by a Gaussian $G(z, \sigma)$, with z_1 and z_2 the true redshift values and σ_1, σ_2 the standard deviations. The denominator normalizes the function, thus if $G_1 = G(z_1, \sigma_1) = G(z_2, \sigma_2) = G_2$ then $D = 0$, while larger is the separation, closer D is to 1. Blending impacts the photo-z measures as a function of C and D, $F(C, D)$. A possible function is $\sqrt{C \cdot D}$ as shown in [44].

5.1.2 Blending impact on Supernovae science

Blending impacts SN science as a function of flux contamination C (Eq. 5.2). SN science is based on the photo-z estimation from the SN itself, or from the host galaxy (before and after the SN explosion). As photo-z is directly affected by C, photo-z estimation of the host galaxy could not always be reliable. A possible, new alternative is the broad-band photometry of the SN directly. The method is less sensitive to blending but is still affected, as the SN extinction is increased by it, producing an uncertainty of a few percent level [44].

5.1.3 Blending impact on Weak Lensing science

Blending impacts WL science, directly influencing shear, and photo-z measures. Catastrophic blends imply that two objects at the same redshift have the same shear, but if the two objects have different SEDs, the combined SED could provide a wrong photo-z estimation. In case the two objects are at different redshift, it is difficult to interpret the shear estimate. Excluding blending in galaxy clusters, the majority of catastrophic blends takes place at different redshift [109]. The approach of excluding galaxies form WL measures does not help with catastrophic blends. Furthermore, it gives a scale-dependent bias in shear-shear correlations, since blends are prevalent in high-density regions [109][71].

5.2 Deblending with Unsupervised Learning

In this section, deblending with an unsupervised learning (topological clustering) technique is investigated. The content of this section is part of two different works currently under preparation. The first concerns the datasets creation and is summarized in Sect. 5.2.1. The second is about the application of a topological clustering technique compared to a multi-threshold approach, the metrics used, the preliminary results achieved. This second work is summarized in Sect. 5.2.2-5.2.3-5.2.4.

5.2.1 The dataset

The datasets used for the analysis are cubes of FITS images. Each cube has a different number of images in it, depending on the samples described below. In each image, according to the dataset, there are 1 or 2 objects, see Table 5.1. The datasets have been generated including all the couples of objects in an simulated extragalactic image, within few FWHM (details in table 5.1) to be involved in deblending issues. The general idea is to create two kinds of dataset, one is generated using Skymaker [18], thus only with simulated objects, the other is generated using real CANDELS-GS cutouts. First of all, we selected the objects in a simulated image Euclid VIS-like created with EGG [150]. We select all the couples with distance between the objects d within a range: $[d_{min}, d_{max}]$, using a KD-tree algorithm described in [110] and implemented using the Python class `scipy.spatial.cKDTree`[1] [82]. After the selection in distance, we proceeded selecting all the objects with a magnitude within the range $[mag_{min}, mag_{max}]$. The parameters used for the datesets are reported in the Table 5.1.

Dataset	d_{min}	d_{max}	mag_{min}	mag_{max}	N_{obj}
couples	0	10	19	26.0	2
big	0	50	19	23.0	2
single	0	10	19	26.0	1

Table 5.1. Parameters for objects selection in the datasets, “couples” represents a sample of common and typical couples of objects; ”big” represents a sample of bright and big couples of objects; “single” represents all the objects in couples and big, one single object per image. The distances are expressed in pixels

Skymaker Cubes

As a first step we created the cubes using Skymaker, selecting the objects as described above. The cubes are also coupled with true maps cubes and an rms map. The first contains the objects segmentation stored in the different images that compose the cubes, the second is a single image, where every pixel has a the same value, corresponding to the Euclid VIS background standard deviation σ_{VIS} (Skymaker approximates the background to have a Gaussian distribution). In the true maps cubes, the segmentation has been produced with a simple threshold method. The continuous pixels in the image with values α times the rms, to the corresponding pixels in the true map is given the same integer, all the other pixels are set to 0 (we chose $\alpha = 0.8$ as increasing this value produces small segmentation maps, while a smaller value produces fake detections due to the background fluctuations). The overlapping pixels between the two objects is indicated by the value -1. In these cubes we can decide to randomize the objects inclination angles, deciding the number of possible combinations through the parameter $N_{repetitions}$. For the

[1] http://www.scipy.org/

2-objects per image datasets, we require that the objects sizes are smaller than the distance between them. To avoid impossible cases to deblend, both the following conditions must be fulfilled:

- $max(bulge_{radius,1}, disk_{radius,1}) < d_{max}$
- $max(bulge_{radius,2}, disk_{radius,2}) < d_{max}$

where $bulge_{radius,1}$ is the bulge radius of the first object, where $bulge_{radius,2}$ is the bulge radius of the second object, where $disk_{radius,1}$ is the disk radius of the first object, where $disk_{radius,2}$ is the bulge radius of the second object. For every object in the dataset, the essential features extracted from the EGG catalog are stored and listed below:

- `type`: Defines if the object is a star or a galaxy, with values 100 and 200 respectively
- `x`: New X coordinate of the object in the image
- `y`: New Y coordinate of the object in the image
- `mvis`: VIS magnitude of the object
- `BT`: Bulge on total ratio of the object
- `b_R`: Bulge radius in arcsec
- `b_RATIO`: Bulge inclination angle, normalized between 0 and 1, where 0 means edge-on while 1 means face-on
- `PA`: Position angle of the object bulge and radius, perpendicular to the line of sight
- `d_R`: Disk radius in arcsec
- `d_RATIO`: Disk inclination angle, normalized between 0 and 1, where 0 means edge-on while 1 means face-on
- `ID`: Object ID in the EGG catalog
- `z`: Object spectroscopic redshift
- `dx`: Image dimension along the x-axis containing the object
- `dy`: Image dimension along the y-axis containing the object

In the 2-objects datasets, the 2 objects are put in the middle of the image at the same Y coordinate and at a distance d between them on the X-axis. While for the 1-object dataset, the object is simply in the middle of the image.

For each object in the datasets, the surface brightness is evaluated considering the brightest 5, 10 and 50 pixels of the object.

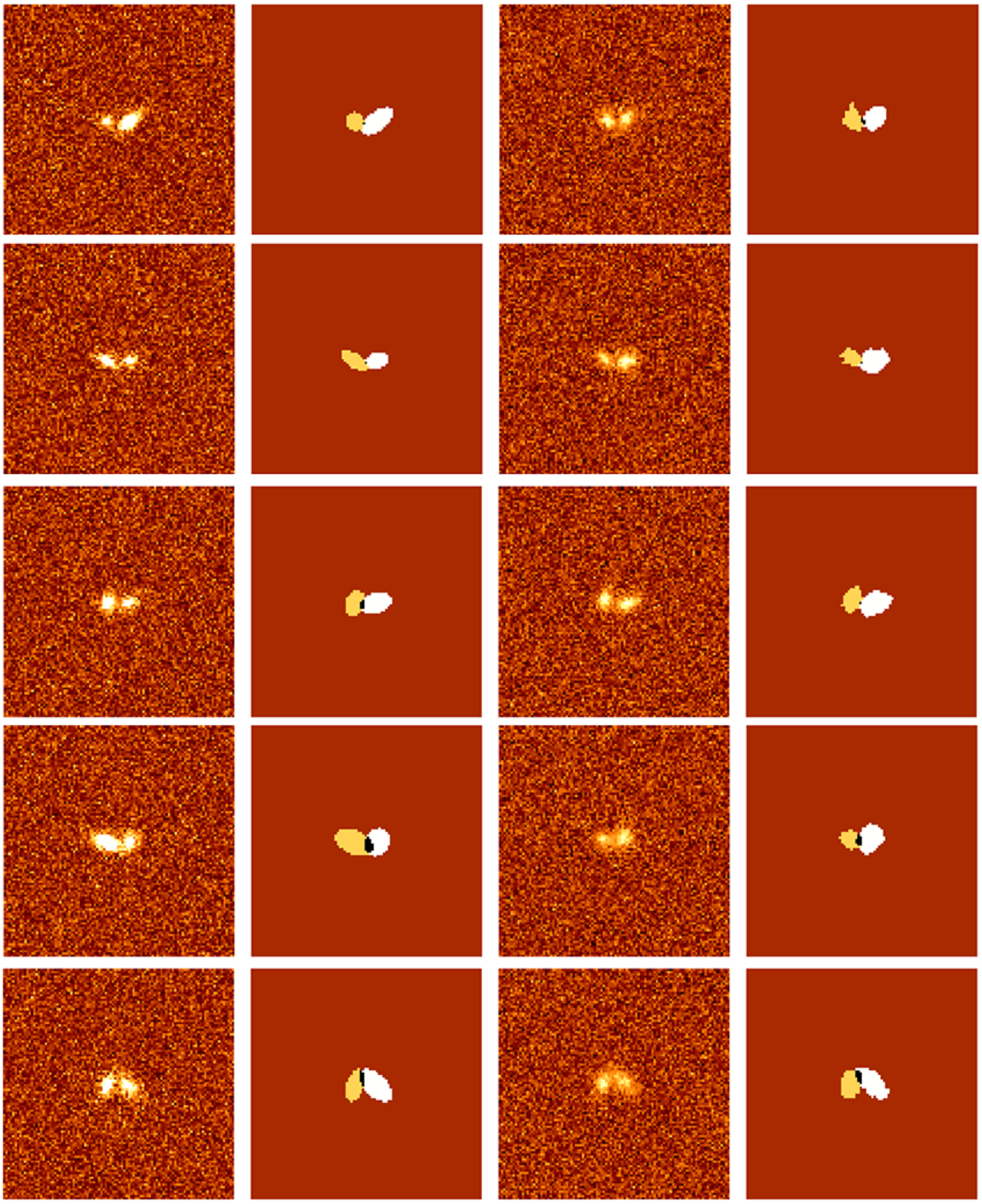

Figure 5.1. A selection of the couples datasets. The first column shows two blended simulated objects produced with SKYMAKER+EGG, the second column shows the related true map (SKYMAKER+EGG), the third column shows the the two associated blended CANDELS cutouts, the fourth column shows the related true map (CANDELS)

CANDELS-GS cutouts Cubes

Saving the SKYMAKER cubes for later we now discuss the CANDELS-GS cutout selection. We selected cutouts cross-correlating different catalogs, classified by both visual [85] and neural networks classifiers [78]. The GALFIT [169] parameters that we use are listed below:

- `ID`: The CANDELS object ID;
- `gfit_mag_h`: The absolute total magnitude from best fitting of Sersic model
- `gfit_f_h`: Quality flag of the fitting.
- `gfit_sma_h`: Semi-major axis of the ellipse containing half of the total light in the best fitting Sersic model
- `gfit_n_h`: The Sersic index best fitting Sersic model for the object
- `redshift`: The object spectroscopic redshift

while the SExtractor [19] parameters from the CANDELS-GS catalog [69] are listed below:

- `MAG_AUTO`: Kron-like elliptical aperture magnitude
- `FLUX_AUTO`: Flux within a Kron-like elliptical aperture
- `FLUXERR_AUTO`: RMS error for `FLUX_AUTO`
- `ELONGATION`: Major axis/Minor axis
- `THETA_IMAGE`: Position angle perpendicular to the line of sight

We selected cutouts with `MAG_AUTO`< 24.5 looking for a trade-off between diversity and fidelity of the sample. After the cutout selection step, we take back the Skymaker datasets, assigning a cutout to each object in the cubes. The assignment algorithms is schematized below:

- Taking the generic Skymaker object for reference, if `BT`> 0.7 we restrict the assignment to objects in the CANDELS cutouts with $3 <$ `gfit_n_h` < 8, otherwise if `BT`< 0.7 we assign cutouts with $0.5 <$ `gfit_n_h` < 2
- We use the visual classifiers f_{disk}, f_{sph} and f_{irr} from [85] to assign a morphological, class defined in [78], to the cutouts, as described below
- The morphological classes used are schematized below:
  - Disk irregular: $f_{disk} > 2/3$ AND $f_{sph} < 2/3$ AND $f_{irr} > 1/10$
  - Disk: $f_{disk} > 2/3$ AND $f_{sph} < 2/3$ AND $f_{irr} < 1/10$
  - Spheroids: $f_{disk} < 2/3$ AND $f_{sph} > 2/3$ AND $f_{irr} < 1/10$

  We also use the same six spectroscopic redshift bins $[0, 0.1, 0.6, 1.1, 1.6, 2.1, 2.6, 3+]$ and the morphological distributions defined per bin
- For Skymaker objects with `BT`< 0.7 we use the `redshift` parameter extracted from the EGG catalog to make a random choice, weighted using the morphological distributions in the proper bin as probability functions, assigning to the object a cutout with the morphology randomly extracted, preferring cutouts with similar surface brightness, dimension, magnitude and elongation

- For SKYMAKER objects with `BT`> 0.7, we automatically assign a cutout belonging to the Spheroids class, again preferring similar surface brightness, dimension, magnitude and elongation.

After that to each object is assigned a cutout, we proceed to create the CANDELS cubes, mimicking the objects positions, magnitudes, dimensions, angles etc. We add the Gaussian noise (in the same way of SKYMAKER does), with:

- $\sigma = \sigma_{VIS}$ to every pixel which doesn't belong to the objects
- $\sigma = \sqrt{\sigma^2_{VIS} - \sigma^2_{CANDELS}}$ to every pixel which belongs to the objects but doesn't belong to the overlapping area
- $\sigma = \sqrt{\sigma^2_{VIS} - \sigma^2_{CANDELS,1} - \sigma^2_{CANDELS,2}}$ to every pixel in the object 1 and 2 overlapping area,

where $\sigma^2_{CANDELS}$ is the background standard deviation measured in the cutout proximity and calculated as the ratio between the cutout flux and its signal-to-noise ratio

In the CANDELS cubes, every image which reproduces the SKYMAKER image is checked, in order to discard reproduced couples which totally overlap or those that don't have a true map over the threshold, meaning that the objects are too faint. Indeed the CANDELS cutouts and the background are rescaled, to reproduce a VIS-like image, but it is possible that, extremely faint (or close) SKYMAKER pairs could not be ideally reproduced (the cutouts can indeed differ from the mathematical models used by SKYMAKER). With the aim of having a one-to-one relationship between the datasets, the images discarded in the CANDELS cubes are also discarded in the SKYMAKER cubes. In the end a dataframe `sky_to_CANDELS.pkl` containing all the information on the related dataset is produced.

5.2.2 Methods

ASTErIsM

ASTErIsM (AStronomical Tools for clustering-based dEtectIon and Morphometry) [162] is a Python package, performing different tasks, among which deblending, performed with DENCLUE (see Sect. 3.2.2). ASTErIsM is designed to work with CCD astronomical images, using a modified versions of the above-cited algorithms to work with such datasets. The modifications to the algorithms are summarized below.

DENCLUE is modified to work with a CCD matrix. As a first step the density estimation is substituted by a kernel convolution. The resulting image after this step is estimated by:

$$f(p_j) \propto \sum_{i=1}^{n} G(\frac{q_j - q_i}{h}) I(p_i) \tag{5.4}$$

where p_j is the j-th pixel with coordinate q_j, G is the non-negative symmetric kernel function centred in q_j, h is the bandwidth of G and $I(p_i)$ is the flux of the pixel p_i.

When a Gaussian is used as kernel, h represents the standard deviation. The fast hill climbing used to associate density attractors to the data points is:

$$q_{t+1} = \frac{\sum_{i=1}^{n} G(\frac{q_t - q_i}{h}) q_i I(p_i)}{\sum_{i=1}^{n} G(\frac{q_t - q_i}{h}) I(p_i)} \tag{5.5}$$

where q_{t+1} are the updated coordinates of the generic point at the $t+1$ iteration. The convergence is reached when: $||q_t - q_{t+1}|| < \epsilon_d$, where ϵ_d is a threshold that can be set. At convergence q_{t+1} defines the position of the *density attractor* p_j^* for the point p_j. With DENCLUE the attractors are evaluated and clustered and after that *source clusters* are eventually deblended:

- Being S a *source cluster*, each point $p_j \in S$ corresponds to the pixel with coordinates q_j and flux $I(p_j)$
- For each p_j a *density attractor* p_j^* is associated
- All the attractors are then clustered using a labeling algorithm, implemented in SCIKIT-IMAGE[2], producing a list of *clusters of attractors*
- The pixels whose *density attractors* belong to the same cluster compose a new *sub-cluster*
- The new sub-clusters are then validated considering the number of pixels (between a minimum and maximum value) and the ratio of the sub-cluster flux compared to the parent cluster

In ASTERISM, a `scale-finder` algorithm sets different values of h as a fraction of the clusters radii, from now on, we talk about h_frac. To identify the optimal value of h_frac, a recursive application of the Laplacian of Gaussian (LoG) *blob detection* algorithm, implemented in SCIKIT-IMAGE [2], is used. The tested values of h_frac are within h_frac_min and h_frac_max. The two parameter are used to find the optimal configuration to handle the objects blending.

For further details about the DENCLUE algorithm see [162]. The strategy promoted to find the best configuration of ASTERISM is described below.

As a first step, the segmentation maps are produced using SEXTRACTOR with deblending turned off, using them as input for both SEXTRACTOR and ASTERISM, simulating the input to deblending of the Euclid pipeline. The segmentation images compose a cube, that is added to the dataset. This step is fundamental for the kind of pixel-wise analysis that we perform, which requires the same segmentation maps. SEXTRACTOR detection parameters `DETECT_THRESH` and `DETECT_MINAREA` are set to 1.2 and 10 respectively. These values provide a fair trade-off between segmentation dimension and significance level. We used the rms map mentioned in Sect. 5.2.1. We also pre-filtered the images through a Gaussian filter with Full Width at Half Maximum (FWHM) of 2 pixels (approximating the VIS Point Spread Function (PSF)) and set the cleaning with `CLEAN_PARAM`=1.0 by default.

Once the segmentation is provided to DENCLUE, the algorithm performs deblending as described above. Here the value of the parameter h assumes an

[2] https://scikit-image.org/

important role in finding the sub-clusters. We explored different combinations of the parameters: h_frac_min and h_frac_max, to find the best configuration which optimizes the deblending. We sampled both values from 0.05 to 0.5 with a step of 0.05. We slightly pre-filter the images with a 0.5 pixels-wide PSF. The results shown in Sect. 5.2.4 refers to a limited number of ASTErIsM configurations, which are not included for simplicity purposes, concerning only combinations of h_frac_min and h_frac_max. We refer to them as *deblending parameters* of ASTErIsM.

Baseline: SExtractor

In this section we describe the approach used for the detection (and deblending) adopted with SExtractor [19]. We tested different configurations to find out the best parameters combination that enhances the deblending performances. As already described for ASTErIsM, we provide the input segmentation (using the same detection+filtering+cleaning parameters) with deblending turned-off to SExtractor. After that the deblending is performed.

We explored 80 combinations of the parameters which handle the deblending: `DEBLEND_MINCONT` and `DEBLEND_NTHRESH`. We sampled `DEBLEND_MINCONT` from 1×10^{-4} to 1×10^{-2} with a step of 1×10^{-3} and `DEBLEND_NTHRESH` from 8 to 64 with step equal to 8. We pre-filtered the images with the 2 pixels-wide PSF. We also wanted to test the impact of cleaning in the deblending performances, turning it off and using the values for `CLEAN_PARAM` equal to 0.1 and 10. The results of the various configurations are shown in Sect. 5.2.4, and concern only combinations of `DEBLEND_MINCONT` and `DEBLEND_NTHRESH`, and represent the *deblending parameters* of SExtractor.

5.2.3 Comparison Metrics

The aim is to compare the results produced by ASTErIsM and SExtractor in their optimal configurations on our datasets, defined in Sect. 5.2.1. We remind that the generic dataset is composed of a cube of images, a cube the related true maps, a cube of related segmentation maps (without deblending), and the related rms maps. Considering a couple of simulated objects in one of our datasets, the objects are represented by an image (in the optical band Euclid VIS), a true map, and segmentation map. After deblending is performed by the algorithms the deblended segmentation map is used to evaluate the performances:

1. If the faintest of the simulated objects $obj_{s,n}$ (n=1,2) has a magnitude $>$ mag_{cut}, where mag_{cut} is an arbitrary threshold, the image containing such objects is not considered for the analysis, otherwise the analysis continues to point 2

2. All the detected objects $obj_{d,m}$ (m=1,2,3, ..) that do not overlap the simulated objects (e.g. spurious detections) are not interesting for the analysis and then are discarded

3. The areas of each $obj_{d,m}$ that overlaps at least one of the $obj_{s,n}$ is calculated, we define this area as Region of Interest (ROI)

4. To each $obj_{s,n}$ is associated a $obj_{d,m}$, that is not already associated, matching the minimum distance of the respective clusters centroids

5. Each non-associated $obj_{d,m}$ overlapping a $obj_{s,n}$ is defined as contaminant $obj_{c,k}$ (k=1,2,..,m) of that $obj_{s,n}$

6. The *recovery* fractions, defined as $recovery = \frac{ROI}{A_{sim}}$, are calculated for each $obj_{s,n}$, A_{sim} is the area of the $obj_{s,n}$ for which *recovery* is calculated including the one in common with the other $obj_{s,n}$

7. The *contamination* fractions, defined as $contamination = \frac{ROI}{A_{sim}}$, are calculated for each $obj_{s,n}$

8. Two thresholds are set: $\epsilon_{sim}, \epsilon_{cont}$ defining the minimum *recovery* fraction and the maximum contamination accepted, respectively

9. $obj_{c,k}$ providing a *contamination* below ϵ_{cont} are not considered in the final count of the objects detected in the image

10. If the number of detected objects is equal to the number of simulated objects in the image ($n = m$) and $recovery > \epsilon_{sim}$ then the image is considered well deblended, and the total number of well deblended images N_{okdeb} increases by 1

11. The points 1-10 are repeated for each image in the cube and in the end N_{okdeb} is divided by the total number of not discarded images (see point 1) in the cube N_{tot}

$\frac{N_{okdeb}}{N_{tot}}$ is the *deblending efficiency*. A schematic representation of the metrics is shown in Fig. 5.2.

The results obtained using this metrics, shown in Sect. 5.2.4, are obtained setting different combinations of the parameters: $\epsilon_{sim}, \epsilon_{cont}$, and mag_{cut}, we refer to them as *analysis parameters*.

5.2.4 Results

The results shown in this section are preliminary and based on a portion of the datasets, at the time of this thesis submission. The complete results are part of a scientific paper still works in progress.

From the performances comparison of the algorithms tested, obtained by applying the metrics described in Sect. 5.2.3, we notice a significant discrepancy between the multi-threshold and the topological clustering approach tested, in terms of *deblending efficiency*.

The results shown in this section are obtained with a set combination of the *analysis parameters*, each of these parameters can assume the following values: $\epsilon_{sim} = [-1, 0.3], \epsilon_{cont} = [-1, 0.1]$, and $mag_{cut} = [-1, 25.3]$. The value -1 indicates that the threshold is turned off. We arbitrarily chose the threshold values according to a visual inspection of the segmentations (post-deblending) produced. When the thresholds are all turned-on, it means that we are considering a successful deblending, if the number of detected objects corresponds to the number of simulated objects,

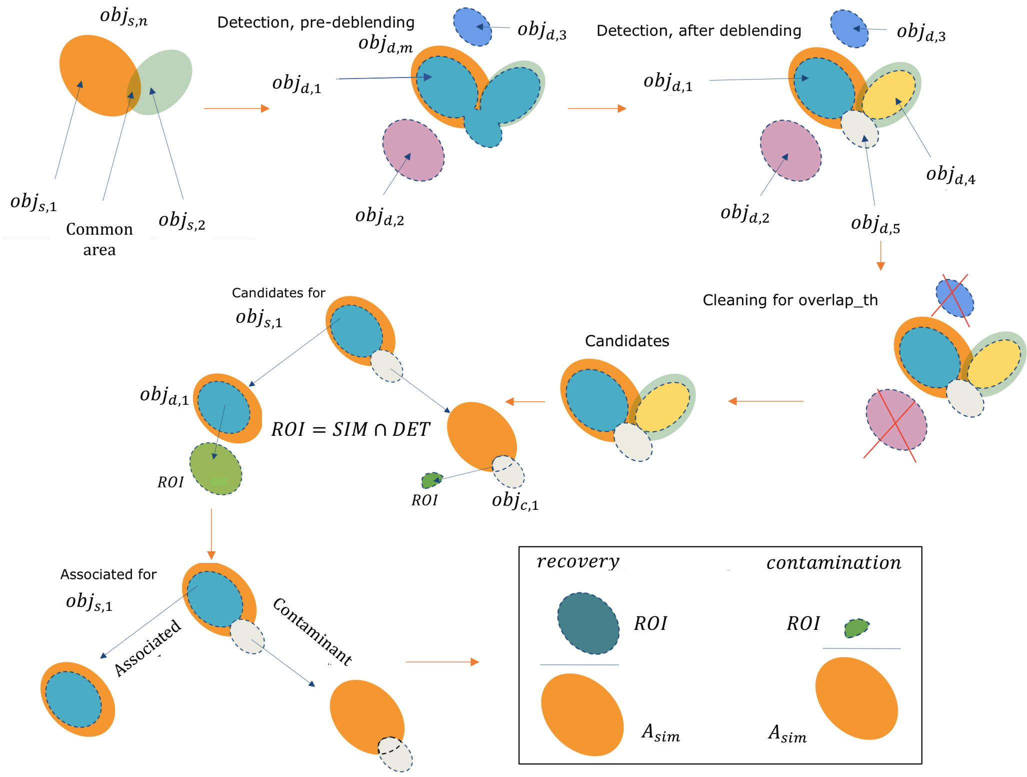

Figure 5.2. A schematic representation of the comparison metrics with two simulated objects $obj_{s,n}$. In the figure a detection without deblending is performed and in this case 3 objects have been detected $objd_{d,m}$. After that, deblending is performed and the detected object $objd_{d,1}$ is splitted in 3. The $objd_{d,m}$ not overlapping any $objs_{s,n}$ are not considered, while the remaining $objd_{d,m}$ become candidates. To each $objs_{s,n}$ is associated an $objd_{d,m}$, based on centroids distance. All the non-associated $objd_{d,m}$ that overlap an $objs_{s,n}$ become contaminants of that $objs_{s,n}$ (e.g. $objd_{d,5}$ becomes $objc_{c,1}$, contaminating $objs_{s,1}$). For each $objd_{d,m}$ the Region of Interest (ROI) is calculated, and the *recovery* and *contamination* quantities are calculated. For further details see Sect. 5.2.3

all the associated objects reconstruct at least 30% of the simulated ones, all the possible non-associated do not contaminate the simulated objects more than 10% and both the simulated objects are brighter than magnitude 25.3 (corresponding to SNR=5 for a Euclid-VIS image).

Results: Variation of the *deblending parameters*

The algorithms performances, varying only the *deblending parameters* and keeping fixed a *analysis parameters* combination, obtained for the different datasets are discussed below:

- **couples_skymaker_r1**: ASTErIsM performances for couples_skymaker_-r1 are shown in Fig. 5.3. Performances with all the *deblending parameters* combinations are obtained with the following *analysis parameters*: ϵ_{cont}=0.1, ϵ_{sim}=0.3, mag_{cut}=25.3. In the left plot, we noticed that the highest performances are obtained with the combination h_frac_min=0.05 and $h\_frac\_$-

max = 0.1. This configuration correctly deblends ∼88% of the total dataset, while in the worst scenario, ∼84% is obtained. We also noticed that variations of the *deblending parameters* produce similar results. Therefore different combinations can be set, without impacting the performances significantly. All the wrongly deblended images in the dataset are shown in the middle and right plots. With the same *deblending parameters* combination ∼10% of the images is under-deblended ($n < m$), while ∼2% is over-deblended ($n > m$).

- **couples_skymaker_r1**: SExtractor performances for couples_skymaker_r1 are shown in Fig. 5.4. Performances with all the *deblending parameters* combinations are obtained, with the same *analysis parameters* set in Fig. 5.3 (ϵ_{cont}=0.1, ϵ_{sim}=0.3, mag_{cut}=25.3). In the left plot, the highest performances, ∼83%, are obtained with the configuration: `DEBLEND_NTHRESH`=64, `DEBLEND_MINCONT`=0.0001. Differently than ASTErIsM, a significant gradient in performances, varying the *deblending parameters* combinations, is obtained. Indeed, in the worst scenario, the performance drops to ∼59%, meaning that the deblending performed by SExtractor is more sensitive to a correct configuration of its parameters. All the wrongly deblended images are divided into under-deblended and over-deblended in the middle and right plots, respectively. We noticed that in the middle plot, the performance gradient is very steep, passing from a minimum of ∼16% to a maximum of ∼41%. On the other hand, in the right plot, the over-deblending is nearly not affected by the *deblending parameters* selection.

- **single_skymaker_r1**: ASTErIsM and SExtractor performances for single_skymaker_r1 are shown in Fig. 5.5-5.6. Performances with all the combinations of the *deblending parameters* for both the algorithms (ASTErIsM: Fig. 5.5, SExtractor: Fig. 5.6) are obtained with the following *analysis parameters* combination: ϵ_{cont}=-1, ϵ_{sim}=-1, mag_{cut}=25.3. We noticed that, for both the cases the algorithms reach ∼100% performance, meaning that over-deblending does not represent a problem for any of the *deblending parameters* combinations. The performances for single_CANDELS_r1 are very similar to the ones shown in Fig. 5.5-5.6, therefore are only reported in Table 5.2.

- **big_skymaker_r10**: ASTErIsM and SExtractor performances big_skymaker_r10 are shown in Fig. 5.8-5.8. Performances with all the combinations of the *deblending parameters* for both the algorithms (ASTErIsM: Fig. 5.8, SExtractor: Fig. 5.7) are obtained with the following *analysis parameters* combination: ϵ_{cont}=-1, ϵ_{sim}=-1, mag_{cut}=25.3. In the left plots we noticed that, both the algorithms show a slight gradient in the performances, making nearly all the configurations viable. ASTErIsM reaches ∼ 99% performance with h_frac_min=0.15 and h_frac_max=0.20. h_frac_min (which defines the lower boundary used by the `scale-finder` to find the correct value of h) weights more than h_frac_max in the performances optimization, since the objects in the dataset are moderately larger than those in “couples”. On the other hand, SExtractor reaches ∼ 95% performances, with `DEBLEND_NTHRESH`=64, `DEBLEND_MINCONT`=0.01. Similarly to ASTErIsM, the optimal configuration takes into account the large size of the objects involved, preferring a high number of sub-thresholds and a less aggressive value of `DEBLEND_MINCONT`. All the wrongly deblended images are divided in under-deblended and over-deblended in the middle and right plots, respectively. ASTErIsM in its optimal configuration over-deblends $\sim$ 1% of the objects without under-deblending one. SExtractor in its optimal configuration under-deblends $\sim$ 3% and over-deblends $\sim$ 2% of the objects.

- **big_CANDELS_r10**: ASTErIsM and SExtractor performances big_CANDELS_r10 are shown in Fig. 5.9-5.9. Performances with all the combinations of the *deblending parameters* for both the algorithms (ASTErIsM: Fig. 5.9, SExtractor: Fig. 5.10) are obtained with the following *analysis parameters* combination: ϵ_{cont}=-1, ϵ_{sim}=-1, mag_{cut}=25.3. Similarly to big_skymaker_r10, in the left plots, both the algorithms show a slight gradient in the performances. ASTErIsM reaches $\sim$ 90% performance with the same h_frac_min=0.15 and h_frac_max=0.20. SExtractor reaches $\sim$ 82% performances, with `DEBLEND_NTHRESH`=48, `DEBLEND_MINCONT`=0.002. Similarly for big_CANDELS_r10, ASTErIsM optimal configuration takes into account the large size of the objects involved. On the contrary, SExtractor optimal configuration is similar only for `DEBLEND_NTHRESH`, while for `DEBLEND_MINCONT` a more aggressive value is preferred. A further investigation is needed to shed light on the different optimal configuration. All the wrongly deblended images are divided in under-deblended and over-deblended in the middle and right plots, respectively. ASTErIsM in its optimal configuration under-deblends $\sim$ 8% and over-deblends $\sim$ 2% of the objects. SExtractor in its optimal configuration under-deblends $\sim$ 15% and over-deblends $\sim$ 5% of the objects.

A discrepancy can be noted between the two “big” datasets. Indeed, both the algorithms perform worse when applied on real objects (ASTErIsM loses a $\sim$ 9% while SExtractor loses a $\sim$ 13%). We expected a similar result due to the natural complexity of real objects compared to simulated ones.

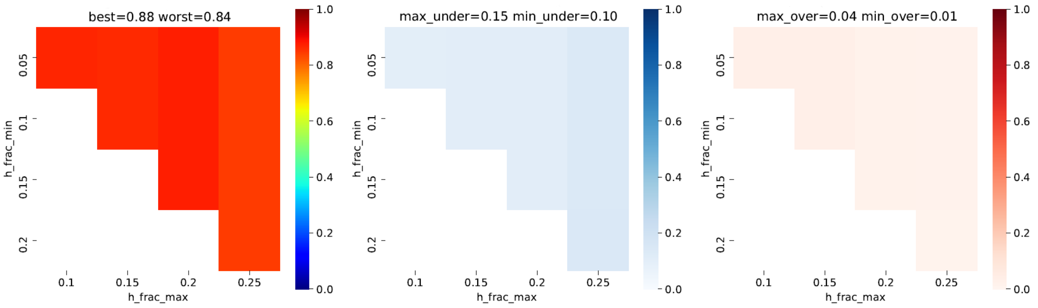

Figure 5.3. ASTErIsM performances with all the h_frac_min, h_frac_max combinations tested on couples_skymaker_r1 dataset. The left plot shows the values of the *deblending efficiency*. The central plot shows the number of under-deblended images in the dataset. The right plot shows the number of over-deblended images in the dataset.

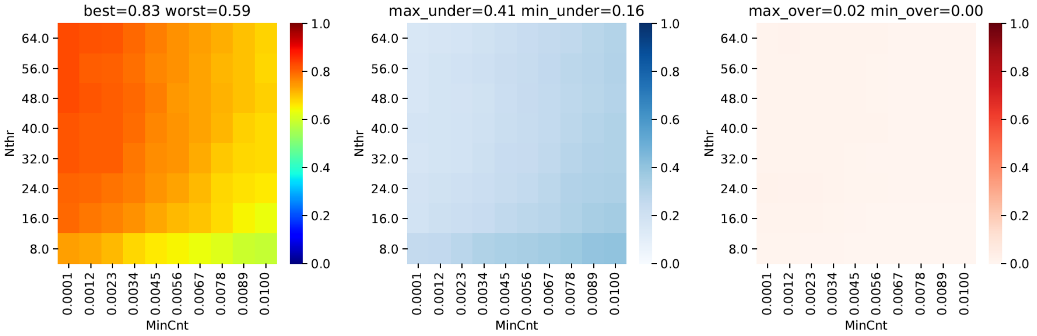

Figure 5.4. SExtractor performances with all the `DEBLEND_NTHRESH`,`DEBLEND_MINCONT` combinations tested on couples_skymaker_r1 dataset. The left plot shows the values of the *deblending efficiency*. The central plot shows the number of under-deblended images in the dataset. The right plot shows the number of over-deblended images in the dataset.

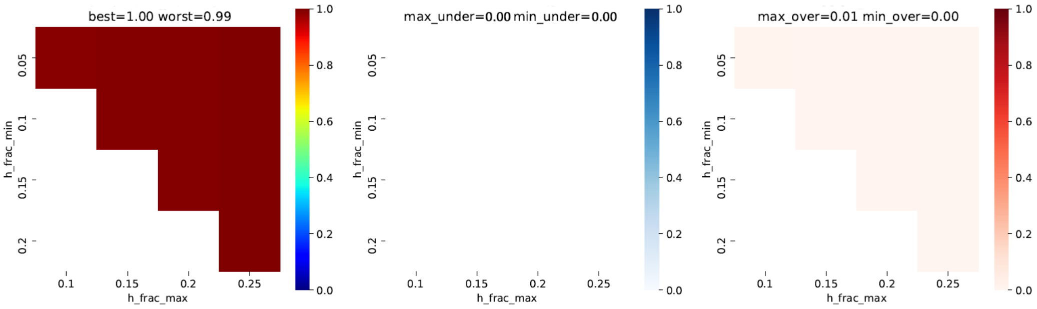

Figure 5.5. ASTErIsM performances with all the h_frac_min, h_frac_max combinations tested on single_skymaker_r1 dataset. The left plot shows the values of the *deblending efficiency*. The central plot shows the number of under-deblended images in the dataset. The right plot shows the number of over-deblended images in the dataset

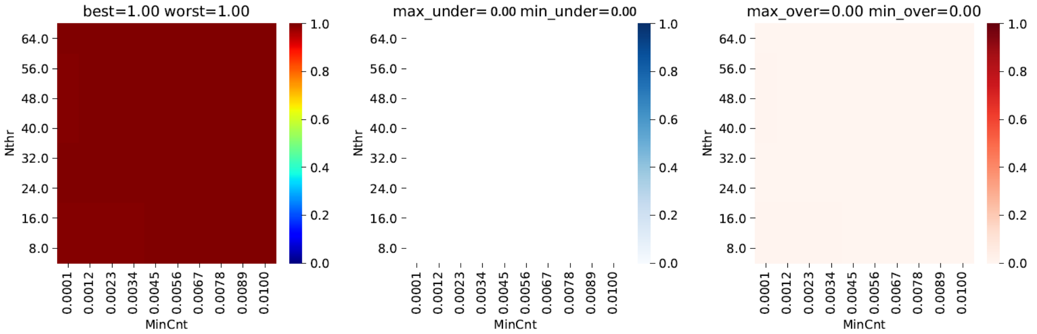

Figure 5.6. SExtractor performances with all the `DEBLEND_NTHRESH`,`DEBLEND_MINCONT` combinations tested on single_skymaker_r1 dataset. The left plot shows the values of the *deblending efficiency*. The central plot shows the number of under-deblended images in the dataset. The right plot shows the number of over-deblended images in the dataset.

Results: Overall combinations scenario and best results

In Table 5.2 the best *analysis parameters* and *deblending parameters* combinations are reported. These results refer to the best *deblending parameters* combination previously found and the *analysis parameters* already set above. Histograms of the

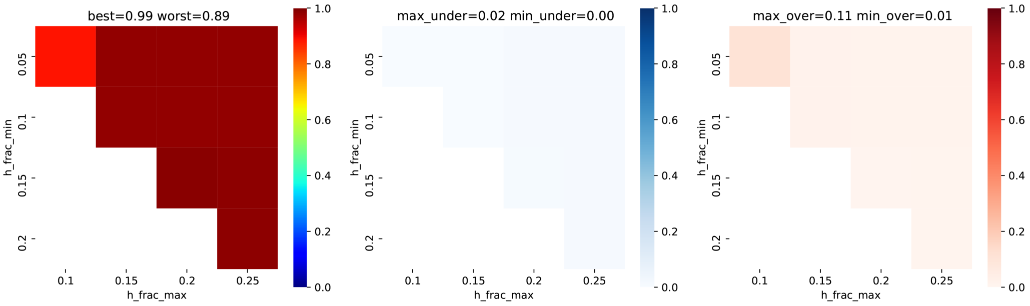

Figure 5.7. ASTErIsM performances with all the h_frac_min, h_frac_max combinations tested on big_skymaker_r10 dataset. The left plot shows the values of the *deblending efficiency*. The central plot shows the number of under-deblended images in the dataset. The right plot shows the number of over-deblended images in the dataset

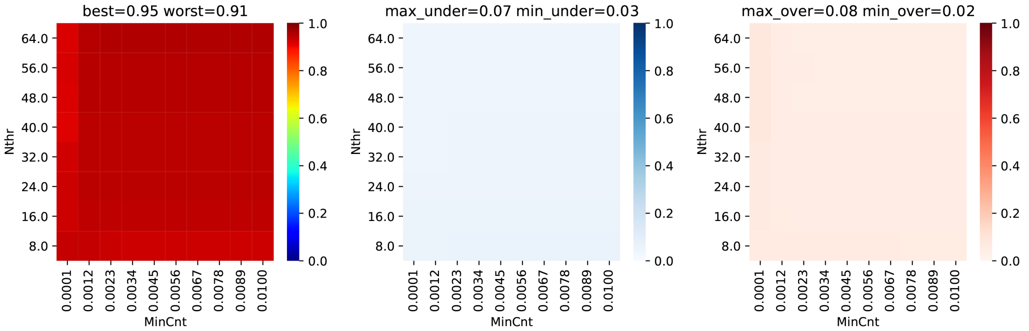

Figure 5.8. SExtractor performances with all the h_frac_min, h_frac_max combinations tested on big_skymaker_r10 dataset. The left plot shows the values of the *deblending efficiency*. The central plot shows the number of under-deblended images in the dataset. The right plot shows the number of over-deblended images in the dataset

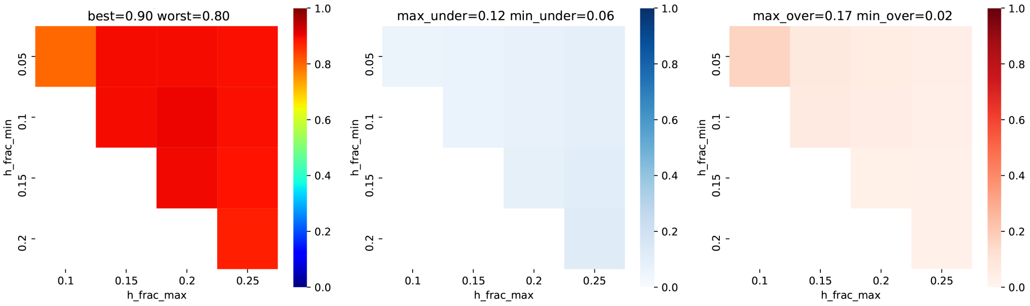

Figure 5.9. ASTErIsM performances with all the h_frac_min, h_frac_max combinations tested on big_CANDELS_r10 dataset. The left plot shows the values of the *deblending efficiency*. The central plot shows the number of under-deblended images in the dataset. The right plot shows the number of over-deblended images in the dataset

performances obtained by the two algorithms, varying all the possible *deblending parameters* and *analysis parameters* combinations, are shown in Fig 5.11. We noticed that, in both the "couple_skymaker" datasets the ASTErIsM performances distribution is narrower and has a larger mode value than the one representing SExtractor. An overlapping area between the two distributions is present, meaning

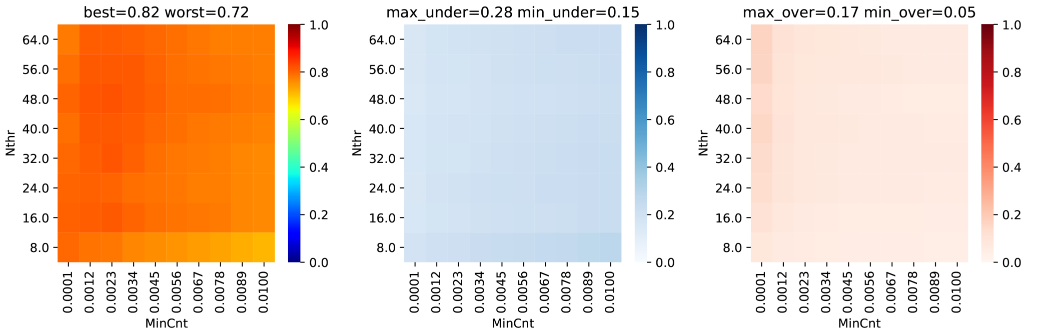

Figure 5.10. SExtractor performances with all the h_frac_min, h_frac_max combinations tested on big_CANDELS_r10 dataset. The left plot shows the values of the *deblending efficiency*. The central plot shows the number of under-deblended images in the dataset. The right plot shows the number of over-deblended images in the dataset

that for a small number of combinations the performances are comparable. On the other hand, for the "single" datasets, SExtractor distribution is narrower and the mode value is higher. In this case the difference between the mode values is small, but a further analysis is required to investigate these results. Lastly, for the "big" datasets a significant difference can be noticed in the mode values. Indeed the absence of an overlapping area, denotes that, for nearly all the combinations, ASTErIsM performs better than SExtractor.

Sample	Multi-th	Nthr	MinCnt	Denclue	Δ	h_{min}	h_{max}
couples_skymaker_r1	82.9574	64	0.0001	87.9699	5.0125	0.05	0.15
couples_skymaker_r5	78.3286	64	0.0001	82.5389	4.2103	0.05	0.10
single_skymaker_r1	99.8514	56	0.0034	99.9009	0.0495	0.05	0.20
single_CANDELS_r1	100.0000	24	0.0100	99.6936	-0.3064	0.15	0.25
big_skymaker_r10	95.3846	64	0.0100	99.3162	3.9316	0.15	0.20
big_CANDELS_r10	82.2222	48	0.0023	90.4274	8.2051	0.15	0.20

Table 5.2. In the table are shown the highest performances obtained on the different datasets tested for both the SExtractor and ASTErIsM. The results refer to the algorithms in their best configuration (represented by the *deblending parameters* discussed in Sect. 5.2.2 and the *analysis parameters* configuration: (couples: ϵ_{cont}=0.1, ϵ_{sim}=0.3, mag_{cut}=25.3; single: ϵ_{cont}=-1, ϵ_{sim}=-1, mag_{cut}=25.3; big: ϵ_{cont}=-1, ϵ_{sim}=-1, mag_{cut}=25.3). Sample is the dataset name, Multi-th is the highest value of the *deblending efficiency* obtained with SExtractor, Nthr is the `DEBLEND_NTHRESH` value, MinCnt is the `DEBLEND_MINCONT` value, DENCLUE is the highest value of the *deblending efficiency* obtained with ASTErIsM, Δ is (Δ = DENCLUE - Multi-th), $h_{min/max}$ is the value used for h_frac_min and h_frac_max. In the dataset name the $N_{repetitions}$ is indicated with r followed by the number of repetitions (e.g. $r5$ means $N_{repetitions} = 5$)

5.3 Deblending with Convolutional Neural Networks

The goal of this section is to explore if deep learning is an approach worth investigating for segmenting blended galaxies and estimating their photometry. The

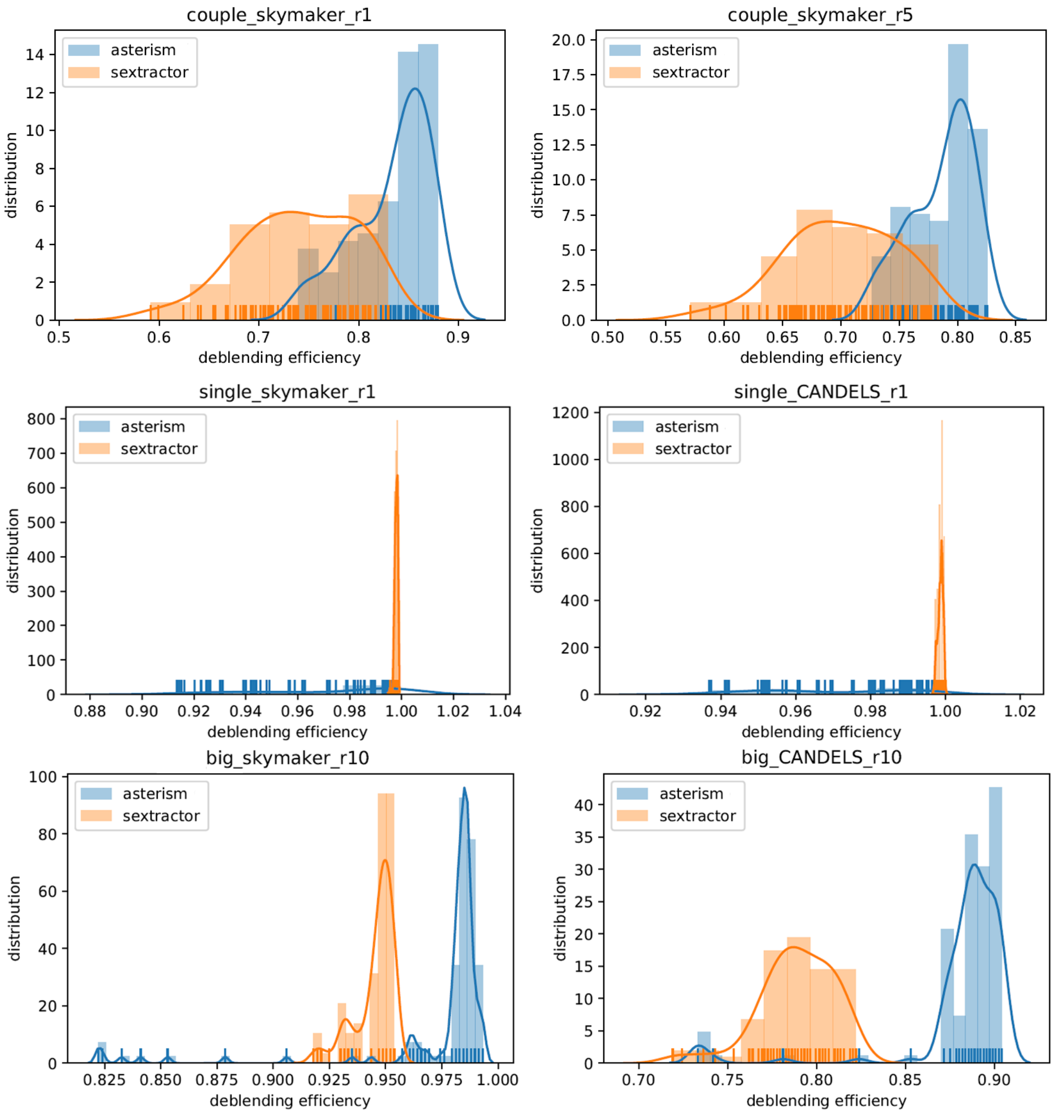

Figure 5.11. The plots show the performances of both the algorithms in all the possible configurations tested, in terms of *deblending parameters* and *analysis parameters*. On the x-axis the *deblending efficiency*. Each plot is for a different dataset.

content of this section is part of a scientific paper (for further details see [26]). We further explore the use of deep learning to both segment and measure the photometry of blended pairs of galaxies.

The goal is thus to obtain a neural network optimized to predict the photometry of pairs of galaxies observed with fairly high spatial resolution in one single band.

5.3.1 The dataset

The quantification of the effects of blending on the derived galaxy properties is a difficult task by nature, due to the integration of photons by our sensors and the intrinsic convolution by the point spread function of the instrument. Most existing methods require additional knowledge (several wavelength bands), or a priori knowledge, like parametric models, of the galaxy profiles, symmetries, etc. Moreover, to assess the accuracy of such methods, we are often left with bottom-up approaches like the simulation of galaxy blending using software like GalSim [146], for which we have access to the true light distribution of each object in the image. But as realistic as they can be, simulated images often show their limits when compared with the diversity and the singularity of real data images [72]. This is particularly critical for machine learning which implicitly assumes that the training sets are fully representative of the real data.

In order to get a realistic representation of observations,for this work we decide to simulate blended objects from real observations. Although this approach eventually propagates the biases and errors existing in the observations, it has the advantage of including fully realistic morphologies. We describe in the next paragraph the methodology we follow to generate our galaxy sample.

Parent Sample

The parent sample used is the H-band selected catalogue from the Cosmic Assembly Near-infrared Deep Extragalactic Legacy (CANDELS) survey, presented in [48]. The catalogue contains galaxies with F_{160W}<23.5, for which both visual morphologies and parametric bulge-disc decomposition are performed. From this parent data set, we first define a clean sub-sample of isolated galaxies with unambiguous morphologies that are then used to perform the blends. More precisely, we use the neural-network-based morphological classification published in [78] and select galaxies with four different morphological types:

- pure bulges: $P_{SPH} > 0.8$
- pure disks: $P_{DISK} > 0.8$
- two component bulge + disk: $P_{SPH} > 0.8$ & $P_{DISK} > 0.8$
- irregular galaxies: $P_{IRR} > 0.8$

Note that the purpose of this selection is not to have a complete sample of galaxies, but to have a clean data set of isolated galaxies with different morphologies for which we can reasonably trust the segmentation procedure. By selecting galaxies with very large probabilities of being in a given morphological type we can be reasonably certain that we remove originally blended systems or complex structures such as mergers. From this initial sample, we generate 128×128 pixel stamps centred on the objects. We then remove all other objects present in the stamps. To that purpose, we apply a morphological dilation to the original segmentation obtained with SExtractor and replace all distinct regions with random pixels sampled from empty regions in the background.

In order to further clean the sample, we visually inspect the selected galaxies and remove the ones which still present anomalies such as originally blended systems not detected by SExtractor, or the ones for which the removal of companions created some visual artefacts in the images.

Blending

To create the artificially blended systems, we combine the galaxies of the clean sample we just obtained using the following procedure. First of all, we randomly select one galaxy, referred to as the *central galaxy*, with a magnitude and an effective radius respectively denoted mag_{cen} and R_{cen}. R_{cen} is the semi-major axis of the best Sersic fit model from the catalog by [48]. Secondly, we pick a second galaxy in the catalogue, referred to as the *companion galaxy* with properties mag_{comp} and R_{comp}, so that it satisfies $mag_{cen} - 2 < mag_{comp} < mag_{cen} + 2$. Then we set R=max(R_{cen},R_{comp}) as the biggest effective radius between the two galaxies and randomly select a couple of shifts (Δx,Δy) from a uniform distribution ranging from $0.5 \cdot R$ and half of the image size. We use these shifts to apply a translation to the stamp of the *companion galaxy*. Finally, the blend is created by adding up the pixels of two stamps. Note that the blending process contains two over-simplifications as compared to real observed blends. Firstly, we avoid overlap in the very inner parts of the *central galaxy* ($<0.5R_e$) and secondly, the *central galaxy* is always placed at the centre of the stamp. We are fully aware of these simplifications but consider this enough complexity for our blends in a first proof-of-concept work.

We repeat this process to build up a sample of 30,000 blended galaxies, which necessarily contains some redundancy because each galaxy appears in multiple stamps. However since there are enough degrees of freedom coming from the selection of the companion and the shifts, this redundancy is not to be considered problematic. It allows us to build a large enough sample to train the networks as described in the following. We show in Figure 5.12 some examples of blended pairs with different magnitude differences and distances between the two galaxies.

With the purpose of triggering the comparison with other approaches, the software used to generate the blends as described above has been publicly released as a package called `candels-blender` [3]

5.3.2 Training, Validation and Test datasets

As explained in the previous sections, the blend stamps contain some level of redundancy since the same galaxy can appear in several of them. This could artificially improve the results evaluated in the test set because the network might have seen already the same galaxy in the training phase. To avoid this potential bias, we adopt a specific procedure. Following a standard approach in machine learning [179], we split the dataset into three subcategories: training, validation and test, respectively 60%, 20% and 20% of the full dataset. During the training, the model loss (i.e. cost function) is periodically computed on the validation sample to ensure it is not diverging from the training, which would indicate over-fitting or a bad convergence of the network.

Training and validation samples can be randomly selected from the same dataset, however the test sample, on which the metrics are computed, must be carefully chosen to be both distinct from and representative of the sample used for the training and validation. To achieve this feature and obtain meaningful results, we isolate the sample of galaxy stamps used for the test dataset at the very beginning by randomly

[3] https://github.com/aboucaud/candels-blender

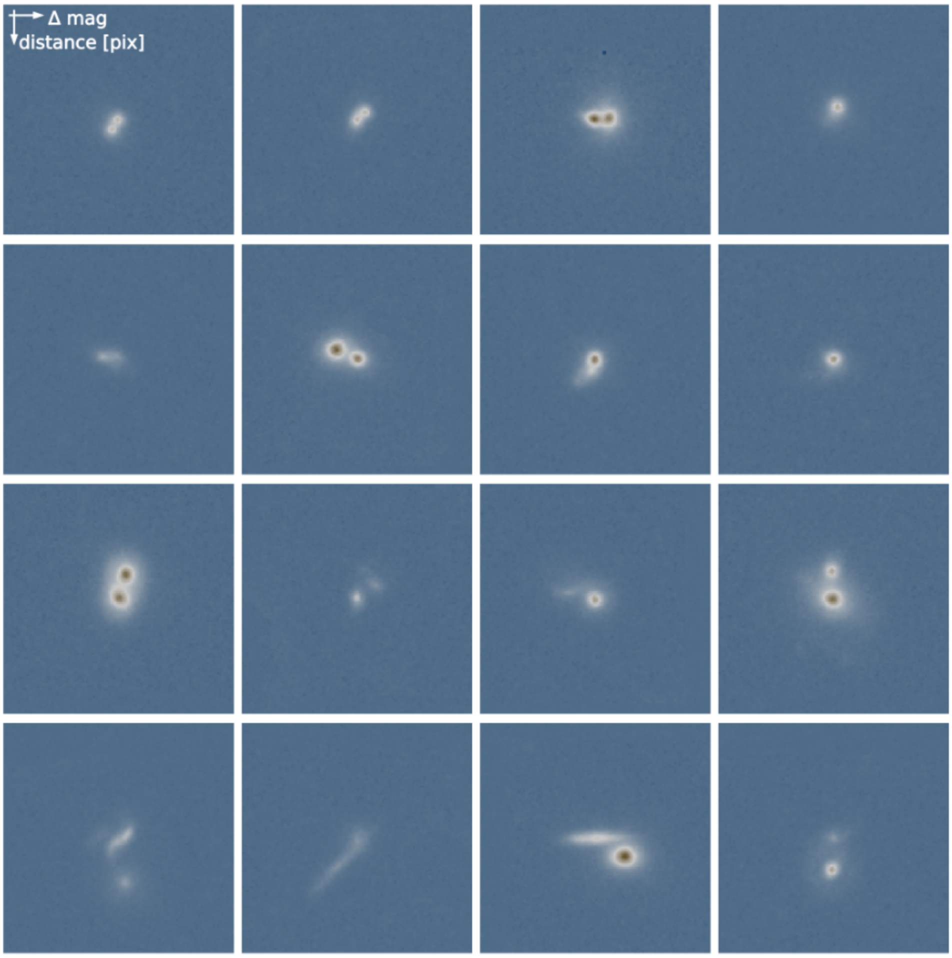

Figure 5.12. Selection of blended systems created and used in this work. The stamps are ordered vertically by the distance in pixels between the galaxy centres, and horizontally with respect to the magnitude difference between the galaxies. The images have all been asinh-stretched for visualisation purposes.

picking them out of the catalogue. This way, all the galaxies used to construct the blends for the training and validation are never to be found in the test sample of blends,and vice-versa. In the end, we have a training/validation set composed of 25,000 blends and a test set of 5,000 blends. This generated data set is used to train several deep neural network architectures as described below.

5.3.3 Methods

Our goal is to recover, with deep learning, the photometry of the two galaxies before the blending process. The sample being made of real galaxies, we make the assumption that the ground truth (also referred to as the target in supervised learning) is the flux of the isolated galaxy computed by SExtractor on the original CANDELS cutout. We also assume that the segmentation mask provided by SExtractor for the isolated galaxies is correct. We understand that these are strong assumptions. However, the main purpose of the work does not depend on the absolute accuracy of the training sample. The main objective is to calibrate how well we can recover the photometry of blended galaxies relative to the accuracy obtained on the same galaxies when they are isolated. In that respect, the ground truth can be replaced with any other measurement.

We perform two different experiments. In the first one we use a standard Convolutional Neural Network (similar to FCN [104]) to directly compute the fluxes of the two galaxies from the blend image. We call this configuration `blend2flux`. In the second experiment, we recover with a unique architecture, both the segmentation maps and the fluxes of the two galaxies. The idea is to calibrate whether having information on the segmentation map helps the network to obtain a more reliable photometry. We call this second experiment `blend2mask2flux`. The networks are implemented, trained and evaluated using the Python API Keras [4], which runs on top of TensorFlow [5].

blend2flux

As seen in Sect. 3.2.4 with deep learning reducing pre-processing to a minimum often provides better results. We thus start off with a deep neural network model that predicts fluxes directly from the blended images without any intermediate step. We use to that purpose a standard CNN configuration including a feature extraction convolutional part followed by a fully connected network. The input of the network is thus a 1-channel image with two blended galaxies and the output is a vector of two floating numbers corresponding to the fluxes of each galaxy.We build a modular version of this sequential network, where the number of layers of both the convolutional and the dense network, as well as their filter size are adjustable.

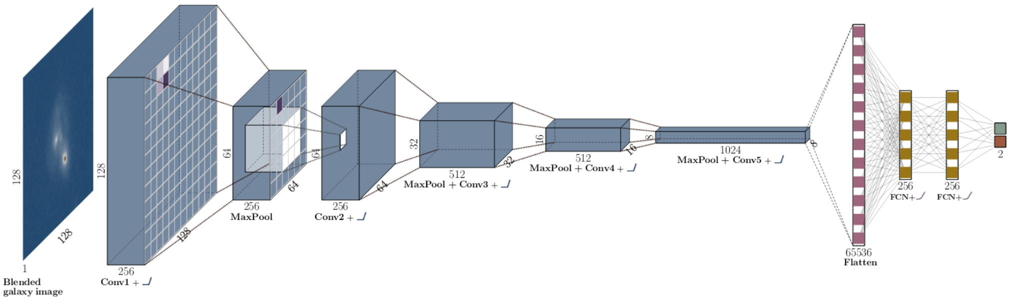

Figure 5.13. Schematic representation of the fiducial `blend2flux` network. The network takes as input an image of a blended system and outputs the fluxes of the two galaxies. The blue boxes correspond to the convolutional part of the network. The yellow part is the fully connected section. The sizes of the different layers and convolutions are also indicated.

The architecture, whose results are shown in this work, is sketched in detail in Figure 5.13. The CNN part is made of five convolutional layers activated using a ReLU function and using convolution kernels of size 3×3 only. Max-pooling layers are inserted in between each convolution layer to downsample the images. The first

[4] https://keras.io

[5] https://github.com/tensorflow/tensorflow

Initial Filter Size	64	256	512
No. parameters [million]	1.6	25.7	102.7
Flux error central [%]	9.33	8.39	8.25
Flux error companion [%]	8.79	8.01	7.98
Total Flux error [%]	9.06	8.20	8.12

Table 5.3. `blend2flux` network performance computed on the entire test set using mean absolute percentage error (MAPE).

layer starts with a filter size of 256, and doubles this filter size every other layer. After the fifth convolutional layer, the data is flattened to be fed to a three-layer classical neural network, finally yielding a vector of size two with the fluxes. Given that our network is aiming at correct relative flux measurements, we choose to use the mean absolute percentage error (MAPE, see equation 5.6) as our loss function. To adjust the weights during training, we select the ADAM optimizer. ADAM is an extended stochastic gradient descent algorithm, meaning that it iteratively updates network weights with individual adaptive learning rates based on both first and second moments of the gradients.

$$MAPE(y_{meas}, y_{true}) = \frac{100}{n} \sum_{n} |\frac{y_{true} - y_{meas}}{y_{true}}| \tag{5.6}$$

This `blend2flux` network, which has about 25.7 million free parameters, is then trained from scratch using the training set of 25,000 images. We consider the network as having converged after the validation loss, computed on the validation part of the training sample, stays on a plateau for a full ten consecutive epochs after having decreasing the learning rate several times [180]. For this network, it happened after 70 epochs which took less than five hours of training on an Nvidia K80 GPU.

The network built being modular, we trained a few variations around the fiducial network presented above to compare their relative performance. The results of the various network models as a function of the number of filters for the first convolutional layer are summarized in Table 5.3 with the fiducial results in the middle column. The table shows that doubling the initial filter size (right column) only slightly increases the performance on the validation set regarding the estimated fluxes in Section 5.3.4, at the expense of quadrupling the number of parameters (hence the training time and computation cost). Using instead a smaller network with an initial filter size of 64 (left column) reduces the number of parameters to about 1.6 million, which has a higher impact on the performance ($\sim$ 1%worse). The network still reaches though a precision below 10% on estimated fluxes, despite being significantly reduced in size. We therefore want to stress here that smaller and simpler networks than our fiducial one still outperform traditional methods.

blend2mask2flux

In a second experiment, we aim at recovering the individual segmentation maps for the two galaxies in addition to the photometry. The objective of this exercise is to quantify if the segmentation maps contain additional information that the networks can use to improve the photometry. We achieve this objective using a concatenation of two different networks, one to produce the segmentation maps,and a second to predict the fluxes from the segmentation maps and the blend image. We call that composite network `blend2mask2flux`. One important constrain when building this network was to ensure it had approximately the same number of free parameters as the fiducial `blend2flux`. To produce the segmentation maps, we use the deep network architecture of U-Net [145]. U-Net was designed to perform bio-medical image segmentation and has already proven useful to detect and segment overlapping chromosomes.

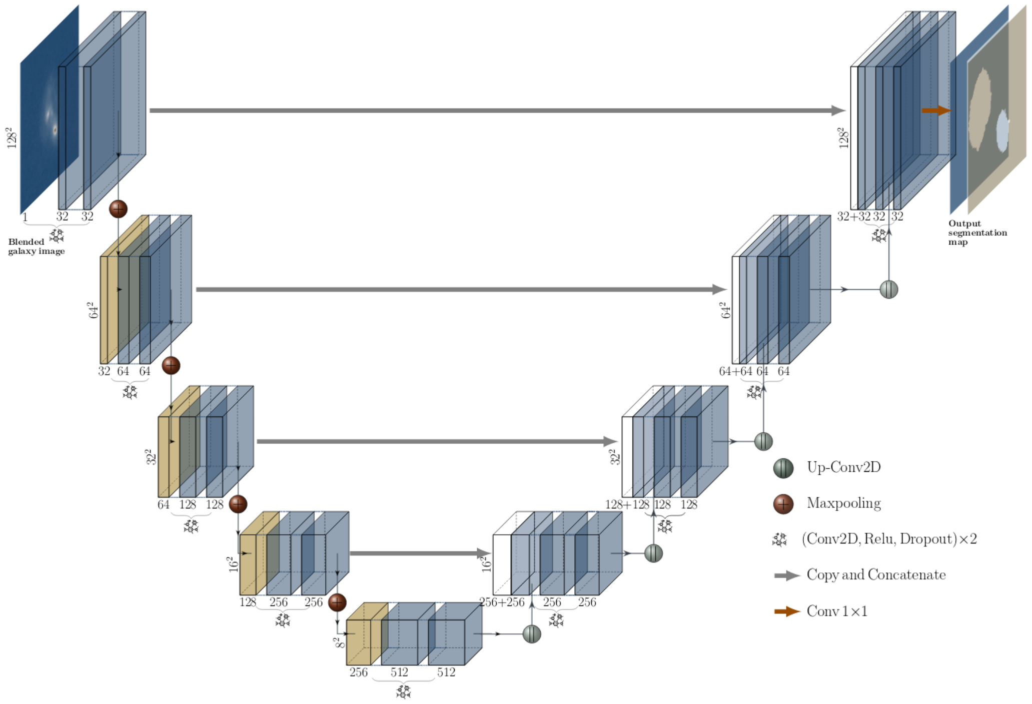

Figure 5.14. Schematic representation of the U-Net part of the `blend2mask2flux` network. The network takes as input an image of blended system and outputs a segmentation map. The lines indicate the connections among the different layers.

The network architecture is quite unique and characterized by an ability to capture both fine and large scale information of the input image by keeping a copy of each downsampling step (convolution+max-pooling) and concatenating it at the upsampling step. For our purpose, we create a modular version of the original U-Net architecture made of blocks of two convolutional layers activated with ReLU, followed by either a downsizing or upsizing layer (respectively, max-pooling and up convolution layers). Because the output images are of the same shape as the input blend, each downsizing block is associated with an upsizing one in the network, and the model can therefore be parametrized by the number of consecutive downsizing blocks, as well as the size of the filters (number of convolution kernels). After some tests and with a range of these parameters, we selected a U-Net with a depth of 5 and an initial filter size of 32, which we also refer to as the fiducial model. The exact architecture of this segmentation network is depicted on Figure 5.14. The last activation of the model is a sigmoid function. These pixels are then thresholded to

Parameter	Hot	Cold
DETECT_THRESH	4	5
DETECT_MINAREA	6	10
DEBLEND_MINCONT	0.0001	0.01
DEBLEND_NTHRESH	64	64

Table 5.4. SExtractor parameters for hot and cold modes.

obtain segmentation maps with binary values 0,1 and we use a binary cross-entropy loss to train the model. Further results of this pure segmentation stage will be discussed in Sect. 5.3.4.

The second part of this composite model is the retrieval of the photometry using the blend image and the segmentation maps obtained with the U-Net. For this part, we use an architecture similar to the blend2flux model, with a reduced number of free parameters, and changing the input to a 3-channel input - the concatenation of the blend image, the segmentation of the central galaxy and the segmentation of the companion galaxy - (instead of 1-channel - the blend image - in the original blend2flux network). Like the blend2flux model, the output of the network is evaluated using the mean absolute percentage error loss.

The composite blend2mask2flux network is trained following a particular process. First, the U-Net is trained alone to produce accurate segmentation maps of the two galaxies. Then we load the pre-trained weights of the U-Net into the blend2mask2flux network, and train the network end-to-end with respect to the flux retrieval, i.e. using the mean absolute percentage error loss on the photometry. Note that we still keep the loss on the segmentation part but with a weight of 0.1 compared to the photometry loss. This last optimisation step, during which we optimize the network with respect to both the segmentation and the photometry loss,also fine-tunes the segmentation maps for flux measurement. A more detailed discussion on this aspect can be found in Section 5.3.4.

This blend2mask2flux network presented above has 18.5 million free parameters, a number very close and even inferior to the fiducial blend2flux model. The U-Net part is trained from scratch on the 25,000 image training set for about 50 epochs. Then the end-to-end blend2mask2flux network is trained during a few hundred epochs with a small learning rate. This full process takes about fifty hours of training on a Nvidia K80 GPU, much longer than that of the blend2flux network. Both the model complexity, and the training process (reduced batch size for the U-Net training) are accountable for that order of magnitude time difference.

Baseline: SExtractor

In order to have a baseline to compare with, we also run a classical SExtractor segmentation procedure on the blended systems. We highlight that the comparison is not completely fair since SExtractor does not only measure photometry but also detects the objects without any prior on the number of existing objects. However, the two deep learning approaches implicitly incorporate a prior on the number of blended galaxies through the training set (networks are trained only with images containing two objects).

In order to minimize that effect, and help SExtractor as much as possible, we adapted the procedure reported by [61], where SExtractor is first ran in a cold mode, aiming to select the larger elements in a blended image followed by a second round where it is ran in a hot mode - which is more sensitive to small structures. In our particular case, where the data is known to have only two elements, the cold mode was used to scan all the images and a subsequent run with the hot mode was restricted to those images for which SExtractor identified only 1 object. Our code used the Python package `sep` [13] and the parameters used for both modes are described in Table 5.4. Following this procedure, results can be divided in 3 cases:

- SExtractor detects exactly two galaxies (75%): fluxes were associated with central or companion galaxy based on the closest detection.
- SExtractor detects only a single object, meaning it is not able to deblend the pair (22%): detected object was associated with the central galaxy if its measured centroid is located within 0.5 R_{cen} from the centre of the image. Otherwise, detection was associated with the companion galaxy.
- SExtractor over-deblends and detects more than two objects (3%): the two brightest detections were considered, others were ignored.

5.3.4 Results

We now evaluate the results of the three experiments described previously. The main objective is to test the photometric accuracy of blended objects as compared with the photometry obtained on the same objects when they are isolated. We use the magnitude difference as the main indicator of accuracy and explore the results as a function of two main parameters: the magnitude difference between the two galaxies and the distance between the two galaxy centroids.

Overall photometric accuracy

Figures 5.15 and 5.16 show the recovered magnitude in the blended systems (hereafter output magnitude) for the three different methods, the `blend2flux` and `blend2mask2flux` networks and SExtractor, as a function of the magnitude measured on the same isolated galaxies (hereafter input magnitude). On Figure 5.15 we focus the results on the central (top) and the companion (bottom) galaxy using the blends for which SExtractor detects them. On Figure 5.16, we aggregate the results on both galaxies, and distinguish the cases for which SExtractor detects the pair (top) and over- or under-deblends (bottom). On both figures, the deep learning architectures behave very similarly. The relation between the two quantities is centred on the one-to-one line and the typical scatter is ~ 0.1 magnitudes. The scatter is roughly constant over all the luminosity range explored which means that the photometry can be recovered with similar accuracy for bright and faint objects in our sample. This is clearly not the case for the SExtractor results which present a noticeable increase of the scatter at the faint end. This difference highlights an important advantage of the machine learning approaches. If the training set is representative of the real data, the algorithm optimises the loss for all objects equally.

In each panel of Figures 5.15 and 5.16, we quantify in more detail the bias and scatter on the recovered photometry. The embedded histograms show the distribution of photometric error Δ_{mag}=mag_{blend}-$mag_{isolated}$ between the output and input magnitudes. The distributions for both the central and the companion galaxy are generally well centred around zero for the three codes, which indicates that the estimators are globally unbiased. We note that the SExtractor panels present a slightly skewed histogram and positive bias of 0.1 mag for the central galaxy. We explain this slight bias by looking at the selection process of the companion galaxy described in Section 5.3.1, which is skewed a bit towards selecting fainter galaxies than the central ones. The visible difference between the methods are shown in the scatter. Both deep learning approaches present a very low scatter of $\sim$ 0.1 mag compared to the ∼0.5-0.7 mag scatter of SExtractor. Another good indicator of the model performance, used for training the models, is the MAPE, computed here on the magnitude. Again, both network model show good and similar performance, with always a slight advantage for the blend2mask2flux, whereas SExtractor is distanced. These two indicators show an overall improvement of the measured photometry of a factor four using the deep learning models compared to using SExtractor on these blended galaxies. Another important difference between the methods is the fraction of catastrophic errors, i.e. cases in which the estimated photometry in the blended systems significantly differs from the input value. We arbitrarily set the threshold value to define catastrophic errors to $|\Delta_{mag}| > 0.75$, which corresponds to an error of a factor of two in flux. The fraction of outliers defined that way is two orders of magnitude smaller with the deep learning methods compared to SExtractor. Both network architectures achieve a comparable fraction of ∼0.1% outliers whereas the SExtractor fraction is of the order of ∼10%, even when restricting the results to the cases where SExtractor detects both objects (see top panel of Figure 5.16). Lastly, as shown on the bottom panels of Figure 5.16, the performance of both `blend2flux` and `blend2mask2flux` models on the galaxies that SExtractor did not manage to accurately deblend (25%) gets affected compared with the well deblended cases (top panels) but remains unbiased with a low scatter and an outliers rate below 0.4%

Photometric accuracy vs. blended properties

Aiming for an unbiased performance for a range of blend properties, we report results as a function of the magnitude difference between blended galaxies and the distance between the two objects. In Figure 5.17 we show the magnitude difference between the isolated and blended galaxies (the bias in our magnitude estimate) as a function of the difference in magnitude between the two galaxies blended together. We observe that the two deep learning approaches present a very stable behaviour across the whole range of magnitude difference. As expected, the bias slightly increases when one of the galaxies in the pair is significantly brighter. However, it remains below ∼0.05. Overall the bias remains always lower than the SExtractor-based estimates. The deep learning results are also very stable in terms of scatter which is of the order of ∼0.1 magnitudes. Here the scatter for SExtractor-based estimates is always significantly larger (∼0.25 magnitudes) than for the networks, and also shows a strong increase with magnitude difference between *central galaxy*

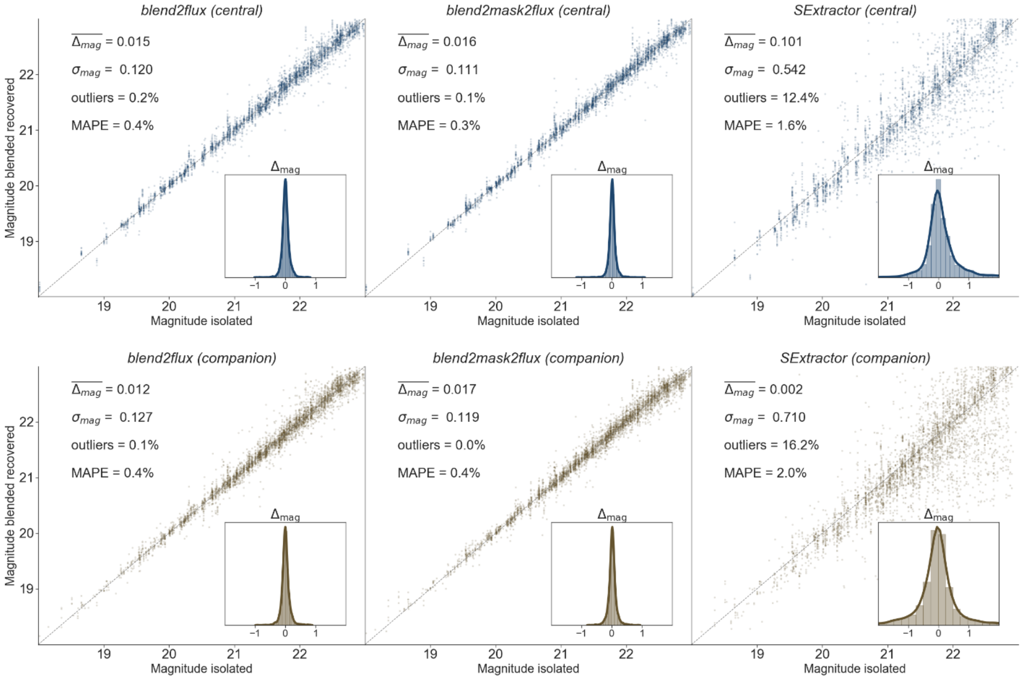

Figure 5.15. Magnitude measured on the blend systems as a function of the magnitude measured by SExtractor on the same isolated galaxies (isolated magnitude). The top row shows the results for the central galaxy using the blends for which SExtractor detected either the two galaxies or only the central one. The bottom row shows the results for the companion galaxy using the blends for which SExtractor detected either the two galaxies or only the companion. The columns refer to different codes or models applied to the blend images, respectively from left to right `blend2flux`, `blend2mask2flux` and SExtractor. The dashed line denotes identical estimation from blended and isolated galaxy images to guide the eye. The inner panels show the histograms of photometric errors (Δ_{mag}=mag_{blend}-$mag_{isolated}$). The numbers in each panel indicate the average photometric error $\overline{\Delta_{mag}}$, the dispersion σ_{mag}, the fraction of outliers, defined as $|\Delta_{mag}|$>0.75, and the mean absolute percentage error (MAPE) on the magnitude.

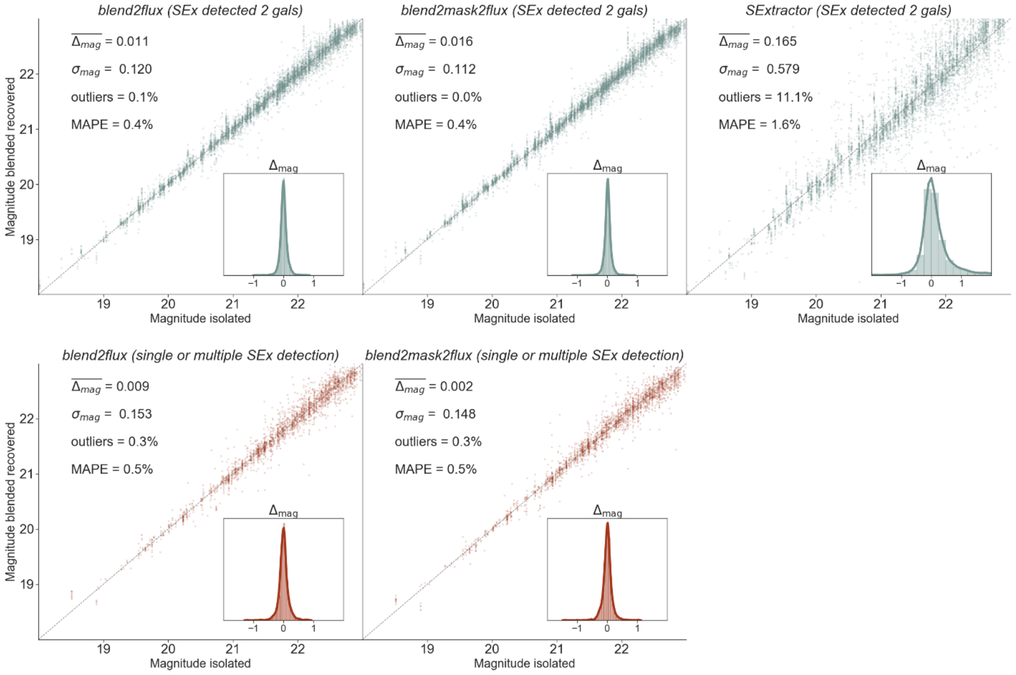

Figure 5.16. Magnitude measured on the blend systems as a function of the magnitude measured by SExtractor on the same isolated galaxies (isolated magnitude). The top row shows the results for the central and companion galaxies on the blends for which SExtractor detected exactly two galaxies while the bottom row show the results on the blends for which SExtractor detected either one or more than two galaxies (under- or over-deblending). The different columns indicate different codes or models applied to the blend images, from left to right `blend2flux`, `blend2mask2flux` and SExtractor. The dashed line denotes identical estimation from blended and isolated galaxy images to guide the eye. The inner panels show the histograms of photometric errors (Δ_{mag}=$mag_{blend} - mag_{isolated}$). The numbers in each panel indicate the average photometric error $\overline{\Delta_{mag}}$, the dispersion σ_{mag}, the fraction of outliers, defined as $|\Delta_{mag}|$>0.75, and the mean absolute percentage error (MAPE) on the magnitude

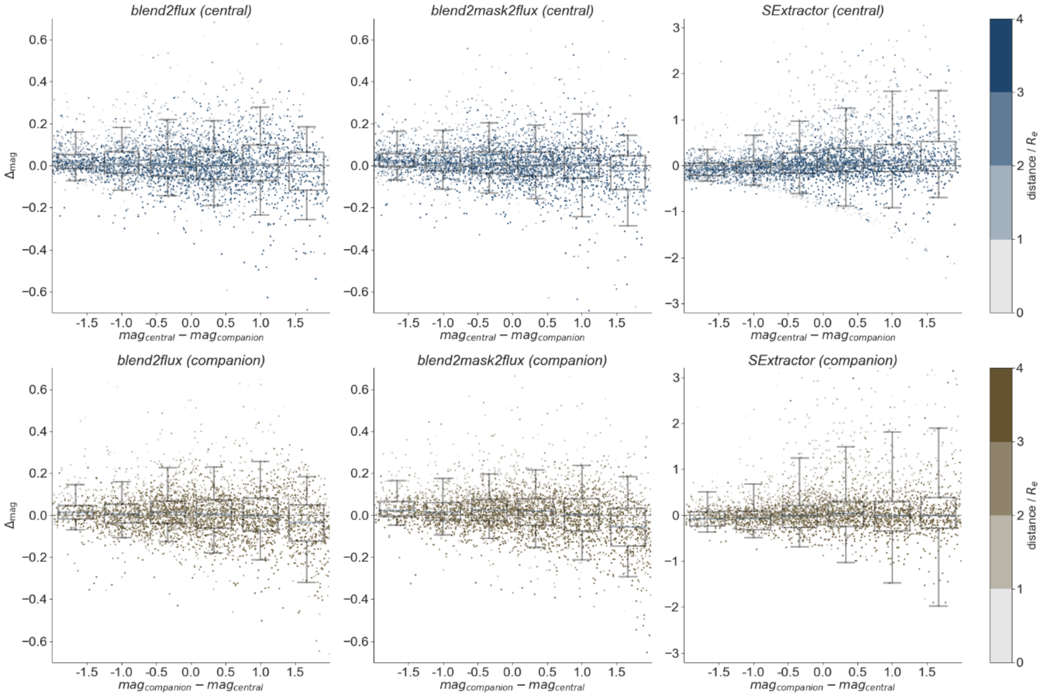

Figure 5.17. Magnitude difference (Δ_{mag}) between the same galaxies when they are isolated (input) and blended (output) as a function of the magnitude difference in the blended system ($mag_{central} - mag_{companion}$). The top row shows the difference for the central galaxy. The bottom row corresponds to the companion galaxy. The columns indicate different codes. From left to right: `blend2flux`,`blend2mask2flux` and SExtractor. The boxplot marks the median and interquartile range (25% - 75%) for different bins in magnitude difference. The lines emanating from the box extend from 5^{th} to 95^{th} percentile of the data in each bin. The colour bar shows, for each blend, the distance between the objects normalized to the effective radius of the central galaxy.

and *companion*. For both networks this trend is only slightly visible.

Figure 5.17 also encodes in the color bar the normalized distance between the two galaxies. Again, the deep learning results display little photometric dependence with distance,for both central galaxy and companion. The SExtractor results show a clearer dependence with distance, underestimating the fluxes up to 1 magnitude for close objects ($<R_e$).

Photometric accuracy and morphology

The galaxies in our sample are classified into four morphological types (pure bulge, pure disk, two component bulge+disk, irregular) and are distributed as was shown in Table 5.3.1. One major property of the machine leaning methods presented here is that they do not assume any prior on the galaxy shape (as opposed to model fitting techniques). We explore in Figure 5.18 the dependence of the photometric accuracy on the morphological type. We plot the median bias and scatter in bins of magnitude and distance now divided by morphological type. In general, the machine learning approaches show little dependence on performance with respect

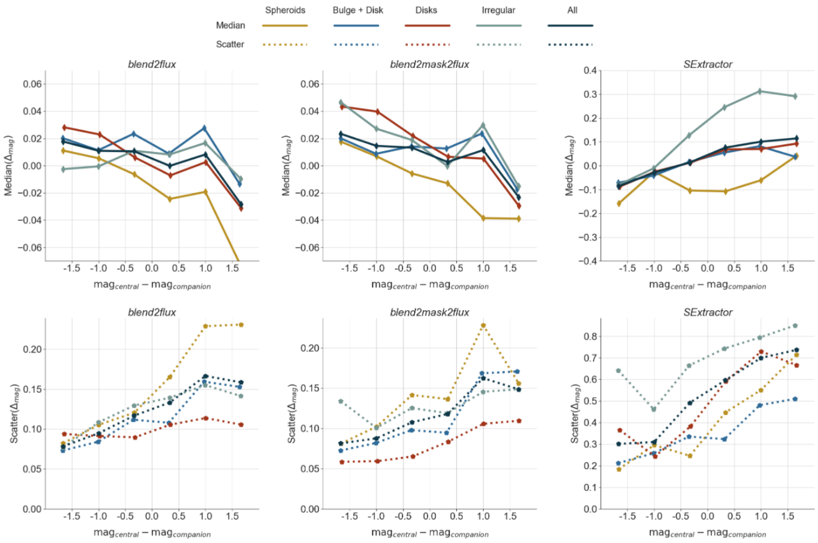

Figure 5.18. Dependence of photometric bias (solid lines, top panels) and scatter (dotted lines, bottom panels) on the morphological type for the three codes considered in this work as a function of the magnitude difference. From left to right, the different panels show the results for `blend2flux`, `blend2mask2flux` and SExtractor respectively. The different colors indicate the morphological type: spheroids (yellow), disk+spheroids (blue), disks (red) and irregulars (light green). The dark blue lines show the results for all galaxies.

to morphology. As expected, irregular galaxies are harder to measure, and hence present a marginally larger scatter from both codes. Surprisingly, spheroidal galaxies tend to present larger errors when these galaxies are fainter than the other galaxy in the blended system ($\Delta_{mag} > 0$).This behaviour seems to be present in both codes. The reason for this is unclear. One possible explanation is that the outskirts of the spheroids are too faint to be detected. Since these objects typically have large Sersic indices (i.e. steep luminosity profiles), the fraction of light in the outskirts is not negligible. SExtractor presents similar trends but overall more dramatic. In particular, the bias in the photometry of irregular galaxies is ∼0.2 larger than for the whole population. Also the luminosity of spheroids is systematically underestimated. As can be seen in Figure 5.18 the photometric accuracy (magnitude scatter) overall is considerably lower for our two deep learning algorithms than for SExtractor results.

Segmentation maps

By now segmentation maps have been considered as a by-product of both SExtractor and the `blend2mask2flux` network, possibly improving the photometry. Here, we focus on the recovery of the segmentation maps of blended galaxies from the deep learning architecture, as well as the comparison between the results of the initial training of the U-Net alone and the ones after the training of the full `blend2mask2flux` network, which is characterized by the tuning of the segmentation maps for photometry. The U-Net is in charge of reproducing the two SExtractor segmentation maps of the original CANDELS galaxy cutouts from the blend im-

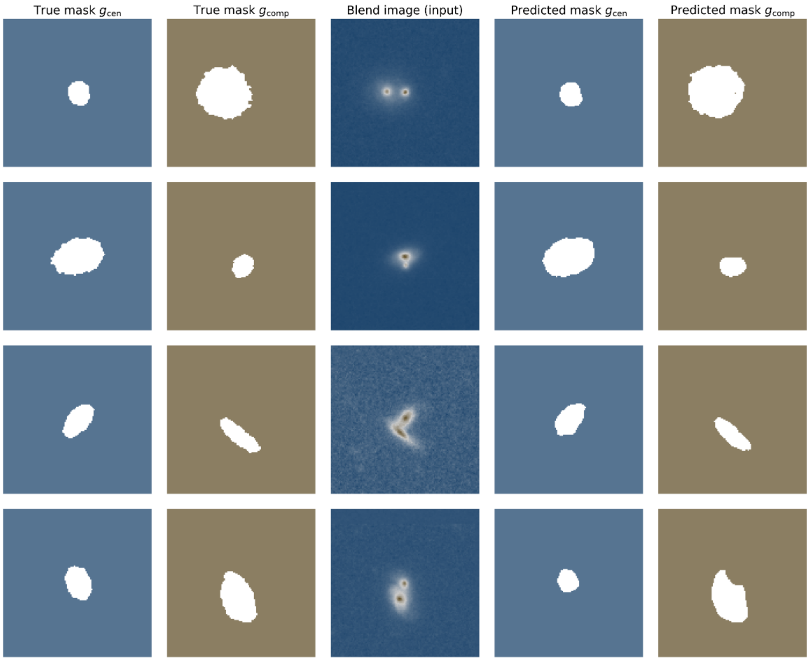

Figure 5.19. Selection of four simulated blends from the test data set and the recovery of the individual galaxy masks through the `blend2mask2flux` network. At the centre is the stamp of blended galaxies that is input of the network. On the left are the segmentation masks obtained on the individual galaxy images with SExtractor, and on the right the segmentation masks recovered by the network out of the blend image.

age. In other words, its task is to take as input the full 128×128 blend image and produce two binary 128×128 images that correspond to the masks of the *central* and *companion galaxy*; this can be seen for a selection of four blends in Figure 5.19. For better assessment of the accuracy of the method, we trained the network to output the segmentation maps in a specific order, *central galaxy* first, and then the *companion*. The cost function (loss) used to train the modified U-Net is a *binary-crossentropy*, which performs well for a pixel-by-pixel binary classification as needed for our segmentation maps.

To score the results, the *binary-crossentropy* loss is not very informative since every pixel rightfully classified as background adds up to the accuracy, while we would like to assess the similarity to the target SExtractor segmentation map. For that purpose, we use a metric called Intersection over Union (IoU) also known as Jaccard index [132] of set A and B:

$$IoU(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{5.7}$$

It is usually defined in computer vision for bounding boxes,but can be adapted to any shape. This metric has the advantage of decreasing very rapidly to zero in case of a mismatch between A and B in terms of location or morphology. Therefore a score superior to 0.5 is considered a good score. After training, the U-Net obtains an average IoU score of 0.82, on the test images, which is an indication of a very good recovery. However, once the `blend2mask2flux` is trained end-to-end to recover the

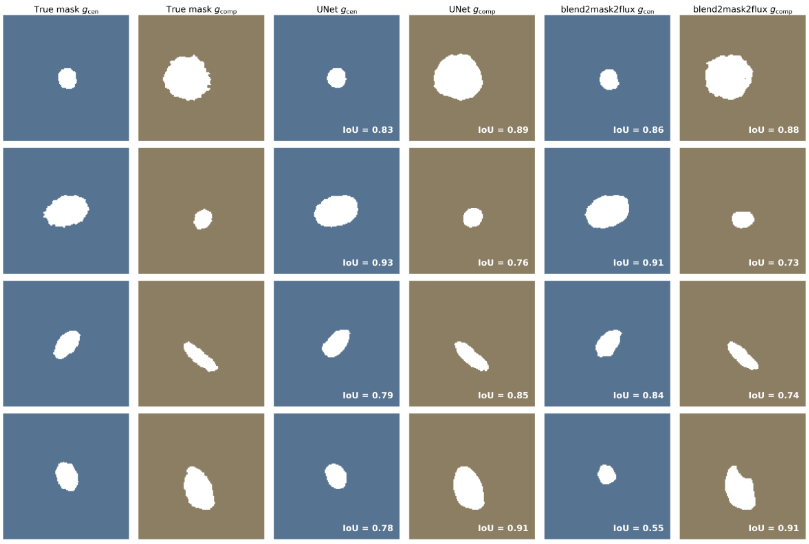

Figure 5.20. Same selection of four blend cases as in Figure 5.19 to compare this time only the results of the converged predicted segmentation masks of blended galaxies, yielded respectively by the U-Net architecture (centre) and the `blend2mask2flux` model (right), to the segmentation maps obtained from SExtractor on the individual galaxies. For each recovered galaxy mask, the segmentation score (IoU) as compared the SExtractor mask is indicated in the lower right corner

photometry, thus allowing the parameters of the U-Net section to vary, the average IoU score on the test data set drops to 0.70.

The outcome and evolution from the pure segmentation objective to the photometry objective can be seen in Figure 5.20, where the selection of blends is the same as the one on Figure 5.19, but the segmentation results of the initial U-Net are shown in the middle and can be directly compared with the one of the `blend2mask2flux` model on the right. The IoU computed on each image with respect to the SExtractor segmentation on the left is indicated on each image.

5.4 Summary and Conclusions

In this section, we summarize and briefly discuss the results obtained in this chapter, for both the presented works. We tested different approaches to handle the objects blending (see Sect. 5.2 and Sect. 5.3), using realistic datasets with images of two blended objects to provide an estimate of their performances when applied on Euclid-VIS images. The two works ran in parallel with a constant rate of feedbacks and information exchange. The machine learning approaches have been compared using the same baseline, but with different datasets and metrics. Both the works produced interesting results, which encourage us to further investigate the possibilities that the studied algorithms, and possible improvements, can provide. Using the tested UL approach for deblending granted the following results, measured using the metrics defined in Sect. 5.2.3:

- High overall performances ($\sim 82\% - 87\%$) with couples of simulated blended objects, slightly better than SExtractor (∼+4%-5%)

- Optimal overall performances ($\sim 100\%$) with single object datasets, both real and simulated ones, similar to SExtractor
- Optimal overall performances ($\sim 99\%$) for couples of large simulated blended objects, slightly better than SExtractor ($\sim +4\%$)
- High overall performances ($\sim 90\%$) with couples of large real blended objects, better deblending than SExtractor ($\sim +8\%$)
- Slightly less performance lost compared to SExtractor ($\sim -4\%$) when deblending large real blended objects compared to simulated ones

The results achieved demonstrate how the topological clustering approach outperforms the standard thresholding one. Performances, in terms of *deblending efficiency*, obtained on the 2-objects datasets reach, in the best scenario, $\sim$9% while in all the others it is never lower than $\sim 4\%$. The performance increments, compared to the standard method, provide then a new strategy to reduce systematics on cosmological/extragalactic measures. Using CNN approaches for deblending granted the following results:

- Unbiased photometry with typical uncertainty of ~ 0.1 magnitudes, implying an improvement in the flux error $\sim$4 times better than SExtractor
- Fraction of outliers is lower than $\sim 1\%$, $\sim$12 times better than SExtractor
- For large magnitudes differences, photometric uncertainties is ~ 0.2 magnitudes, SExtractor can reach 1 magnitude of difference
- Comparable photometric accuracy for all the morphological types explored
- The network optimizes photometry and segmentation maps generating tighter maps than SExtractor

The photometric measurement networks have demonstrated that on monochromatic images, they outperform traditional approaches with respect to photometric accuracy, precision, outliers fraction and stability towards different morphological types. Furthermore probabilistic maps are produced meaning that a better photometry than the direct mapping between the blend image to the flux measurement can be achieved, once again reducing systematics.

From the discussed results, we are motivated to promote machine learning techniques to handle objects blending. Both approaches provide significant improvements to deblending, compared to the standard approach deployed, enhancing the objects' correct separation and the relative fluxes reconstruction. These results are directly translated in a boost for all the measures that are heavily dependent on the correct flux (see Sect. 5.1), and therefore require an optimal deblending to be performed.

Chapter 6

Conclusion and future directions

In this thesis we have discussed the application of state-of-the-art image analysis techniques on astronomical images. We examined the improvements provided by such methods to simulated digital images aimed at reproducing next-generation extragalactic surveys data products, and to real images from recent deep sky surveys.

Next-generation imaging surveys aim to shed light on many deep questions about the Cosmos. Euclid, LSST, WFIRST and DES (taken as examples in this thesis, see Chapt. 2) will investigate fundamental questions about the nature of our Universe, its evolution and its constituents, studying the nature of Dark Energy and the distribution of Dark Matter. These surveys will provide unprecedented amounts of data, stored in digital images, which will be analyzed by complex software pipelines to extract scientific information. Billions of sources will be observed to yield information about weak lensing, galaxy clusters, supernovae Ia, and baryonic acoustic oscillations.

The technical features of the instrumentation and the observational integration time should represent, in principle, the only limitations for the quality of imaging surveys. However, in reality many sources of noise, intrinsically unavoidable in astronomical images, lower the signal-to-noise ratio, and the blending of sources creates systematic errors in photometric estimations. Thus noise and blending affect the quality of the data, and therefore directly impact the quality of the available scientific content.

To maximize the amount of valuable information, several image analysis techniques can be implemented, enhancing the quality of the images. In this work, we focused on techniques concerning the topics of "denoising" and "deblending", obtaining interesting results for the scientific community.

We investigated state-of-the-art denoising algorithms, vastly applied in other fields that heavily rely on digital images, applying them to a sample of astronomical extragalactic images. The tested methods represent various noise reduction families that have different performances dependent on the kind of image. We compared the algorithms (introduced in Sect. 3.1) with standard approaches used in astronomy, such as PSF filtering and wavelets. From the various tests, we identified a small number of techniques that provide exciting advancements (see Sect. 4.6). We demonstrated that Total-Variation techniques (ATVD-TVL2 and TV-Chambolle), Anisotropic Diffusion (Perona-Malik), and Bilateral filtering enhance faint galaxies detection (by definition, the most difficult to detect, and the less statistically represented) for images with high resolution and signal-to-noise ratio (such as images from space telescopes), without drastically altering shapes and fluxes of bright objects.

Furthermore, we investigated whether the applicatiton of machine learning techniques might adequately handle the blending of the sources in images simulating Euclid-VIS. In two different works, we compared Unsupervised learning (ASTErIsM-DENCLUE) and Convolutional Neural Networks (`blend2flux` and `blend2mask2flux`) to the standard multi-threshold approach (SExtractor), testing their ability to recover the blended object's correct fluxes and to properly separate them. From our tests (see Sect 5.2.4), the unsupervised learning algorithm provided higher performances in reconstructing the shapes of the objects than the multi-threshold approach ($\sim 4\% - 8\%$ better *deblending efficiency*), on various datasets composed of simulated and real galaxies. In the same way (see Sect. 5.3.4), both the Convolutional Neural Networks have demonstrated to recover the correct fluxes of blended objects properly, with a scatter of the distribution of the errors four times smaller compared to the multi-threshold approach, and drastically restricting the number of catastrophic misclassifications, $<1\%$. These results show how deblending algorithms tested, improve the shape and fluxes recovery, reducing systematic errors, and enhancing the analysis related to flux estimation, of extreme importance for optical/near-infrared surveys.

Future work will focus on further investigations and refinements. A conspicuous number of denoising algorithms are still to be tested in the astronomical scenario. Moreover, some machine learning techniques have been successfully applied to reduce noise in other fields, and they could, in principle, yield impressive results also on astronomical images. Furthermore, additional tests and multiple extra datasets can be considered to produce an even more detailed analysis of the algorithms. For what concerns deblending, the obtained results encourage us to compare the algorithms with a common metric. Unsupervised learning algorithms and convolutional neural networks work in an entirely different way, so a thorough comparison between these approaches is needed to analyze their performances on many levels. Moreover, additional datasets, reproducing blending for other surveys such as LSST, can be considered to study their behavior in different circumstances and to take advantage of the strengths provided by the two techniques.

In conclusion, we tested several image analysis algorithms on astronomical images, aiming to improve the amount of valuable information extracted. State-of-the-art and new emerging algorithms, never experimented in our field, proved to be a valuable asset for the astronomical images analysis, a solid alternative to the standard approaches.

Appendices

A MSE comparison tables and plots

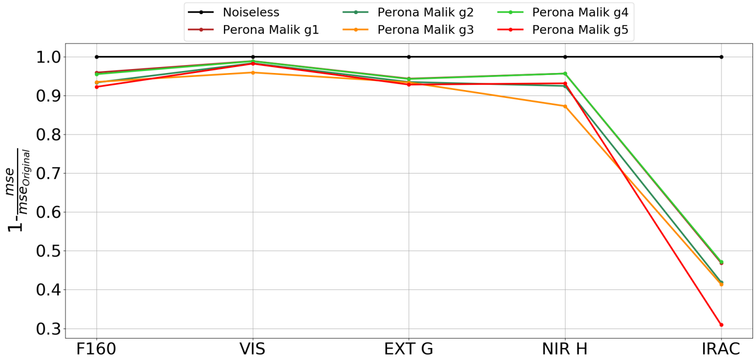

Figure A1. Step 1: MSE comparison between Perona Malik functions on CM. On the x-axis, all the simulated CM crops in the different bands, whereas on the y-axis $1 - \frac{mse}{mse_{Original}}$.

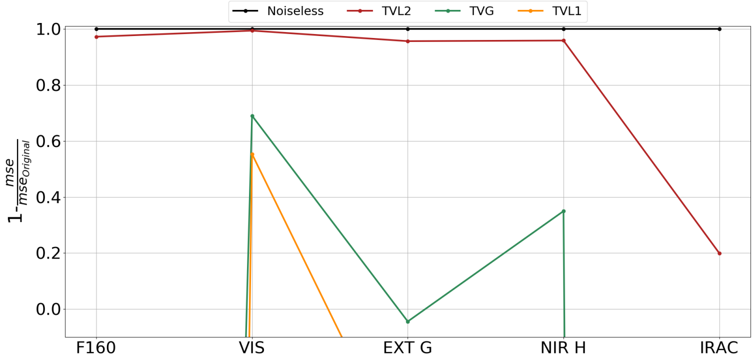

Figure A2. Step 1: MSE comparison between ATVD algorithms on BG. On the x-axis, all the simulated BG crops in the different bands, whereas on the y-axis $1 - \frac{mse}{mse_{Original}}$.

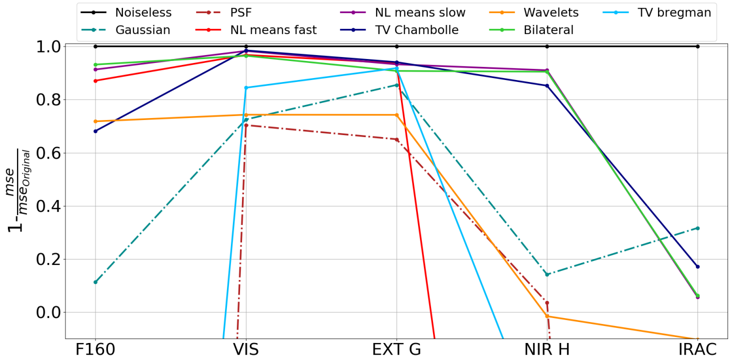

Figure A3. Step 1: MSE comparison between the other algorithms excluding ATVD and PM on CL. On the x-axis, all the simulated CL crops in the different bands, whereas on the y-axis $1 - \frac{mse}{mse_{Original}}$.

Name	MSE_{H160}	MSE_{VIS}	MSE_{EXTG}	MSE_{NIRH}	MSE_{IRAC}
TVL2	$\mathbf{5.320 \times 10^{-8}}$	$\mathbf{7.286 \times 10^{-7}}$	$\mathbf{3.672 \times 10^{-7}}$	$\mathbf{1.964 \times 10^{-4}}$	7.051×10^{-5}
TVL1	7.297×10^{-5}	5.155×10^{-5}	1.325×10^{-5}	4.409×10^{-2}	4.429×10^{-2}
TVG	3.922×10^{-5}	3.576×10^{-5}	8.760×10^{-6}	3.071×10^{-3}	6.370×10^{-3}
PM g=1 k=1×10^{-3}	9.490×10^{-8}	1.108×10^{-6}	6.776×10^{-7}	2.528×10^{-4}	5.473×10^{-5}
PM g=2 k=1×10^{-3}	9.790×10^{-8}	8.941×10^{-5}	8.147×10^{-6}	1.926×10^{-3}	7.073×10^{-5}
PM g=3 k=1×10^{-3}	9.930×10^{-8}	1.122×10^{-4}	8.390×10^{-6}	3.640×10^{-3}	7.855×10^{-5}
PM g=4 k=1×10^{-4}	7.960×10^{-8}	8.291×10^{-5}	7.641×10^{-6}	2.760×10^{-4}	6.288×10^{-5}
PM g=5 k=1×10^{-3}	1.077×10^{-7}	8.843×10^{-5}	8.131×10^{-6}	1.775×10^{-3}	7.157×10^{-5}
PSF	8.557×10^{-6}	1.833×10^{-5}	1.615×10^{-6}	3.492×10^{-3}	1.174×10^{-3}
Original	1.912×10^{-6}	1.154×10^{-4}	8.390×10^{-6}	4.722×10^{-3}	8.811×10^{-5}
TV Bregman	3.778×10^{-6}	5.908×10^{-6}	6.147×10^{-7}	1.960×10^{-3}	2.620×10^{-4}
Gaussian	1.301×10^{-6}	1.724×10^{-5}	9.434×10^{-7}	3.469×10^{-3}	6.955×10^{-5}
NL means slow	8.940×10^{-8}	1.513×10^{-6}	5.992×10^{-7}	3.066×10^{-4}	6.454×10^{-5}
NL means fast	1.019×10^{-7}	1.628×10^{-6}	5.750×10^{-7}	5.551×10^{-3}	1.389×10^{-4}
Bilateral	1.109×10^{-7}	3.754×10^{-6}	7.972×10^{-7}	4.612×10^{-4}	6.297×10^{-5}
TV Chambolle	1.876×10^{-7}	2.112×10^{-6}	4.914×10^{-7}	2.894×10^{-4}	$\mathbf{5.444 \times 10^{-5}}$
Wavelets	5.119×10^{-7}	2.962×10^{-5}	2.117×10^{-6}	4.776×10^{-3}	9.583×10^{-5}

Table A1. MSE table of BG crops. The lowest MSE value per band is indicated in bold

Name	MSE_{H160}	MSE_{VIS}	MSE_{EXTG}	MSE_{NIRH}	MSE_{IRAC}
TVL2	$\mathbf{5.590 \times 10^{-8}}$	$\mathbf{8.746 \times 10^{-7}}$	$\mathbf{3.268 \times 10^{-7}}$	$\mathbf{1.808 \times 10^{-4}}$	5.699×10^{-5}
TVL1	5.878×10^{-4}	1.075×10^{-3}	5.485×10^{-5}	2.390×10^{-1}	8.910×10^{-2}
TVG	2.977×10^{-5}	3.147×10^{-5}	4.477×10^{-5}	3.672×10^{-3}	5.752×10^{-3}
PM g=1 k=1×10^{-3}	1.248×10^{-7}	2.277×10^{-6}	4.646×10^{-7}	2.625×10^{-4}	$\mathbf{3.886 \times 10^{-5}}$
PM g=2 k=1×10^{-3}	1.257×10^{-7}	1.889×10^{-6}	4.310×10^{-6}	3.503×10^{-4}	4.969×10^{-5}
PM g=3 k=1×10^{-3}	1.228×10^{-7}	8.549×10^{-6}	7.173×10^{-6}	1.386×10^{-3}	5.755×10^{-5}
PM g=4 k=1×10^{-4}	8.470×10^{-8}	1.281×10^{-6}	3.914×10^{-6}	2.011×10^{-4}	4.697×10^{-5}
PM g=5 k=1×10^{-3}	1.468×10^{-7}	2.006×10^{-6}	8.287×10^{-6}	3.186×10^{-4}	5.055×10^{-5}
PSF	4.853×10^{-5}	1.532×10^{-4}	9.444×10^{-6}	6.024×10^{-3}	5.052×10^{-3}
Original	1.902×10^{-6}	1.116×10^{-4}	8.288×10^{-6}	4.679×10^{-3}	7.319×10^{-5}
TV Bregman	2.170×10^{-5}	1.084×10^{-4}	2.264×10^{-6}	1.023×10^{-2}	6.225×10^{-4}
Gaussian	1.835×10^{-6}	6.773×10^{-5}	1.913×10^{-6}	4.373×10^{-3}	7.094×10^{-5}
NL means slow	1.201×10^{-7}	1.883×10^{-6}	4.885×10^{-7}	7.916×10^{-4}	1.559×10^{-4}
NL means fast	1.990×10^{-7}	5.729×10^{-6}	4.655×10^{-7}	6.982×10^{-3}	1.814×10^{-4}
Bilateral	1.104×10^{-7}	4.027×10^{-6}	7.348×10^{-7}	3.630×10^{-4}	4.779×10^{-5}
TV Chambolle	4.964×10^{-7}	1.698×10^{-6}	4.723×10^{-7}	6.008×10^{-4}	5.398×10^{-5}
Wavelets	5.303×10^{-7}	2.999×10^{-5}	2.128×10^{-6}	4.732×10^{-3}	7.795×10^{-5}

Table A2. MSE table of CM crops. The lowest MSE value per band is indicated in bold

Name	MSE_{H160}	MSE_{VIS}	MSE_{EXTG}	MSE_{NIRH}	MSE_{IRAC}
TVL2	$\mathbf{7.070 \times 10^{-8}}$	$\mathbf{8.958 \times 10^{-7}}$	$\mathbf{3.443 \times 10^{-7}}$	$\mathbf{2.451 \times 10^{-4}}$	3.633×10^{-4}
TVL1	2.754×10^{-4}	1.707×10^{-4}	1.471×10^{-5}	1.121×10^{-1}	5.820×10^{-2}
TVG	4.424×10^{-5}	3.336×10^{-5}	1.387×10^{-5}	4.519×10^{-3}	1.013×10^{-2}
PM g=1 k=1×10^{-3}	1.463×10^{-7}	1.752×10^{-6}	5.923×10^{-7}	3.433×10^{-4}	3.312×10^{-4}
PM g=2 k=1×10^{-3}	1.437×10^{-7}	1.865×10^{-5}	7.553×10^{-6}	4.436×10^{-4}	3.433×10^{-4}
PM g=3 k=1×10^{-3}	1.431×10^{-7}	6.034×10^{-5}	8.342×10^{-6}	1.413×10^{-3}	3.501×10^{-4}
PM g=4 k=1×10^{-4}	1.029×10^{-7}	1.076×10^{-5}	6.809×10^{-6}	2.893×10^{-4}	3.397×10^{-4}
PM g=5 k=1×10^{-3}	1.668×10^{-7}	1.432×10^{-5}	7.471×10^{-6}	4.163×10^{-4}	3.439×10^{-4}
PSF	2.728×10^{-5}	3.421×10^{-5}	2.913×10^{-6}	4.576×10^{-3}	2.952×10^{-3}
Original	1.915×10^{-6}	1.155×10^{-4}	8.342×10^{-6}	4.748×10^{-3}	3.640×10^{-4}
TV Bregman	1.233×10^{-5}	1.795×10^{-5}	6.815×10^{-7}	7.211×10^{-3}	6.886×10^{-4}
Gaussian	1.699×10^{-6}	3.175×10^{-5}	1.211×10^{-6}	4.074×10^{-3}	$\mathbf{2.487 \times 10^{-4}}$
NL means slow	1.664×10^{-7}	1.976×10^{-6}	5.627×10^{-7}	4.264×10^{-4}	3.432×10^{-4}
NL means fast	2.470×10^{-7}	3.702×10^{-6}	5.260×10^{-7}	1.975×10^{-2}	4.581×10^{-4}
Bilateral	1.307×10^{-7}	4.055×10^{-6}	7.697×10^{-7}	4.521×10^{-4}	3.413×10^{-4}
TV Chambolle	6.098×10^{-7}	1.740×10^{-6}	4.938×10^{-7}	7.015×10^{-4}	3.017×10^{-4}
Wavelets	5.399×10^{-7}	2.968×10^{-5}	2.149×10^{-6}	4.821×10^{-3}	4.016×10^{-4}

Table A3. MSE table of CL crops. The lowest MSE value per band is indicated in bold

Name	$T_{H160}[s]$	$T_{VIS}[s]$	$T_{EXTG}[s]$	$T_{NIRH}[s]$	$T_{IRAC}[s]$
TVL2	9.047	3.588	0.688	0.132	0.0938
TVL1	5.408	0.814	0.339	0.020	0.0031
TVG	8.910	5.177	0.334	0.134	0.0914
PM g=1 k=1×10^{-3}	6.353	5.283	2.127	0.362	0.063
PM g=2 k=1×10^{-3}	10.519	22.826	0.248	8.347	0.159
PM g=3 k=1×10^{-3}	6.847	30.578	12.136	5.544	0.142
PM g=4 k=1×10^{-4}	88.506	323.720	109-356	50.417	6.512
PM g=5 k=1×10^{-3}	6.778	16.001	51.941	11.955	0.125
PSF	0.071	0.019	0.008	0.008	0.001
Original	n.a.	n.a.	n.a.	n.a.	n.a.
TV Bregman	0.222	0.093	0.021	0.025	0.011
Gaussian	**0.055**	**0.017**	**0.005**	**0.002**	**0.001**
NL means slow	75.94	27.75	6.863	3.344	1.058
NL means fast	7.514	3.104	1.118	0.448	0.208
Bilateral	37.09	13.46	3.602	1.567	0.489
TV Chambolle	10.61	0.7668	0.109	0.587	0.034
Wavelets	0.827	0.329	0.091	0.038	0.012

Table A4. CPU Time table of CM crops .The lowest time value per band is indicated in bold

B PSF and Depth comparison plots

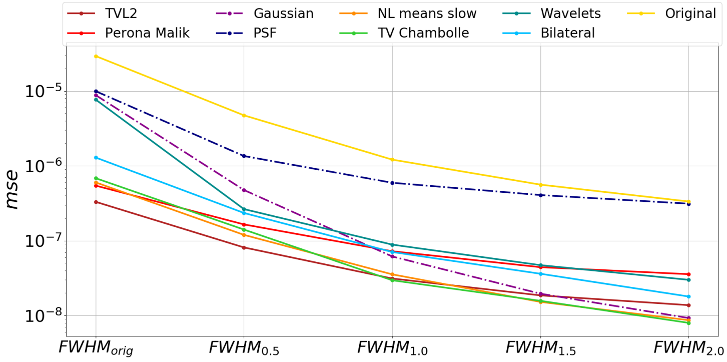

Figure B1. VIS FWHM variation comparison plot. On the x-axis the VIS images with FWHM equal to the original value, 0.5, 1.0, 1.5 and 2.0 arcsecs, whereas on the y-axis mse.

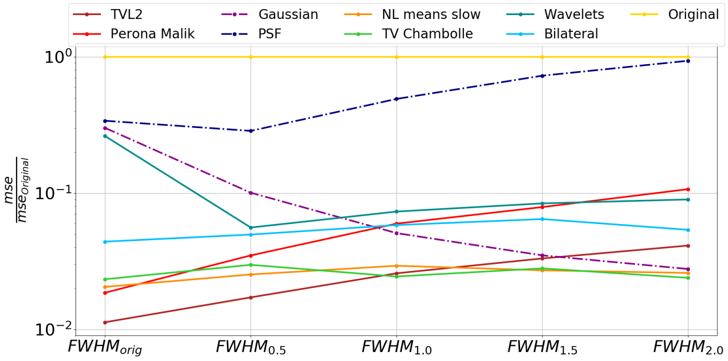

Figure B2. VIS FWHM variation comparison plot. On the x-axis the VIS images with FWHM equal to the original value, 0.5, 1.0, 1.5 and 2.0 arcsecs, whereas on the y-axis $\frac{mse}{mse_{Original}}$.

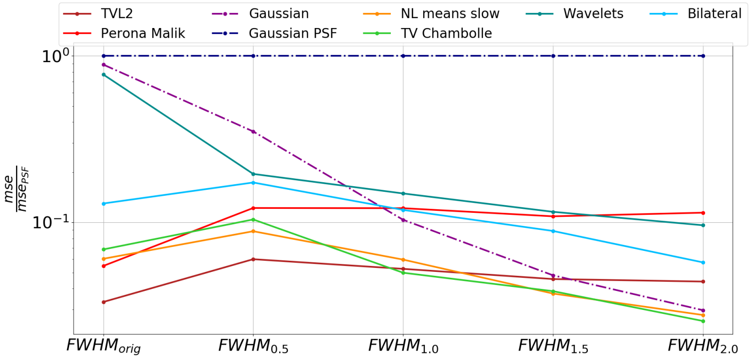

Figure B3. VIS FWHM variation comparison plot. On the x-axis the VIS images with FWHM equal to the original value, 0.5, 1.0, 1.5 and 2.0 arcsecs, whereas on the y-axis $\frac{mse}{mse_{PSF}}$.

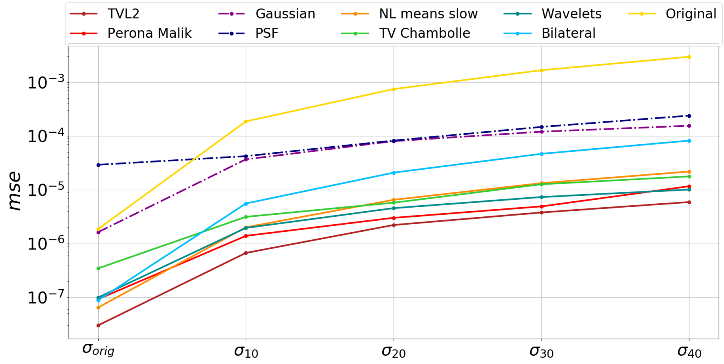

Figure B4. H160 depth variation comparison plot. On the x-axis the H160 images with Gaussian noise standard deviation equal to 1, 10, 20, 30 and 40 times the original value, whereas on the y-axis mse.

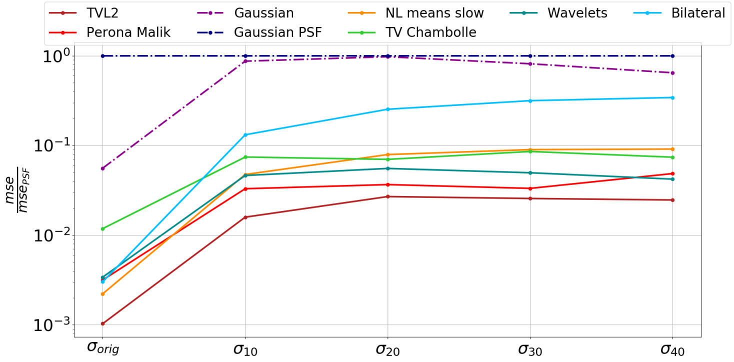

Figure B5. H160 depth variation comparison plot. On the x-axis the H160 images with Gaussian noise standard deviation equal to 1, 10, 20, 30 and 40 times the original value, whereas on the y-axis $\frac{mse}{mse_{PSF}}$.

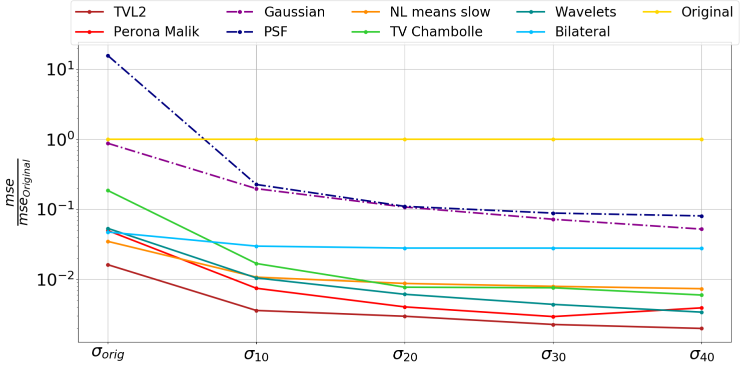

Figure B6. H160 depth variation comparison plot. On the x-axis the H160 images with Gaussian noise standard deviation equal to 1, 10, 20, 30 and 40 times the original value, whereas on the y-axis $\frac{mse}{mse_{Original}}$.

C VIS Crops visual comparison

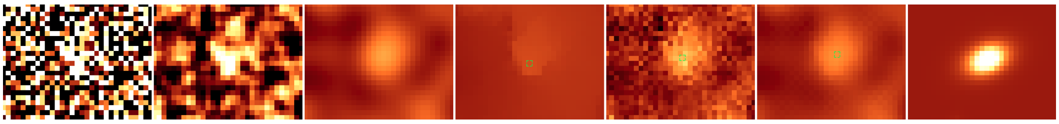

Figure C1. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 38.8 with magnitude of 25.79

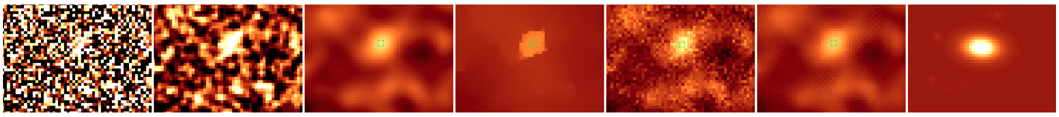

Figure C2. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 48.2 with magnitude of 24.76

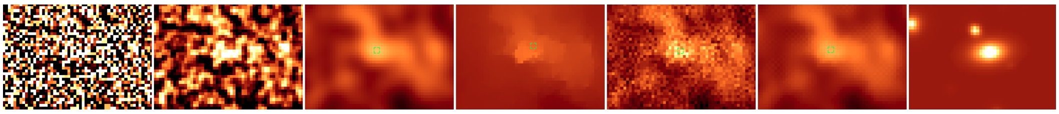

Figure C3. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 72.9 with magnitude of 23.82

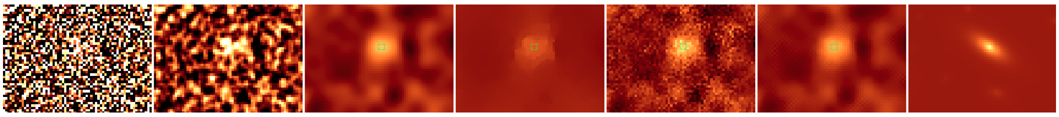

Figure C4. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 47.5 with magnitude of 25.01

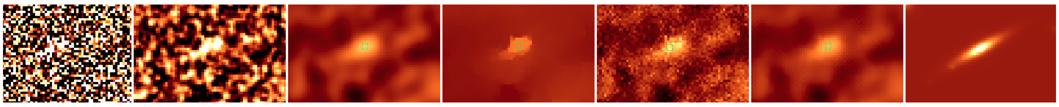

Figure C5. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 35.44 with magnitude of 25.39

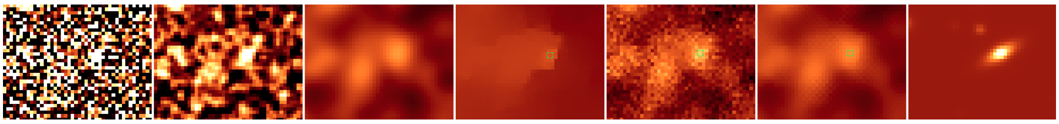

Figure C6. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 21.26 with magnitude of 26.48

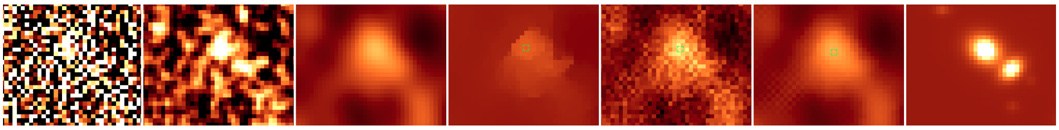

Figure C7. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 27.70 with magnitude of 25.72

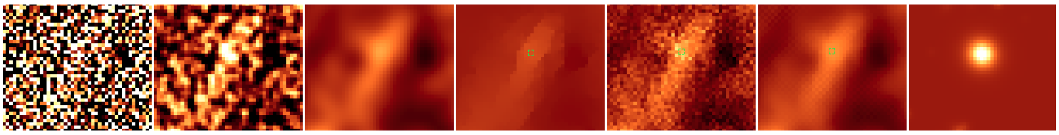

Figure C8. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 45.57 with magnitude of 24.97

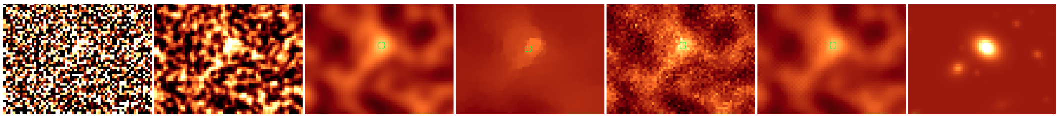

Figure C9. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 68.29 with magnitude of 24.14

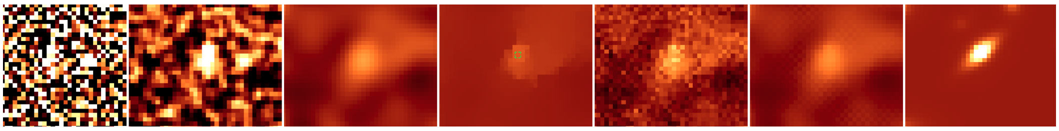

Figure C10. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 26.74 with magnitude of 26.13

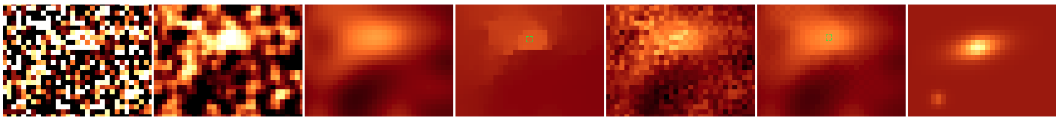

Figure C11. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 25.44 with magnitude of 26.19

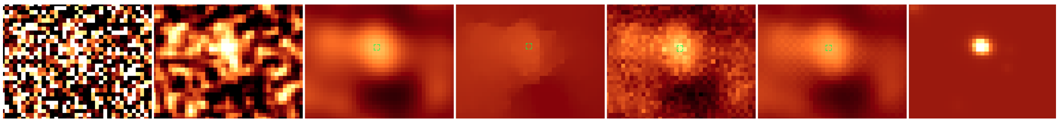

Figure C12. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 36.99 with magnitude of 25.71

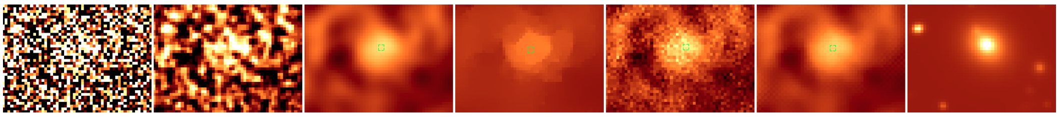

Figure C13. VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 90.35 with magnitude of 23.34

D GSDEEP Crops visual comparison

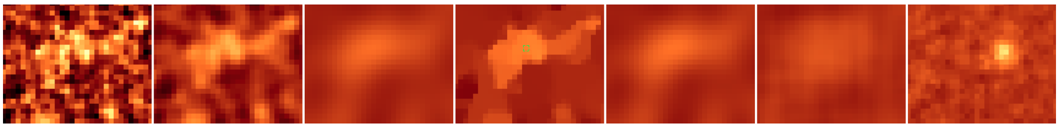

Figure D1. GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 6.71 with magnitude of 27.47

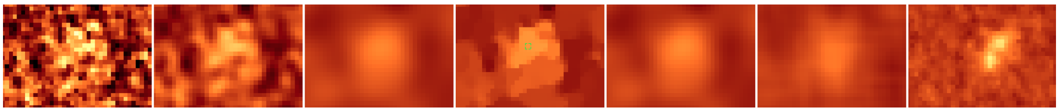

Figure D2. GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 7.37 with magnitude of 27.20

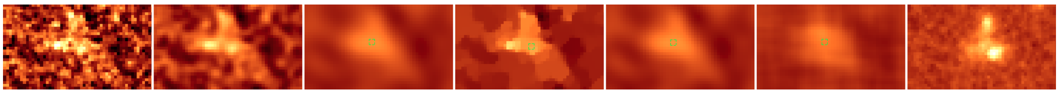

Figure D3. GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 8.44 with magnitude of 26.80

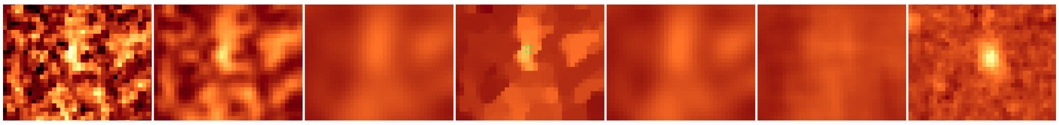

Figure D4. GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 6.88 with magnitude of 27.18

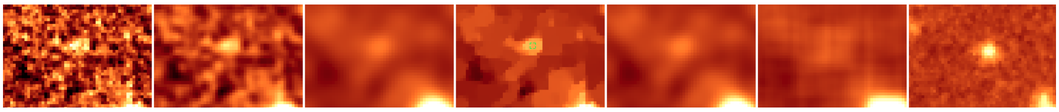

Figure D5. GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 5.82 with magnitude of 27.48

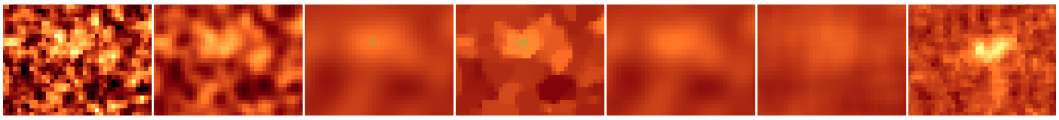

Figure D6. GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 5.08 with magnitude of 27.48

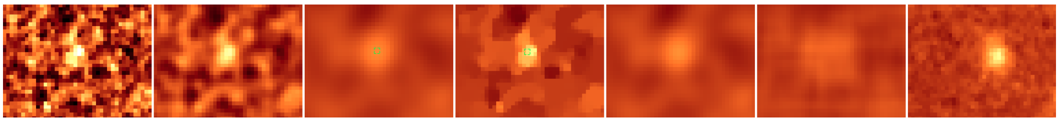

Figure D7. GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 6.51 with magnitude of 27.58

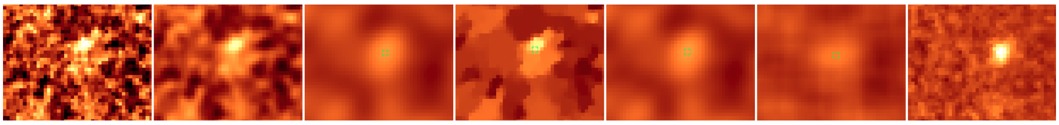

Figure D8. GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 12.48 with magnitude of 27.17

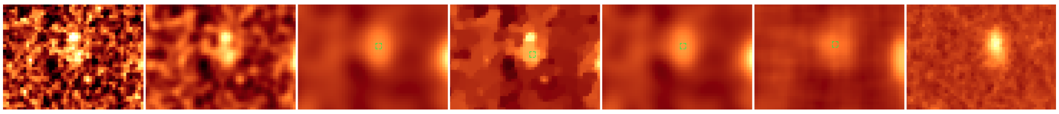

Figure D9. GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 9.77 with magnitude of 27.36

Bibliography

[1] The dark energy survey. https://www.darkenergysurvey.org/the-des-project/survey-and-operations/.

[2] Euclid, a space mission to map the dark universe. https://www.cosmos.esa.int/web/euclid.

[3] Lsst data management. https://www.lsst.org/about/dm.

[4] Sloan digital sky surveys. https://www.sdss.org/surveys/. https://www.sdss.org/surveys/.

[5] The Invention of the Telescope and the Foundation of Modern Astronomy. https://www.juliantrubin.com/bigten/telescope_invention.html.

[6] Rakesh Agrawal and Paul Stolorz, editors. *KDD'98: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1998.

[7] Mihael Ankerst, Markus M. Breunig, Hans peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. pages 49–60. ACM Press, 1999.

[8] Archaeology. The world's oldest writing. https://www.archaeology.org/issues/213-1605/features/4326-cuneiform-the-world-s-oldest-writing, 2016.

[9] Jean-François Aujol, Guy Gilboa, Tony Chan, and Stanley Osher. Structure-texture image decomposition—modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, Apr 2006.

[10] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv e-prints*, page arXiv:1511.00561, Nov 2015.

[11] Nicholas M. Ball and Robert J. Brunner. Data Mining and Machine Learning in Astronomy. *International Journal of Modern Physics D*, 19(7):1049–1106, Jan 2010.

[12] F. Banterle, M. Corsini, P. Cignoni, and R. Scopigno. A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain. *Computer Graphics Forum*, 31(1):19–32, 2012.

[13] Kyle Barbary. Sep: Source extractor as a library. *The Journal of Open Source Software*, 1(6):58, 10 2016.

[14] Hamish Barwick. Ska telescope to generate more data than entire internet in 2020. https://www.computerworld.com.au/article/392735/ska_telescope_generate_more_data_than_entire_internet_2020/.

[15] W. A. Baum. Problems of extra-galactic research. *p. 390, IAU Symposium No. 15*, 1962.

[16] N. Berggren, J. & Sidoli. Aristarchus's On the Sizes and Distances of the Sun and the Moon: Greek and Arabic Texts. 2007.

[17] Charl Botha Bernhard Preim. *Image Analysis for Medical Visualization*. 2014.

[18] E. Bertin. SkyMaker: astronomical image simulations made easy. *Mem. Societa Astronomica Italiana*, 80:422, 2009.

[19] E. Bertin and S. Arnouts. SExtractor: Software for source extraction. *Astronomy & Astrophysics Supplement Series*, 117:393–404, June 1996.

[20] M. Bethermin, Herve Dole, Morgane Cousin, and N. Bavouzet. Fastphot: A simple and quick idl psf-fitting routine. *Astrophysics Source Code Library*, pages 02008–, 02 2013.

[21] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[22] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, March 1998.

[23] T. V. P. Bliss and G. L. Collingridge. A synaptic model of memory: long-term potentiation in the hippocampus. *Nature*, 361:31–39, January 1993.

[24] Abhishake Kumar Bojja, Franziska Mueller, Sri Raghu Malireddi, Markus Oberweger, Vincent Lepetit, Christian Theobalt, Kwang Moo Yi, and Andrea Tagliasacchi. HandSeg: An Automatically Labeled Dataset for Hand Segmentation from Depth Images. *arXiv e-prints*, page arXiv:1711.05944, Nov 2017.

[25] Kirk Born. Top 10 big data challenges – a serious look at 10 big data v's. https://mapr.com/blog/top-10-big-data-challenges-serious-look-10-big-data-vs/.

[26] Alexandre Boucaud et al. Photometry of high-redshift blended galaxies using deep learning. *arXiv e-prints*, page arXiv:1905.01324, May 2019.

[27] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65 vol. 2, June 2005.

[28] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-Local Means Denoising. *Image Processing On Line*, 1:208–212, 2011.

[29] Colin J. Burke, Patrick D. Aleo, Yu-Ching Chen, Xin Liu, John R. Peterson, Glenn H. Sembroski, and Joshua Yao-Yu Lin. Deblending and Classifying Astronomical Sources with Mask R-CNN Deep Learning. *arXiv e-prints*, page arXiv:1908.02748, Aug 2019.

[30] Jacqueline Bush. Bregman algorithms. Master's thesis, University of California, Santa Barbara, 2011. http://www.math.ucsb.edu/~cgarcia/UGProjects/BregmanAlgorithms_JacquelineBush.pdf.

[31] M. Castellano, A. Fontana, K. Boutsia, A. Grazian, L. Pentericci, R. Bouwens, M. Dickinson, M. Giavalisco, P. Santini, S. Cristiani, F. Fiore, S. Gallozzi, E. Giallongo, R. Maiolino, F. Mannucci, N. Menci, A. Moorwood, M. Nonino, D. Paris, A. Renzini, P. Rosati, S. Salimbeni, V. Testa, and E. Vanzella. Evidence of a fast evolution of the UV luminosity function beyond redshift 6 from a deep HAWK-I survey of the GOODS-S field. *Astronomy and Astrophysics*, 511:A20, February 2010.

[32] M. Castellano, D. Ottaviani, A. Fontana, E. Merlin, S. Pilo, and M. Falcone. Improving Resolution and Depth of Astronomical Observations via Modern Mathematical Methods for Image Analysis. In A. R. Taylor and E. Rosolowsky, editors, *Astronomical Data Analysis Software and Systems XXIV (ADASS XXIV)*, volume 495 of *Astronomical Society of the Pacific Conference Series*, pages 257–260, 2015.

[33] IERS ICRS Center. The icrf. http://hpiers.obspm.fr/icrs-pc/newwww/icrf/index.php, 2018.

[34] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, Jan 2004.

[35] S. G. Chang, Bin Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, Sept 2000.

[36] Li Chen, H.D. Cheng, and Jianping Zhang. Fuzzy subfiber and its application to seismic lithology classification. *Information Sciences - Applications*, 1:77–95, 03 1994.

[37] D. H. Clark and M. D. H. Clark. *Measuring the cosmos : how scientists discovered the dimensions of the universe*. 2004.

[38] Astropy Collaboration. Astropy: A community Python package for astronomy. October 2013.

[39] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[40] Euclid Consortium. Ground segment. https://www.euclid-ec.org/?page_id=2625.

[41] Kirkby D. Cosmic shear measurements in deep surveys. presentation to lsst desc weak lensing telecon., 2013.

[42] J. Darbon, A. Cunha, T. F. Chan, S. Osher, and G. J. Jensen. Fast nonlocal filtering applied to electron cryomicroscopy. In *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1331–1334, May 2008.

[43] Dark Energy Survey Collaboration. The Dark Energy Survey: more than dark energy - an overview. *MNRAS*, 460(2):1270–1299, Aug 2016.

[44] W. A. Dawson and M. D. Schneider. Complementarity of LSST and WFIRST: Regarding object blending. 2 2014.

[45] William A. Dawson, Michael D. Schneider, J. Anthony Tyson, and M. James Jee. THE ELLIPTICITY DISTRIBUTION OF AMBIGUOUSLY BLENDED OBJECTS. *The Astrophysical Journal*, 816(1):11, dec 2015.

[46] C. De Santis, A. Grazian, A. Fontana, and P. Santini. ConvPhot: A profile-matching algorithm for precision photometry. *New Astronomy*, 12:271–288, February 2007.

[47] Richard DeWitt. *An Introduction to the History and Philosophy of Science*. Worldviews, 2010.

[48] Paola Dimauro et al. A catalog of polychromatic bulge-disc decompositions of ∼17.600 galaxies in CANDELS. *MNRAS*, 478(4):5410–5426, Aug 2018.

[49] David L. Donoho and Jain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[50] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[51] B Everitt. *Finite Mixture Distributions*. 1981.

[52] Skrondal A. Everitt B.S. *Cambridge Dictionary of Statistics*. Cambridge University Press, 2010.

[53] Kitty Ferguson. The glassmaker who sparked astrophysics. http://nautil.us/issue/11/light/the-glassmaker-who-sparked-astrophysics, 2014.

[54] A. Fontana, J. S. Dunlop, D. Paris, T. A. Targett, K. Boutsia, M. Castellano, A. Galametz, A. Grazian, R. McLure, E. Merlin, L. Pentericci, S. Wuyts, O. Almaini, K. Caputi, R.-R. Chary, M. Cirasuolo, C. J. Conselice, A. Cooray, E. Daddi, M. Dickinson, S. M. Faber, G. Fazio, H. C. Ferguson, E. Giallongo, M. Giavalisco, N. A. Grogin, N. Hathi, A. M. Koekemoer, D. C. Koo, R. A. Lucas, M. Nonino, H. W. Rix, A. Renzini, D. Rosario, P. Santini, C. Scarlata, V. Sommariva, D. P. Stark, A. van der Wel, E. Vanzella, V. Wild, H. Yan, and S. Zibetti. The Hawk-I UDS and GOODS Survey (HUGS): Survey design and

deep K-band number counts. *Astronomy and Astrophysics*, 570:A11, October 2014.

[55] George Forbes. *History of Astronomy, Illustrated, George Forbes 1909 First Edition.* Watts & Co, London, 1909.

[56] C.S. Foster, M.; Sherrington. *Textbook of Physiology.* , volume 3 (7th ed.), London: Macmillan. p. 929., 1897.

[57] freeCodeCamp. An intuitive guide to Convolutional Neural Networks, 2018.

[58] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:2007, 2007.

[59] I. Furenlid. A comparison of CCD's and photographic plates. *American Astronomical Society*, Issue 36(1984/2):5–8, 1984.

[60] D. Gabor. Information theory in electron microscopy. *Laboratory Investigation*, 14(6):801 – 807, 1965.

[61] A. Galametz et al. CANDELS Multiwavelength Catalogs: Source Identification and Photometry in the CANDELS UKIDSS Ultra-deep Survey Field. *The Astrophysical Journal Supplement*, 206:10, June 2013.

[62] Adam Gauci, Kristian Zarb Adami, and John Abela. Machine Learning for Galaxy Morphology Classification. *arXiv e-prints*, page arXiv:1005.0390, May 2010.

[63] Pascal Getreuer. Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman. *Image Processing On Line*, 2:74–95, 2012.

[64] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[65] Sanjay Churiwala Gopinath Rebala, Ajay Ravi. *An Introduction to Machine Learning.* Springer, 2019.

[66] Norman A. Grogin et al. CANDELS: The Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey. *ApJ*, 197(2):35, Dec 2011.

[67] James E. Gunn et al. The 2.5 m Telescope of the Sloan Digital Sky Survey. *The Astronomical Journal*, 131(4):2332–2359, Apr 2006.

[68] Z. Guo, J. Sun, D. Zhang, and B. Wu. Adaptive perona–malik model based on the variable exponent for image denoising. *IEEE Transactions on Image Processing*, 21(3):958–967, March 2012.

[69] Yicheng Guo et al. CANDELS Multi-wavelength Catalogs: Source Detection and Photometry in the GOODS-South Field. *The Astrophysical Journal Supplement Series*, 207(2):24, Aug 2013.

[70] R.; Mahowald M. A.; Douglas R. J.; Seung H. S. Hahnloser, R.; Sarpeshkar. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405:947–951, 2000.

[71] Hartlap, J., Hilbert, S., Schneider, P., and Hildebrandt, H. A bias in cosmic shear from galaxy selection: results from ray-tracing simulations. *A&A*, 528:A51, 2011.

[72] B. Häussler, D. H. McIntosh, M. Barden, E. F. Bell, H.-W. Rix, A. Borch, S. V. W. Beckwith, J. A. R. Caldwell, C. Heymans, K. Jahnke, S. Jogee, S. E. Koposov, K. Meisenheimer, S. F. Sánchez, R. S. Somerville, L. Wisotzki, and C. Wolf. GEMS: Galaxy Fitting Catalogs and Testing Parametric Galaxy Fitting Codes: GALFIT and GIM2D. *The Astrophysical Journal Supplement Series*, 172:615–633, October 2007.

[73] Martin Hilbert. *Big Data for Development: A Review of Promises and Challenges. Development Policy Review.* martinhilbert.net, 2015.

[74] Alexander Hinneburg and Hans-Henning Gabriel. Denclue 2.0: Fast clustering based on kernel density estimation. In Michael R. Berthold, John Shawe-Taylor, and Nada Lavrač, editors, *Advances in Intelligent Data Analysis VII*, pages 70–80, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[75] B. W. Holwerda. Source Extractor for Dummies v5. *arXiv e-prints*, pages astro–ph/0512139, Dec 2005.

[76] Steve B. Howell. *Handbook of CCD Astronomy*. Cambridge Observing Handbooks for Research Astronomers. Cambridge University Press, 2 edition, 2006.

[77] E. P. Hubble. Cepheids in spiral nebulae. *The Observatory*, 48:139–142, May 1925.

[78] M. Huertas-Company, R. Gravet, G. Cabrera-Vives, P. G. Pérez-González, J. S. Kartaltepe, G. Barro, M. Bernardi, S. Mei, F. Shankar, P. Dimauro, E. F. Bell, D. Kocevski, D. C. Koo, S. M. Faber, and D. H. Mcintosh. A Catalog of Visual-like Morphologies in the 5 CANDELS Fields Using Deep Learning. *The Astrophysical Journal Supplement Series*, 221(1):8, Nov 2015.

[79] IERS / IVS. The second realization of the international celestial reference frame by very long baseline interferometry, 2009.

[80] B. Jain et al. The Whole is Greater than the Sum of the Parts: Optimizing the Joint Science Return from LSST, Euclid and WFIRST. *arXiv e-prints*, page arXiv:1501.07897, Jan 2015.

[81] H. L. Johnson and W. W. Morgan. Fundamental stellar photometry for standards of spectral type on the revised system of the Yerkes spectral atlas. *ApJ*, 117:313, May 1953.

[82] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. [Online; accessed <today>].

[83] Joseph, R., Courbin, F., and Starck, J.-L. Multi-band morpho-spectral component analysis deblending tool (muscadet): Deblending colourful objects. *A&A*, 589:A2, 2016.

[84] Shilpa Kamdi and R K Krishna. Image segmentation and region growing algorithm. *Int. J. Comput. Technol.*, 2, 01 2012.

[85] Jeyhan S. Kartaltepe et al. CANDELS Visual Classifications: Scheme, Data Release, and First Results. *Astrophysical Journal, Supplement*, 221(1):11, Nov 2015.

[86] Henry C. King. *The History of the Telescope.* Dover Pubns, 1979.

[87] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, Dec 2014.

[88] Hazan Elad Kingma, John and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, page 12: 2121–2159, 2011.

[89] Zador A. Koch C. The function of dendritic spines: devices subserving biochemical rather than electrical compartmentalization. *Journal of Clinical Neurophysiology*, 13 (2):413–422, 1993.

[90] Anton M. Koekemoer et al. CANDELS: The Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey—The Hubble Space Telescope Observations, Imaging Data Products, and Mosaics. *ApJ*, 197(2):36, Dec 2011.

[91] A. M. Kolmogorov and S. V. Fomin. *Elements of the theory of functions and functional analysis.* Dover Publications, 1957.

[92] D. C. Koo. Optical multicolors : a poor person's Z machine for galaxies. *AJ*, 90:418–440, Mar 1985.

[93] R. G. Kron. Photometry of a complete sample of faint galaxies. *Astrophysical Journal Supplement Series*, 43:305–325, June 1980.

[94] Jeffrey Kruk. WFIRST: Project overview and status. https://wfirst.ipac.caltech.edu/mtgs/AAS231/Kruk_355.39_v2.pdf.

[95] V. G. Laidler, C. Papovich, N. A. Grogin, R. Idzi, M. Dickinson, H. C. Ferguson, B. Hilbert, K. Clubb, and S. Ravindranath. TFIT: A Photometry Package Using Prior Information for Mixed-Resolution Data Sets. *Publications of the Astronomical Society of the Pacific*, 119:1325–1344, November 2007.

[96] Doug Laney. Application delivery strategies. *META Group*, 2001.

[97] Laureijs et al. Euclid Definition Study Report. *arXiv e-prints*, page arXiv:1110.3193, Oct 2011.

[98] H. S. Leavitt and E. C. Pickering. Periods of 25 Variable Stars in the Small Magellanic Cloud. *Harvard College Observatory Circular*, 173:1–3, March 1912.

[99] G. R. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.

[100] Abbé G. Lemaître. A Homogeneous Universe of Constant Mass and Increasing Radius accounting for the Radial Velocity of Extra-galactic Nebulæ. *Monthly Notices of the Royal Astronomical Society*, 91(5):483–490, 03 1931.

[101] M Lindenbaum, M. Fischer, and A. Bruckstein. On gabor's contribution to image enhancement. *Pattern Recognition*, 27(1):1 – 8, 1994.

[102] M. Livio and A. G. Riess. Measuring the Hubble constant. *Physics Today*, 66(10):41, 2013.

[103] Steve Lohr. The origins of 'big data': An etymological detective story. https://bits.blogs.nytimes.com/2013/02/01/the-origins-of-big-data-an-etymological-detective-story/.

[104] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *arXiv e-prints*, page arXiv:1411.4038, Nov 2014.

[105] LSST Science Collaboration. LSST Science Book, Version 2.0. *arXiv e-prints*, page arXiv:0912.0201, Dec 2009.

[106] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[107] Piero Madau, Henry C. Ferguson, Mark E. Dickinson, Mauro Giavalisco, Charles C. Steidel, and Andrew Fruchter. High-redshift galaxies in the Hubble Deep Field: colour selection and star formation history to z$\sim$4. *MNRAS*, 283(4):1388–1404, Dec 1996.

[108] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 11(7):674–693, july 2001.

[109] Rachel Mandelbaum. Weak Lensing for Precision Cosmology. *Annual Review of Astronomy and Astrophysics*, 56:393–433, Sep 2018.

[110] Songrit Maneewongvatana and David M. Mount. Analysis of approximate nearest neighbor searching with clustered point sets. *CoRR*, cs.CG/9901013, 1999.

[111] Bashir ZI Massey PV. Long-term depression: multiple forms and implications for brain function. *Trends in Neurosciences*, 30 (4):176–84, 2007.

[112] Ian S. McLean. *Electronic Imaging in Astronomy*. Springer, 2 edition, 2008.

[113] P. Melchior, F. Moolekamp, M. Jerdee, R. Armstrong, A.-L. Sun, J. Bosch, and R. Lupton. scarlet: Source separation in multi-band images by constrained matrix factorization. *Astronomy and Computing*, 24:129–142, Jul 2018.

[114] E. Merlin, N. Bourne, M. Castellano, H. C. Ferguson, T. Wang, S. Derriere, J. S. Dunlop, D. Elbaz, and A. Fontana. T-phot version 2.0: Improved algorithms for background subtraction, local convolution, kernel registration, and new options. *Astronomy & Astrophysics*, 595:A97, Nov 2016.

[115] E. Merlin, A. Fontana, H. C. Ferguson, J. S. Dunlop, D. Elbaz, N. Bourne, V. A. Bruce, F. Buitrago, M. Castellano, C. Schreiber, A. Grazian, R. J. McLure, K. Okumura, X. Shu, T. Wang, R. Amorín, K. Boutsia, N. Cappelluti, A. Comastri, S. Derriere, S. M. Faber, and P. Santini. T-PHOT: A new code for PSF-matched, prior-based, multiwavelength extragalactic deconfusion photometry. *Astronomy & Astrophysics*, 582:A15, Oct 2015.

[116] Y. Meyer. *Ondelettes et opérateurs: Ondelettes*. Hermann, 1990.

[117] E. D. Miner. *Book review: Uranus, The Planet, Rings and Satellites, by Ellis D. Miner (2nd edition)*. The Journal of the British Astronomical Association, 1998.

[118] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1998.

[119] Simon Mitton. *The Crab Nebula*. Faber & Faber; First U.S. Edition, 1979.

[120] Hossein Mobahi, Shankar R. Rao, Allen Y. Yang, S. Shankar Sastry, and Yi Ma. Computer vision, 2001.

[121] A. F. J. Moffat. A Theoretical Investigation of Focal Stellar Images in the Photographic Emulsion and Application to Photographic Photometry. *Astronomy & Astrophysics*, 3:455, December 1969.

[122] David B. Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[123] NASA. Eclipses and the saros. https://web.archive.org/web/20071030225501/http://sunearth.gsfc.nasa.gov/eclipse/SEsaros/SEsaros.html, 2007.

[124] J. Neyman and E. S. Pearson. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 231 (694706)::289337, 1933.

[125] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[126] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning Deconvolution Network for Semantic Segmentation. *arXiv e-prints*, page arXiv:1505.04366, May 2015.

[127] United States Naval Observatory. International celestial reference system (icrs). https://aa.usno.navy.mil/faq/docs/ICRS_doc.php, 2018.

[128] School of Mathematics and Scotland Statistics, University of St Andrews. Hipparchus of rhodes. http://www-groups.dcs.st-and.ac.uk/~history/Biographies/Hipparchus.html, 2007.

[129] J. B. Oke. Absolute Spectral Energy Distributions for White Dwarfs. *ApJ*, 27:21, February 1974.

[130] J. B. Oke and J. E. Gunn. Secondary standard stars for absolute spectrophotometry. *ApJ*, 266:713–717, March 1983.

[131] Stanley Osher and James A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[132] Jaccard P. Étude comparative de la distribution florale dans une portion des alpes et des jura, 1901.

[133] Rutecki PA. Neuronal excitability: voltage-dependent currents and synaptic transmission. *Journal of Clinical Neurophysiology*, 9 (2):195–211, 1992.

[134] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space, November 1901.

[135] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[136] P. J. E. Peebles. *The large-scale structure of the universe*. 1980.

[137] R. Pello, J. M. Miralles, J. F. Le Borgne, J. P. Picat, G. Soucail, and G. Bruzual. Identification of a high redshift cluster. in the field of Q2345+007 through deep BRIJK' photometry. *A&A*, 314:73–86, Oct 1996.

[138] A. A. Penzias and R. W. Wilson. A Measurement of Excess Antenna Temperature at 4080 Mc/s. *ApJ*, 142:419–421, July 1965.

[139] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, Jul 1990.

[140] K. Price D. Raj Reddy R. Ohlander. Computer graphics and image processing, 1978.

[141] M. Jordan Raddick, Georgia Bracey, Pamela L. Gay, Chris J. Lintott, Phil Murray, Kevin Schawinski, Alexander S. Szalay, and Jan Vandenberg. Galaxy Zoo: Exploring the Motivations of Citizen Science Volunteers. *Astronomy Education Review*, 9(1):010103, Jan 2010.

[142] Jason Rhodes et al. Scientific Synergy between LSST and Euclid. *The Astrophysical Journal Supplement Series*, 233(2):21, Dec 2017.

[143] A. G. Riess et al. Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant. *The Astronomical Journal*, 116:1009–1038, September 1998.

[144] Lior Rokach and Oded Maimon. *Clustering methods. Data mining and knowledge discovery handbook.* Springer US, 2005.

[145] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv e-prints*, page arXiv:1505.04597, May 2015.

[146] B. T. P. Rowe et al. GALSIM: The modular galaxy image simulation toolkit. *Astronomy and Computing*, 10:121–150, April 2015.

[147] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259 – 268, 1992.

[148] E. S. Rykoff et al. redMaPPer. I. Algorithm and SDSS DR8 Catalog. *The Astrophysical Journal*, 785(2):104, Apr 2014.

[149] Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, pages 71–105, 1959.

[150] C. Schreiber, D. Elbaz, M. Pannella, E. Merlin, M. Castellano, A. Fontana, N. Bourne, K. Boutsia, F. Cullen, J. Dunlop, H. C. Ferguson, M. J. Michałowski, K. Okumura, P. Santini, X. W. Shu, T. Wang, and C. White. EGG: hatching a mock Universe from empirical prescriptions⋆. *Astronomy and Astrophysics*, 602:A96, Jun 2017.

[151] J. L. Sérsic. Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy. *Boletin de la Asociacion Argentina de Astronomia La Plata Argentina*, 6:41, 1963.

[152] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *National Academy of Sciencies, Proceeding*, 93(14):1591–1595, 1995.

[153] Linda G. Shapiro and George C. Stockman. Segmentation of natural images by texture and boundary compression, 2011.

[154] R. E. Skelton et al. 3D-HST WFC3-selected Photometric Catalogs in the Five CANDELS/3D-HST Fields: Photometry, Photometric Redshifts, and Stellar Masses. *The Astrophysical Journal Supplement*, 214:24, October 2014.

[155] W. Boyle G. Smith. U.S. patent US3796927A. https://patents.google.com/patent/US3796927, 1969.

[156] D. Spergel et al. Wide-Field InfrarRed Survey Telescope-Astrophysics Focused Telescope Assets WFIRST-AFTA 2015 Report. *arXiv e-prints*, page arXiv:1503.03757, Mar 2015.

[157] Geoffrey E. Hinton Alexander Krizhevsky Ilya Sutskever Nitish Srivastva. System and method for addressing overfitting in a neural network. 2013.

[158] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, Nov 1992.

[159] Charles C. Steidel, Mauro Giavalisco, Mark Dickinson, and Kurt L. Adelberger. Spectroscopy of Lyman Break Galaxies in the Hubble Deep Field. *AJ*, 112:352, Aug 1996.

[160] Robert L. Thorndike. *Psychometrika.* Springer-Verlag, 1953.

[161] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, Jan 1998.

[162] A. Tramacere, D. Paraficz, P. Dubath, J. P. Kneib, and F. Courbin. ASTErIsM: application of topometric clustering algorithms in automatic galaxy detection and classification. *MNRAS*, 463(3):2939–2957, Dec 2016.

[163] Sik-Ho Tsang. Review: Fcn — fully convolutional network (semantic segmentation). https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1.

[164] Scott E Umbaugh. *Digital image processing and analysis : human and computer vision applications with CVIPtools.* (2nd edition) CRC Press, 2010.

[165] Rice University. Satellites of Jupiter, The Galileo Project, 1995.

[166] C. Valens. A really friendly guide to wavelets, 1999.

[167] Peter Bühlmann Sara van de Geer. *Statistics for High-Dimensional Data.* Springer, 2011.

[168] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, June 2014.

[169] A. van der Wel, E. F. Bell, B. Häussler, E. J. McGrath, Yu-Yen Chang, Yicheng Guo, D. H. McIntosh, H. W. Rix, M. Barden, E. Cheung, S. M. Faber, H. C. Ferguson, A. Galametz, N. A. Grogin, W. Hartley, J. S. Kartaltepe, D. D. Kocevski, A. M. Koekemoer, J. Lotz, M. Mozena, M. A. Peth, and Chien Y. Peng. Structural Parameters of Galaxies in CANDELS. *Astrophysical Journal, Supplement*, 203(2):24, Dec 2012.

[170] E. C. Vasconcellos, R. R. de Carvalho, R. R. Gal, F. L. LaBarbera, H. V. Capelato, H. Frago Campos Velho, M. Trevisan, and R. S. R. Ruiz. Decision Tree Classifiers for Star/Galaxy Separation. *The Astronomical Journal*, 141(6):189, Jun 2011.

[171] M. Vetterli and J. Kovačevic. *Wavelets and Subband Coding*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[172] Robert M. Wald. *General Relativity*. University Of Chicago Press, first edition edition, 1984.

[173] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[174] Joachim Weickert. *Anisotropic diffusion in image processing*. Teubner, 1998.

[175] D. C. Wells, E. W. Greisen, and R. H. Harten. FITS - a Flexible Image Transport System. *AAPS*, 44:363, June 1981.

[176] Rafael C. Gonzalez; Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2007.

[177] F. Y. Wu. The potts model. *Rev. Mod. Phys.*, 54:235–268, Jan 1982.

[178] Antoine Bordes Xavier Glorot and Yoshua Bengio. Deep sparse rectifier neural networks. 2011.

[179] Bengio Y. *Practical Recommendations for Gradient-Based Training of Deep Architectures*. Springer, Berlin, Heidelberg, 2012.

[180] Y. Yao and A. Caponnetto. Constr approx. 2007.

[181] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, pages 103–114, New York, NY, USA, 1996. ACM.

[182] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2):103–114, June 1996.

[183] Yanxia Zhang and Yongheng Zhao. Astronomy in the big data era. *Data Science Journal*, 14:1–9, 05 2015.