Alma Mater Studiorum - Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea Magistrale in Astrofisica e Cosmologia

Characterization of Convolutional Neural Networks for the identification of Galaxy-Galaxy Strong Lensing events

TESI DI LAUREA MAGISTRALE

Presentata da:
Laura Leuzzi

Relatore:
Chiar.mo Prof.
Massimo Meneghetti

Correlatore:
Giuseppe Angora

Appello IV
Anno Accademico 2019-2020

Sommario

L'osservazione di eventi di lensing forte tra galassie consente diversi tipi di studi, che spaziano dalla ricostruzione della distribuzione di massa delle galassie che li producono alla determinazione della costante di Hubble. Le future survey fotometriche, come quella che verrà effettuata dal telescopio spaziale Euclid, contribuiranno alla scoperta di migliaia di questi sistemi, che dovranno però essere ricercati tra miliardi di sorgenti osservate. In questo contesto, è evidente l'importanza di sviluppare tecniche automatiche e affidabili per l'analisi veloce di grandi volumi di dati. Le Reti Neurali Convoluzionali sono una tecnica di Deep Learning che si è particolarmente affermata negli ultimi anni come potente mezzo di indagine in questo campo, grazie alla loro velocità di esecuzione e capacità di generalizzazione. In questo lavoro di tesi, in particolare, abbiamo esaminato l'abilità delle reti di individuare eventi di questo tipo in base alle loro caratteristiche morfologiche, confrontando l'efficacia di tre diverse architetture. Per farlo, abbiamo utilizzato due dataset, costituiti di immagini simulate per riprodurre la qualità delle osservazioni attesa dal telescopio Euclid: il primo dataset contiene eventi di morfologia variegata e complessa, mentre il secondo è principalmente caratterizzato da lenti chiare. Nello specifico, abbiamo sia valutato la prestazione delle reti su diverse selezioni di dati, contenenti frazioni crescenti di oggetti di difficile classificazione, che verificato la loro capacità di identificare lenti ovvie. Abbiamo inoltre indagato l'impatto di alcune delle principali caratteristiche di lenti e sorgenti sui risultati ottenuti. La nostra analisi ha confermato le potenzialità delle reti neurali per l'identificazione di lenti ovvie, mentre ha messo in evidenza la necessità di training specifici per l'individuazione di lenti meno caratteristiche.

Abstract

Studying Galaxy-Galaxy Strong Lensing events allows to tackle several problems, that include the reconstruction of the mass distribution of the lens galaxies and the estimation of the Hubble constant. Thousands of these systems are expected to be detected in upcoming imaging surveys, such as the one that will be carried out by the Euclid space telescope, but they will have to be identified among the billions of sources that will be observed. In this context, the development of automated and reliable techniques for the examination of large volumes of data is of crucial importance. Convolutional Neural Networks are a Deep Learning technique that has proven particularly effective in the past years as a poweful tool for the analysis of large datasets, because of their speed of execution and capacity of generalization. In this thesis work, in particular, we evaluate the ability of this kind of Neural Networks to identify these events on the basis of their morphological characteristics, comparing the performance of three different architectures. For this purpose, we have used two datasets, composed of images simulated to mimic the data quality expected by the observations of the Euclid space telescope: the lenses in the first dataset are characterized by a diverse and complex morphology, while the lenses in the second one are mainly recognizable because of large arcs and rings. Specifically, we have evaluated the performance of the networks on different selections of images, gradually including larger fractions of borderline objects, as well as their ability to identify the most evident lenses. Moreover, we have investigated the possible impact of some of the main characteristics of the lenses and sources on our results. Our analysis has confirmed the potential of the application of this method for the identification of clear lenses, while it has highlighted the need of specific training for the detection of fainter lensing features.

Contents

List of Figures

List of Tables

Introduction

Galaxy-galaxy strong lensing (GGSL) events occur when a background source falls within the caustic of a foreground galaxy. Finding and studying these events is very important for several reasons. Recent studies showed that GGSL events in the core of some massive clusters are relatively abundant (a few per cluster). Their number exceeds the expectations from numerical simulations in the ΛCDM cosmology context by about one order of magnitude (Meneghetti et al., 2020). These puzzling results may signal potential undiagnosed systematic issues with simulations or incorrect assumptions about Dark Matter properties. Confirming these results by finding more events and studying how the probability of GGSL depends on several cluster properties may help to understand this issue. Besides, GGSL events provide constraints for recovering the mass distribution of both the lensing galaxies and their host galaxy clusters (see e.g. Bergamini et al. 2019). Studying the interplay between Dark Matter and baryons by mapping their spatial distributions on different scales is essential for understanding the processes that shape the growth and the evolution of the cosmic structures.

Upcoming large-scale imaging surveys from space, such as the one that will be carried out by the Euclid mission, can potentially increase the number of known strong lensing galaxies in clusters by several orders of magnitude. For this to happen, images of tens of millions of galaxies will need to be inspected to identify potential GGSL candidates.

One of the most common techniques for the identification of GGSL events has been the visual inspection of images after a pre-selection of potential candidates based on their color and luminosity (see e.g. Jackson 2008; Pawase et al. 2014). Moreover, semi-automated algorithms that detect lenses depending on the elongation of arc-shaped or ring-shaped features have also been developed (see e.g. Seidel & Bartelmann 2007; Gavazzi et al. 2014).

However, these methods are not fast enough to analyze large data sets and flexible enough to detect diverse lens configurations. In this context, Deep Learning (DL) techniques are a viable option, since they automatize the problem of feature extraction and guarantee a fast analysis of great amounts of images. In fact, it is expected that they will play a key role in the future of astronomical data analysis methodologies. Convolutional Neural Networks (CNNs), in particular, are able to learn directly from a training set the features that will be the most relevant for the classification of images in different categories.

In the past few years, several works have proven the potential of CNNs for detecting GGSL events in survey data. Lanusse et al. (2018), for example, apply this method

on a dataset of LSST-like[1] mock observations, obtaining 90% of completeness for the lenses with Einstein radii larger than 1.43". Petrillo et al. (2019) train and test their models on a dataset of real galaxies from the Kilo Degree Survey (KiDS, de Jong et al. 2015) and simulated lenses, recovering the 75% of the confirmed lenses in the catalog of observations. Jacobs et al. (2019) search for strong lenses at high redshift in the Dark Energy Survey (DES, The Dark Energy Survey Collaboration 2005), finding 84 new candidates after having trained the networks on simulated and real images.
In this work, we investigate the ability of CNNs to identify GGSL events depending on the observed morphology. We implement three popular CNN architectures, inspired by the VGG Network (Simonyan & Zisserman, 2015), the Inception Networks (Szegedy et al., 2015, 2016), and the Residual Networks (He et al., 2016; Xie et al., 2017). These models have been employed in image classification problems with great success in the past few years and have become a benchmark for the scientific community.
We train and test our models on two datasets, simulated to mimic the data quality expected from the Euclid space mission. The first one consists of 100000 images, simulated for the Strong Gravitational Lens Finding Challenge (Metcalf et al., 2019), while the second one consists of 20000 images. Both these datasets are divided into two classes, according to whether they show lensing characteristics or not.
The main difference between these two datasets is that, while in the case of the first one morphological and photometric criteria are applied to distinguish between the images of the two categories, in the case of the second one the division is straightforward and determined by the presence or absence of a background source.
In particular, we investigate how the presence of borderline images in the training sets affects the performance of our models, and how the properties of the simulated lenses and sources impact on the purity and completeness of the recovered catalogues.
This work is organized as follows:

1. In Chapter 1 we introduce the fundamentals of modern Cosmology, with a special focus on the physical laws that regulate the expansion of the Universe (Section 1.1). Moreover, we introduce the Theory of Structure Formation (Section 1.2), that investigates the evolution of the primordial density fluctuations into the structures observed at the present time. Finally, we characterize in greater detail the clusters of galaxies (Section 1.3), that are the most massive virialized structures in the Universe: because of the high density of their central regions, they are the ideal environment for the observation of Strong Lensing events.

2. In Chapter 2 we introduce the Theory of Gravitational Lensing, defining some of the fundamental quantities and equations used for the description and comprehension of this phenomenon. In particular, in Section 2.1 we define the deflection angle, the lens equation, the lensing potential, the time delay surface and we discuss the magnification and distortion of background sources. In Section 2.2 we outline the main properties of some commonly employed

[1]Large Synoptic Survey Telescope (LSST Science Collaboration et al., 2009).

lens models. Finally, in Section 2.3 we take into consideration the specific phenomenology of Gravitational Lensing in galaxy clusters.

3. In Chapter 3 we outline the basics of supervised Machine Learning Theory. We firstly focus on the description of the structure and training process of Neural Networks (Section 3.2). Afterwards, we discuss in greater detail the advantages and applications of Convolutional Neural Networks (Section 3.4), the type of models investigated in this work. We pay specific attention to the architectures that inspired the models implemented in this work, presenting them in Sections 3.4.1, 3.4.2 and 3.4.3 .

4. In Chapter 4 we present the datasets used to train, validate and test our models. In particular, we introduce the Lens Finding Challenge dataset in Section 4.1, while we describe the Euclid VIS dataset in Section 4.2. We describe the simulations and discuss the main properties of the images, as well as the most relevant characteristics of the galaxy sources and lenses. In Section 4.3 we discuss the main differences between the two datasets.

5. In Chapter 5 we describe the preprocessing steps applied to the images (Section 5.1) before using them in the experiments and we characterize the metrics employed for the evaluation of the performance of our models (Section 5.2). The greatest part of this Chapter is, however, devoted to the delineation of the setup of the experiments and and on the discussion of the results obtained. We illustrate the experiments conducted with the Lens Finding Challenge dataset in Sections 5.3 and 5.4, while we focus on the tests carried on the Euclid VIS dataset in Section 5.5.

6. In Chapter 6 we summarize the most important results and discuss the future perspectives and extensions of our work (Section 6.1).

7. In Appendix A we give an overview of the models examined in this work. In particular, in Section A.1 we describe the models implemented, while in Section A.2 we outline the setup of the training procedure.

8. In Appendix B we describe a procedure to simulate GGSL events in galaxy clusters, that is currently under implementation.

Chapter 1

Fundamentals of Cosmology

1.1 Cosmological background

Cosmology is the branch of Astrophysics involved in the study of the formation and evolution of the Universe on large scales.
The model currently used to describe it is the Standard Model of Cosmology, which is based on two strong and fundamental assumptions:

1. *Cosmological Principle*
   This is the assertion that at sufficiently large scales (that today are circa hundreds of Mpc[1]) the Universe is homogeneous and isotropic. Homogeneity is the property of being identical everywhere in space, while isotropy is the property of looking the same in every direction.
   Observational evidences that the Universe is characterised by both these properties derive from the Cosmic Microwave Background[2] (CMB) and the distribution of galaxies on Mpc scales.

2. *Gravity is described by General Relativity*
   Gravity is the most relevant interaction when describing large scales, and it is so far best described by Einstein's Theory of General Relativity. According to this theory, the Universe's content of matter-energy determines its geometrical properties, i.e. its curvature, and gravity is a manifestation of these properties.

In General Relativity, the Einstein equations relate the geometry of space-time, represented by the metric tensor $g_{\mu\nu}$, to the matter content of the Universe, described by the energy-momentum tensor $T_{\mu\nu}$

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu} + \Lambda g_{\mu\nu} \tag{1.1}$$

where Λ is the cosmological constant, c is the speed of light, G is the universal gravitational constant and R and $R_{\mu\nu}$ are the Ricci scalar and tensor, respectively. The indices $\mu, \nu = 0, 1, 2, 3$ represent one time coordinate and three spatial coordinates.

[1] $1\ Mpc = 3.086 \cdot 10^{24}\ cm$

[2] The CMB, first observed in 1965 (Penzias, A. A. & Wilson, R. W., 1965), is the cooled remnant of the early phases of the Big Bang. It has a perfect black body spectrum at $T \sim 2.73K$, uniform over the whole sky (Longair, S. M., 2011)

It should be noted that the presence of the term $\Lambda g_{\mu\nu}$ is not strictly necessary from a mathematical point of view, but it has been added for physical reasons that will be discussed later in this Section.
Any metric tensor that represents a cosmological model must incorporate homogeneity and isotropy. Under these circumstances, it can be shown (Schutz, 2009) that the distance between two points in space-time, expressed as $ds^2 = g_{\mu\nu}(x)dx^\mu dx^\nu$ in its general form in General Relativity, can be reduced to the form

$$ds^2 = c^2dt^2 - a^2(t)\left[\frac{dr^2}{1-kr^2} + r^2(d\theta^2 + \sin^2\theta d\phi^2)\right] \tag{1.2}$$

in spherical polar coordinates r (dimensionless by convention) θ, ϕ. Here t is the proper time; $a(t)$ is the *scale factor*, a function to be determined which has the dimensions of a length; k is the *curvature parameter*, a constant whose value can only be 1, 0, -1, which is related to the Universe's geometry.
It can be proven (Coles & Lucchin, 2002) that:

- if $k = 0$, equation (1.2) describes a flat space, whose geometry is Euclidean;
- if $k = 1$, equation (1.2) describes a closed space whose geometry is hyperspherical;
- if $k = -1$, equation (1.2) describes an open Universe, whose geometry is hyperbolic.

The metric (1.2) is called the *Robertson-Walker metric*.

1.1.1 The expansion of the Universe

The Hubble law

The *proper distance* between two points P and P_0, which for simplicity is taken to define the origin of a set of polar coordinates r, θ, ϕ (we can take $d\theta = d\phi = 0$), is defined as the distance measured by a chain of rulers held by observers between the two points, at the same cosmic time t ($dt = 0$).
From equation (1.2) we can define it as

$$d_P = \int_0^r \frac{a(t)dr'}{(1-kr'^2)^{1/2}} = a(t)f(r) \tag{1.3}$$

where the function $f(r)$ is k-dependent and is given by:

$$f(r) = \begin{cases} \sin^{-1}(r) & \text{for} \quad k = 1, \\ r & \text{for} \quad k = 0, \\ \sinh^{-1}(r) & \text{for} \quad k = -1. \end{cases} \tag{1.4}$$

The *comoving distance* is defined as the proper distance evaluated at the present time t_0:

$$d_C = d_P(t_0) = a(t_0)f(r) = \frac{a(t_0)}{a(t)}d_P(t). \tag{1.5}$$

The proper distance of point P might change with time because of the time-dependence of the scale factor a. So, one can calculate its time derivative and define a radial velocity between two points as

$$v_R = \frac{dd_{PR}}{dt} = \dot{a} f(r) = \frac{\dot{a}}{a} d_{PR}. \tag{1.6}$$

Equation (1.6) is the *Hubble law*. It states that two points depart from each other at a relative velocity proportional to their relative distance. Moreover

$$H(t) = \frac{\dot{a}(t)}{a(t)} \tag{1.7}$$

is called the *Hubble parameter*, and it has the dimensions of inverse time.
It can be proven (Coles & Lucchin, 2002) that the Hubble law is directly derivable from the Cosmological Principle.

Redshift

In the spectra of distant and luminous sources a shift is observed between the wavelength at which the photons have been emitted (λ_e) at time t and that at which they are observed (λ_0) at time t_0. This difference is described by a parameter called *redshift*:

$$z = \frac{\lambda_0 - \lambda_e}{\lambda_e}. \tag{1.8}$$

It can be shown (Coles & Lucchin, 2002) that z and a are related through

$$1 + z = \frac{a(t_0)}{a(t)}. \tag{1.9}$$

This means that the redshift of the signal is related to the distance between two objects. In addition, the observed values of z are positive, so we can conclude that $a(t_0) > a(t)$, which means that the Universe is expanding, namely $\dot{a} > 0$.

The Deceleration Parameter

The scale factor $a(t)$ is a function of time t. We can expand it for values of t close to t_0 in a power series:

$$a(t) = a(t_0) \left[1 + H_0(t - t_0) - \frac{1}{2} q(t_0) H_0^2 (t - t_0)^2 + ... \right] \tag{1.10}$$

where

$$q(t_0) = q_0 = -\frac{\ddot{a}(t_0) a(t_0)}{\dot{a}^2(t_0)} \tag{1.11}$$

is the *deceleration parameter* at $t = t_0$, a dimensionless quantity that is related to the second derivative of the scale factor and hence describes the acceleration of the expansion.

1.1.2 Cosmological distances

The definitions given in equations (1.3) and (1.5) have little operational significance, since it is not possible to measure simultaneously all the distance elements between us and some astronomical object. However, one can define operationally other kinds of distance which can, at least in principle, be measured directly.

- **Luminosity distance** d_L.
  This definition aims to preserve the Euclidean inverse-square law for the diminution of luminosity with distance from a point source:

$$d_L \equiv \left(\frac{L}{4\pi F}\right)^{1/2}, \tag{1.12}$$

  where L is the luminosity of the source at distance r, emitting light at time t, while F is the flux measured by the observer at time t_0.

  Several effects need to be taken into account. Because of the expansion of the Universe, the signal is redshifted by a factor $a(t)/a(t_0)$. Due to the time-dilation effect (Schutz, 2009) time intervals between the photons' arrivals are enlarged of another factor $a(t)/a(t_0)$. Also, the area of a spherical surface centred on the source and passing through the observer at time t_0 is $4\pi a(t_0)^2 r^2$. So we obtain:

$$F = \frac{L}{4\pi a^2(t_0) r^2}\left(\frac{a(t)}{a(t_0)}\right)^2 = \frac{L}{4\pi a^2(t_0) r^2}(1+z)^{-2}, \tag{1.13}$$

  from which

$$d_L = a(t_0) r (1+z). \tag{1.14}$$

- **Angular diameter distance** d_A
  This definition aims to preserve a geometrical property of the Euclidean space, namely the variation of the angular size of an object with its distance from the observer:

$$d_A = \frac{D_P}{\Delta\theta} = a(t) r \tag{1.15}$$

  where $D_P = a(t) r \Delta\theta$ is the proper diameter of the source at distance r at time t, and $\Delta\theta$ is the angle subtended by D_P.

  From equation (1.15) follows that the luminosity distance and the angular diameter distance are related through

$$d_A = d_L \frac{a^2(t)}{a^2(t_0)} = \frac{d_L}{(1+z)^2}. \tag{1.16}$$

Since the Universe is not flat and static, these definitions do not coincide in general, so it is not possible to define length distances in a unique way. Nevertheless, when considering sources in the local Universe, namely at $z \sim 0$, the effects of curvature and expansion are negligible and they are equivalent.

1.1.3 Friedmann equations

The Friedmann equations can be directly derived from Equations (1.1) under two conditions:

1. The Universe is homogeneous and isotropic, as stated by the Cosmological Principle and assumed in the Robertson-Walker metric in Equation (1.2);
2. The Universe is considered a perfect fluid, so its energy-momentum tensor is

$$T_{\mu\nu} = -Pg_{\mu\nu} + (P + \rho c^2)u_\mu u_\nu, \tag{1.17}$$

where u_μ is the fluid four-velocity, ρc^2 is its energy density and P represents its pressure.

At the beginning of the past century, the Universe was widely accepted to be static, but this was not possible according to the Einstein equations as they were originally formulated (Coles & Lucchin, 2002). For this reason, Einstein modified the equations, by introducing the *cosmological constant* Λ, whose value can be appropriately chosen to obtain a static cosmological model.
For this reason, a new term was added to the tensor (1.17), defining the *effective energy-momentum tensor*:

$$\widetilde{T_{\mu\nu}} \equiv T_{\mu\nu} + \frac{\Lambda c^4}{8\pi G} g_{\mu\nu} = (\tilde{\rho}c^2 + \tilde{P})u_\mu u_\nu - \tilde{P}g_{\mu\nu}, \tag{1.18}$$

where

$$\begin{cases} \tilde{\rho} = \rho + \rho_\Lambda = \rho + \frac{\Lambda c^2}{8\pi G} & \text{is the effective energy density} \\ \tilde{P} = P + P_\Lambda = P - \frac{\Lambda c^4}{8\pi G} & \text{is the effective pressure} \end{cases}. \tag{1.19}$$

P_Λ and ρ_Λ hence represent the modifications to the pressure and density of the perfect fluid due to the introduction of Λ.
With the discovery of the expansion of the Universe, Λ was removed from the equations. After the observation of the Universe's accelerated expansion (Riess, A. G. et al., 1998; Perlmutter, S. et al., 1999), however, it was re-introduced to physically justify the observations, as we will clarify later in the Section.
Inserting Equations (1.2) and (1.18) in (1.1), one finds the first and the second *Friedmann equations*, respectively:

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3}\left(\rho + \frac{3P}{c^2}\right) + \frac{\Lambda c^2}{3} = -\frac{4\pi G}{3}\left(\tilde{\rho} + \frac{3\tilde{P}}{c^2}\right), \tag{1.20a}$$

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho - \frac{kc^2}{a^2} + \frac{\Lambda c^2}{3} = \frac{8\pi G}{3}\tilde{\rho} - \frac{kc^2}{a^2}. \tag{1.20b}$$

In reality, these equations are not independent: the second one can be recovered from the first one, if one takes into account the adiabatic expansion of the Universe, namely

$$d(\rho c^2 a^3) = -Pda^3 \tag{1.21}$$

that represents the condition of energy conservation in the Universe.
Equation (1.20b) can be rewritten in terms of the Hubble parameter. At $t = t_0$ we obtain

$$H_0^2 \left(1 - \frac{\Lambda c^2}{3H_0^2} - \frac{8\pi G\rho_0}{3H_0^2}\right) = -\frac{kc^2}{a_0^2} \tag{1.22}$$

with the notation $x_0 = x(t = t_0)$ for the generic function x. Equation 1.22 can be reformulated as

$$H_0^2 \left(1 - \frac{\rho_{\Lambda,0}}{\rho_{c,0}} - \frac{\rho_0}{\rho_{c,0}}\right) = -\frac{kc^2}{a_0^2} \tag{1.23}$$

once the *critical density* of the Universe has been defined as

$$\rho_{c,0} = \frac{3H_0^2}{8\pi G}. \tag{1.24}$$

We can introduce the *density parameter* as $\Omega_{i,0} = \frac{\rho_{i,0}}{\rho_{c,0}}$ for the i-th component of the Universe.
So equation (1.23) can be rewritten in the form

$$H_0^2(1 - \Omega_{\Lambda,0} - \Omega_0) = -\frac{kc^2}{a_0^2}. \tag{1.25}$$

The main components of the Universe can be divided into three groups: ultra-relativistic matter and radiation; non-relativistic matter (baryonic and Dark Matter); dark energy, of which the cosmological constant is one of the possible forms.
As we will show in Section 1.1.4, radiation is currently negligible. So, at present time, we can relate the curvature of the Universe to its energy content from equation (1.25) as follows:

- $k = 0 \quad \Leftrightarrow \quad \Omega_{\Lambda,0} + \Omega_0 = 1 \quad$ (flat Universe)
- $k = 1 \quad \Leftrightarrow \quad \Omega_{\Lambda,0} + \Omega_0 > 1 \quad$ (closed Universe)
- $k = -1 \quad \Leftrightarrow \quad \Omega_{\Lambda,0} + \Omega_0 < 1 \quad$ (open Universe)

Furthermore, the measurements of the deceleration parameter q clearly indicate that $\ddot{a} > 0$ (Riess, A. G. et al., 1998). From Equation (1.20a), one can show that this is only possible if

$$\Lambda c^2 > 4\pi G \left(\rho + \frac{3P}{c^2}\right)$$

i.e., the observed accelerated expansion of the Universe is made possible by a positive cosmological constant.

Equation of state

We introduce now an equation of state that defines the relation between pressure and density of the cosmic fluid. This is necessary to find solutions of the Friedmann equations, that do not allow a definition of $\rho(t)$, $p(t)$, $a(t)$.

In Standard Cosmology this equation assumes the general form

$$P = w\rho c^2 \tag{1.26}$$

where w is the *state parameter* and its value is different for each component of the cosmic fluid.
For the three main components in the Universe:

1. Non-relativistic matter, characterised by $w = 0 \;\Rightarrow\; P_M = 0$.
   This is the so-called dust-dominated Universe.
2. Ultra-relativist matter and radiation, characterised by
   $w = \frac{1}{3} \;\Rightarrow\; P_R = \frac{1}{3}\rho_R c^2$.
3. Cosmological constant, characterised by $w = -1 \;\Rightarrow\; P_\Lambda = -\rho_\Lambda c^2$.

Inserting Equation (1.26) in (1.21) one can calculate a simple relation between ρ and a:

$$\rho_w \propto a^{-3(1+w)} \propto (1+z)^{3(1+w)}. \tag{1.27}$$

which yields, in the three cases described:

1. Dust-dominated Universe: $w = 0 \;\Rightarrow\; \rho_M = \rho_{M,0}\left(\frac{a_0}{a}\right)^3$.
2. Radiation-dominated Universe: $w = \frac{1}{3} \;\Rightarrow\; \rho_R = \rho_{R,0}\left(\frac{a_0}{a}\right)^4$.
3. Λ-dominated Universe: $w = -1 \;\Rightarrow\; \rho_\Lambda = \rho_{\Lambda,0}\left(\frac{a_0}{a}\right)^0 = \rho_{\Lambda,0}$.

1.1.4 Evolution of densities with cosmic time

As seen in Section 1.1.3, the energy density of each component has a different evolution with cosmic time.
It is possible to divide the history of the Universe into three main epochs, according to the dominant component: the radiation-dominated era, the matter-dominated era and the dark energy-dominated era.
It is feasible to calculate the transition redshifts from one era to the other, starting from Equation (1.27):

- Matter-radiation equivalence, when $\rho_M = \rho_R$:

$$\begin{aligned} \rho_{M,0}(1+z_{eq})^3 = \rho_{R,0}(1+z_{eq})^4 \Rightarrow \\ (1+z_{eq}) = \frac{\rho_{M,0}}{\rho_{R,0}} \sim 3\cdot 10^4 \Rightarrow z_{eq}^{MR} \sim 3\cdot 10^4 \end{aligned} \tag{1.28}$$

- Matter-Λ equivalence, when $\rho_\Lambda = \rho_M$

$$\begin{aligned} \rho_{\Lambda,0} = \rho_{M,0}(1+z_{eq})^3 \Rightarrow \\ (1+z_{eq}) = \left(\frac{\rho_{\Lambda,0}}{\rho_{M,0}}\right) \sim 1.7 \Rightarrow z_{eq}^{\Lambda M} \sim 0.7 \end{aligned} \tag{1.29}$$

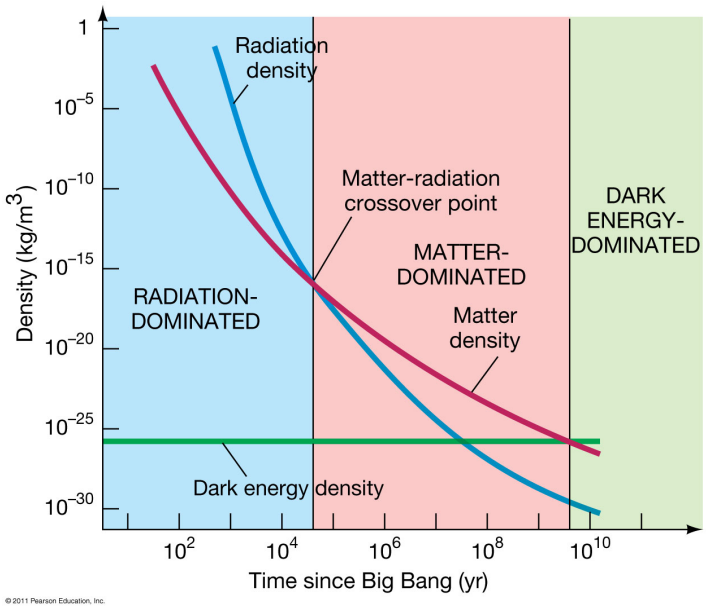


Figure 1.1: Evolution of the densities of cosmic components with cosmic time. The blue, pink and green solid lines represent, respectively, the evolution of radiation, matter and dark energy densities. Image from https://pages.uoregon.edu/jimbrau/. Credits to 2011 Pearson Education, Inc.

The evolution of densities with cosmic time is schematically pictured in Figure 1.1. It is evident that dark energy is now dominant, while in the redshift range $0.7 \lesssim z \lesssim 3 \cdot 10^4$ matter was the main component of the Universe and at $z \gtrsim 3 \cdot 10^4$ radiation was predominant.
As a first approximation, one can use the main component to describe the evolution of the Universe in the three different epochs:

- at $z \gtrsim 3 \cdot 10^4 \Rightarrow w_{eff} \sim 1/3$,
- in $0.7 \lesssim z \lesssim 3 \cdot 10^4 \Rightarrow w_{eff} \sim 0$;
- at $z \lesssim 0.7 \Rightarrow w_{eff} \sim -1$.

Combining Equations (1.26) and (1.20a), it is possible to show (Coles & Lucchin, 2002) that for the expansion of the Universe to be accelerated ($\ddot{a} > 0$), it is necessary that $w < 0$. In particular, this is the case of a Λ-dominated Universe, while the expansion would be decelerated if one of the other two components, matter or radiation, were predominant.
Assembling this information with the observational evidence that $\dot{a} > 0$, one can calculate that at some time in the past, labeled $t = 0$, the scale factor was null: $a(0) = 0$. At this instant, called *Big Bang*, the density and the Hubble parameter diverge.
Moreover, since $a(t)$ is a concave function, the time between the singularity and the epoch t must always be less than the characteristic expansion time of the Universe, the *Hubble time* $\tau_H = 1/H = a/\dot{a}$. This is shown in Figure 1.2.

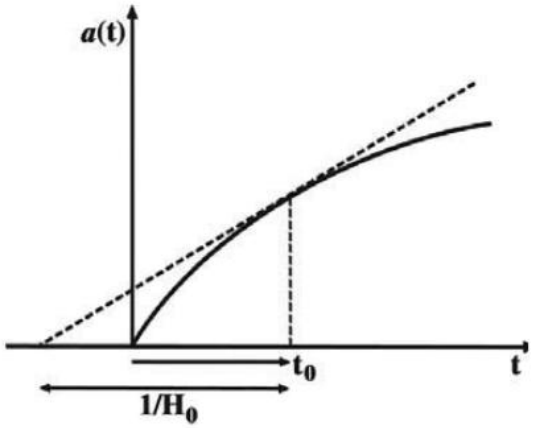


Figure 1.2: The concavity of the scale factor together with the observational evidence that $\dot{a} > 0$, ensures that there must have been a singularity in a finite time in the past, i.e. a point when a = 0. It also secures that the age of the Universe t_0 is less than the Hubble time $1/H_0$. Figure from Coles & Lucchin (2002).

1.1.5 Friedmann model for a flat Universe

In a Universe dominated by one component, equation (1.25) is of the form

$$H_0^2(1-\Omega_{w,0}) = -\frac{kc^2}{a_0^2}. \tag{1.30}$$

In this scenario Equation (1.20a) along with Equation (1.27) give

$$H^2(t) = H_0^2\left(\frac{a_0}{a}\right)^2\left[1-\Omega_{w,0}+\Omega_{w,0}\left(\frac{a_0}{a}\right)^{1+3w}\right]. \tag{1.31}$$

In the case of a flat Universe, i.e. $k = 0$, $\Omega_{w,0} = 1$, it is possible to find explicit solutions for $H(t)$, $a(t)$, $q(t)$, $\rho_w(t)$:

$$t_0 = \frac{1}{H_0}\frac{2}{3(1+w)}, \tag{1.32a}$$

$$a(t) = a_0\left(\frac{t}{t_0}\right)^{\frac{2}{3(1+w)}}, \tag{1.32b}$$

$$H(t) = \frac{2}{3(1+w)}\frac{1}{t}, \tag{1.32c}$$

$$q(t) = \frac{3(1+w)}{2} - 1 = q_0, \tag{1.32d}$$

$$\rho(t) = \frac{1}{6\pi G(1+w)^2 t^2}. \tag{1.32e}$$

As this set of Equations points out, in this model the Universe undergoes an indefinite expansion, dependent on w, namely on its main component.
Moreover, in this scenario, the geometry of the Universe cannot change during its evolution. This is evident by combining Equations (1.27) and (1.31):

$$\Omega_w(z) = \frac{\Omega_{w,0}(1+z)^{1+3w}}{1-\Omega_{w,0}+\Omega_{w,0}(1+z)^{1+3w}} \tag{1.33}$$

which shows that $\Omega_{w,0}$ stays greater, equal or smaller than 1 throughout the evolution of the Universe, even while approaching the Big Bang at $z \to \infty$.

1.1.6 Measurements of the cosmological parameters

We shall now give a brief review of the measured values of the main cosmological parameters.
One of the most reliable measurements of H_0 is the one obtained by the Planck experiment (Planck collaboration et al., 2020), that is based on the analysis of the properties of the Cosmic Microwave Background Radiation. It has been estimated that

$$H_0 = (67.4 \pm 0.5)\ km\, s^{-1} Mpc^{-1}.$$

It is conventional to express $H_0 = 100\, h\, km\, s^{-1} Mpc^{-1}$, where h is a parameter that takes into account uncertainties related to the value of H_0. In many practical applications it is considered $h = 0.7$.
With this notation it is possible to evaluate the present-day value of the critical density:

$$\rho_{c,0} = \frac{3H_0^2}{8\pi G} \approx 2 \cdot 10^{-29} g\, cm^{-3} h^2.$$

Other important results of the Planck collaboration are the measurement of

$$\Omega_M = 0.315 \pm 0.007$$

and the constraint put on the curvature parameter, strongly indicating that the Universe is flat.
These results are in agreement with the outcomes of other experiments, such as COBE and BOOMERANG experiments (Jaffe, A. H. et al., 2001), also focused on the CMB, and high-redshift type Ia Supernovae observations (Riess, A. G. et al., 1998).
From these measurements follows that $\Omega_\Lambda \sim 0.7$ and $q \sim -0.55$.
In summary, the cosmological model with the lowest number of free parameters in agreement with the theory of General Relativity is the Λ-CDM model, according to which the Universe is flat and is composed of Dark Matter and Λ.

1.2 Theory of cosmological perturbations

At present time the Universe can be considered homogeneous and isotropic on large scales. On smaller scales, though, many structures, like galaxies and galaxy clusters,

are observed. In order to explain how they originally formed and then evolved, it is necessary to introduce density fluctuations in the primordial Universe and study their growth.
The origin of the first perturbations is explained by the *Inflationary Theory*, whose predictions are in agreement with today observations. This theory also explains (Coles & Lucchin, 2002) some of the Universe's most important properties, such as flatness and the absence of magnetic monopoles.
According to this Theory, primordial density fluctuations arise from quantum effects in the early Universe. It can be shown that the amplitude of density fluctuations at the time of recombination[3] is proportional to the amplitude of temperature fluctuations observed in the CMB (Bennett, C. L. et al., 1996):

$$\frac{\delta\rho}{\rho} \propto \frac{\delta T}{T} \approx 10^{-5} \qquad at\ z \sim 1100. \tag{1.34}$$

Structure formation theories describe how these fluctuations grow into the structures observed today, characterised by a density field with fluctuations of about $\sim 10^2$ on the scales of collapsed objects.
A linear theory is useful to describe the first stages of the evolution of perturbations, but its assumptions will not hold after $\delta \sim 1$: from that moment on it is necessary to turn to a non-linear theory.

1.2.1 Jeans theory

Jeans theory (Jeans, J. H., 1902) is a linear theory that describes how small fluctuations evolve into collapsed structures.
Firstly, it is convenient to introduce the concept of *radius of the cosmological horizon*, that defines the radius of the sphere of all the events that can be in casual connection with the centre of the sphere.

$$R_H(t) = a(t) \int_0^{a(t)} \frac{da'}{\dot{a}' a'(t')} \tag{1.35}$$

For scales bigger than the cosmological horizon the only possible interaction is the gravitational one, so all the fluctuations with a greater length can grow. On smaller scales, however the gravitational collapse of dark and ordinary matter might be prevented by other physical processes opposing to gravity, namely the cosmic fluid pressure.
The fundamental assumption when describing the gravitational collapse of structures through a linear theory is that the density contrast between the background density of the Universe ρ_b and the density of the perturbation ρ is small, namely

$$\delta = \frac{\delta\rho}{\rho} = \frac{\rho - \rho_b}{\rho_b} \ll 1. \tag{1.36}$$

This is true in the early stages of perturbations' growth.

[3] Recombination is the epoch at which matter became neutral for the first time in cosmic history. Prior to recombination photons were scattered by free particles (electrons and protons) and could not travel freely in the Universe. After recombination the Universe's optical depth decreased rapidly, so the CMB is the first event possibly observable.

Growth of perturbations in a static Universe

A homogeneous fluid satisfies the following conditions: the continuity equation, that implies the conservation of mass; the Euler equation, that guarantees the conservation of the momentum; the Poisson equation and the entropy conservation (since only adiabatic perturbations are considered), respectively:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla(\rho \vec{v}) = 0 \\ \frac{\partial \vec{v}}{\partial t} + \vec{v}\nabla\vec{v} = -\frac{1}{\rho}\nabla P - \nabla\Phi \\ \Delta\Phi = 4\pi G\rho \\ \frac{dS}{dt} = 0 \end{cases} \tag{1.37}$$

where $\vec{v}$ is the velocity of the considered fluid element, Φ is its Newtonian gravitational potential and S represents its entropy.
This system has a solution of the kind $\rho = \rho_b = const$, $P = P_b = const$, $\Phi = \Phi_b = const$, $\vec{v} = 0$. Now, we add small perturbations to these quantities ($\delta\rho$, δP, $\delta\Phi$ and $\delta\vec{v}$) and substitute them in the system (1.37), keeping the first-order terms and discarding all the higher-order ones. We obtain the following set of equations:

$$\begin{cases} \frac{\partial \delta\rho}{\partial t} + \rho_b\nabla(\delta\vec{v}) = 0 \\ \frac{\partial \delta\vec{v}}{\partial t} = -\frac{v_s^2}{\rho_b}\nabla\delta\rho - \nabla\delta\Phi \\ \Delta\delta\Phi = 4\pi G\delta\rho\rho_b \end{cases} \tag{1.38}$$

where v_s^2 is the sound speed of the fluid in adiabatic conditions, when it is possible to apply the relation $\delta P = v_s^2\delta\rho$.
We now consider the perturbations in the form of plane waves:

$$f(r,t) = f_k \exp(ikr + i\omega t) \tag{1.39}$$

where f stands for a generic quantity that describes the system (ρ, P, etc.), f_k is the wave amplitude and k is the *wave number*.
Inserting this Equation in the system (1.38), we find a new system whose determinant is null if

$$\omega^2 = v_s^2k^2 - 4\pi G\rho_b. \tag{1.40}$$

This condition is called *dispersion relation*. We can distinguish between three different cases:

1. $\omega^2 > 0$: the dispersion relation has two real solutions, leading to the propagation of plane waves without significant increase or decrease;

2. $\omega^2 < 0$: the dispersion relation has two imaginary solutions, one of which implies an exponential growth of the amplitude of the perturbation with time;

3. $\omega^2 = 0$: this condition represents the boundary between the two former ones. Recalling that $k = \frac{2\pi}{\lambda}$, we can define the *Jeans length* as the scale $\lambda = \lambda_J$ at which $\omega^2 = 0$:

$$\lambda_J = v_s\sqrt{\frac{\pi}{G\rho_b}}. \tag{1.41}$$

Equivalently, one can define the *Jeans mass* as the mass of the sphere of density ρ_b and radius λ_J:

$$M_J = \frac{4}{3}\pi\rho_b\lambda_J^3. \tag{1.42}$$

Summarising, if the size λ of a perturbation is such that $\lambda > \lambda_J$, it will be subjected to gravitational collapse, otherwise it will propagate as a stable wave.

Growth of perturbations in an expanding Universe

Three length scales will be considered in the following discussion:

1. *The cosmological horizon* R_H: as already explained, the only possible interaction on scales greater than this one is the gravitational one;
2. *Jeans scale* R_J: it is important that $R_J < R_H$;
3. *Dissipation scale* R_D: on smaller scales there are no perturbations because of dissipation.

Moreover, two important points in time need to be taken into account:

1. *Equivalence time* t_{eq}: before this time radiation was predominant, while after matter can be considered the main component of the Universe;
2. *Decoupling time* t_{DEC}: before decoupling, matter could not undergo gravitational collapse because the radiation's pressure opposed it.

It should be noted that Dark Matter and baryons decouple from radiation at different times, and for this reason they collapse separately.
We can start from the case $\lambda > R_H$. The perturbation can be seen as a closed Universe ($\Omega > 1$) in a flat background Universe ($\Omega_b = 1$). Studying the Friedmann equations one finds

$$\delta \propto \frac{1}{\rho_b a^2} \tag{1.43}$$

where the *density contrast* δ describes the evolution of the perturbation's density and is defined as

$$\delta = \frac{\rho - \rho_b}{\rho_b}. \tag{1.44}$$

The density contrast has a different evolution with time before and after the equivalence:

$$\begin{cases} t < t_{eq} \Rightarrow \delta \propto a^2 \propto t \\ t > t_{eq} \Rightarrow \delta \propto a \propto t^{2/3} \end{cases}. \tag{1.45}$$

These equations clearly show that all the perturbations grow outside the cosmological horizon. Also, they are applied to all the components of the cosmic fluid, not just to the dominant one, since the only relevant force is gravity.

Moving on to the case $\lambda < R_H$, the solutions we will find will only hold for the main component of the Universe, because the others may be influenced by other microphysical processes.
Taking the expansion of the Universe into account, one can still assume a solution of the kind $f = f + \delta f$, where $f = \rho, P, \phi$, but two different terms have to be considered for $\vec{v}$: the peculiar motion of the fluid element $\vec{v}_p$ and the velocity of the Hubble flow:

$$\vec{v} = \frac{d\vec{r}}{dt} = \frac{d(a\vec{x})}{dt} = a\frac{d\vec{x}}{dt} + \vec{x}\frac{da}{dt} = H\vec{r} + \vec{v}_p \tag{1.46}$$

where $\vec{r}$ is the proper position of the element, while $\vec{x}$ is the comoving one.
Following an approach similar to the one employed in the case of a static Universe for solving Equation (1.37), we can consider solutions in the form of plane waves $\delta(r,t) = \delta_k(t)\exp(ikx)$ and find the dispersion relation

$$\ddot{\delta}_k + 2\dot{\delta}_k\frac{\dot{a}}{a} + \delta_k\left[k^2v_s^2 - 4\pi G\rho_b\right] = 0 \tag{1.47}$$

that can be solved with the ansatz $\delta_k \propto t^\alpha$.
In a matter-dominated, flat Universe ($t > t_{eq}$) this brings to

$$\lambda_J = \frac{v_s}{5}\sqrt{\frac{24\pi}{G\rho_b}} \tag{1.48}$$

that, if $\lambda > \lambda_J$, gives two solutions

$$\begin{cases} \delta_- \propto t^{-1} \propto a^{-3/2} \propto (1+z)^{3/2} \\ \delta_+ \propto t^{2/3} \propto a \propto (1+z)^{-1} \end{cases} \tag{1.49}$$

It is noticeable that these equations do not depend on the kind of matter, whether it is dark or baryonic, but since they decouple at different times from radiation, it will change the time at which the growth starts.
In a radiation-dominated Universe ($t < t_{eq}$), the dispersion relation becomes

$$\ddot{\delta} + \dot{\delta}_k\frac{\dot{a}}{a} + \delta_k\left[k^2v_s^2 - \frac{32}{3}\pi G\rho_b\right] = 0 \tag{1.50}$$

that, with the same ansatz used on equation (1.47), leads to

$$\lambda_J = v_s\sqrt{\frac{3\pi}{8G\rho_b}}. \tag{1.51}$$

If $\lambda > \lambda_J$ perturbations grow according to

$$\delta_+ \propto t \propto a^2 \propto (1+z)^{-2}. \tag{1.52}$$

However, it can be proven that under these conditions $\lambda_J > R_H$.
Even though for $\lambda < R_H$ these results are only valid for the dominant component, which is radiation in this case, at $t < t_{eq}$ radiation and matter were coupled because of the frequent scatterings. The results found are hence valid for both radiation and matter.

However, Dark Matter decoupled from radiation before baryonic matter. This means that Dark Matter perturbations were able to grow before the equivalence. It can be shown (Coles & Lucchin, 2002), that in $t_{eq} < t < t_{DEC}$

$$\delta_{k,DM} = 1 + \frac{3}{2}\frac{a}{a_{eq}} \quad \Rightarrow \quad \delta_{k,DM} \leq \frac{5}{2} \qquad at \;\; t = t_{eq}. \tag{1.53}$$

This is called the *Meszaros effect* or *stagnation effect*: Dark Matter perturbations don't grow much before the equivalence, and after that they grow following Equation (1.45).
At decoupling, when baryons could undergo gravitational collapse, the Universe was already filled with wells of Dark Matter. This generated the so-called *baryon catch-up*: baryonic perturbations followed the Dark Matter ones:

$$\delta_{k,B} = \delta_{k,DM}\left(1 - \frac{a_{DEC}}{a}\right) \quad at \;\; t > t_{DEC}. \tag{1.54}$$

In a Universe with no Dark Matter, the baryonic perturbations would grow slower, and the structures observed today would be smaller.
A summary of the results listed in this Subsection are given in Table 1.1.

Table 1.1: Summary of the growth of perturbations of dark and baryonic matter on different scales, before and after the equivalence.

	$t < t_{eq}$	$t_{eq} < t < t_{DEC}$	$t > t_{DEC}$
$\lambda > R_H$	$\delta_R,\ \delta_{DM},\ \delta_B \propto a^2$	$\delta_{DM},\ \delta_B,\ \delta_R \propto a$	$\delta_{DM},\ \delta_B,\ \delta_R \propto a$
$R_J < \lambda < R_H$	$\delta_R,\ \delta_B$ oscillate δ_{DM} undergoes stagnation effect	$\delta_{DM} \propto a$ $\delta_R,\ \delta_B$ oscillate	$\delta_{DM} \propto a$ $\delta_B \propto \delta_{DM}\left(1 - \frac{a_{DEC}}{a}\right)$
$\lambda < R_J$	δ_{DM} oscillate	$\delta_{DM},\ \delta_R,\ \delta_B$ oscillate	δ_R oscillate

As already seen, perturbations not massive enough to collapse propagate as density waves, influenced by the gravitational field. This phenomenon causes a leveling of the existing perturbations, which will have at least the scale reached by the Dark Matter perturbations. This scale is called *free streaming scale*:

$$\lambda_{FS}(t) = a(t)\int_0^t \frac{v_s dt'}{a(t')} \tag{1.55}$$

where v_s is the speed of the Dark Matter perturbations. This scale increases with time. It is also possible to define the *free streaming mass*, $M_{FS} \propto \lambda_{FS}^3$. At $t < t_{eq}$, $M_{FS} = M_J$, while at $t > t_{eq}$ $M_{FS} = M_J(t_{eq})$ and stays constant.
At the time of equivalence only perturbations with $M > M_{FS}$ existed. M_J has a trend with time and it can be proven (Coles & Lucchin, 2002) that it reached its maximum value, which is the smallest one a perturbation should have in order to keep growing, at $t = t_{eq}$. However, its value depends on the properties of the Dark Matter, i.e. its velocity, as shown in Equation (1.41).

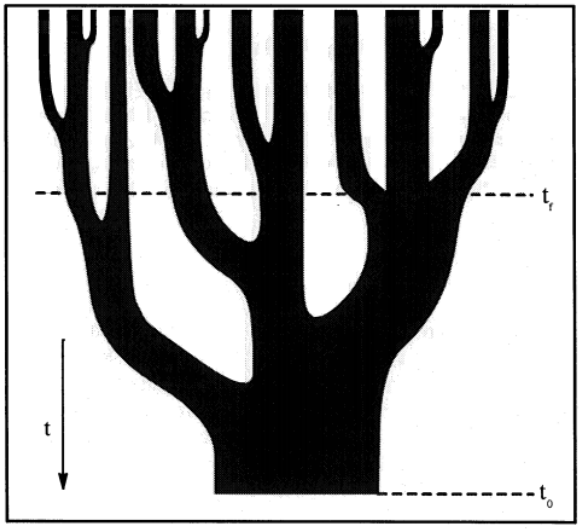


Figure 1.3: A schematic representation of the growth of perturbations for a series of mergers in a bottom-up scenario is given by the merger tree models. The width of the branches represent the masses of the individual parent halos. Time increases from top to bottom. Figure from Lacey, C. & Cole, S. (1993).

Studying the growth of perturbations allows to make some assertions on the nature of Dark Matter. Two models have been been mainly discussed in the scientific community:

- *Cold Dark Matter (CDM) models*: according to this theory the velocity of Dark Matter particles at decoupling is non-relativistic. They predict $M_J(t_{eq}) \sim 10^6 M_\odot$, which is smaller than the average galaxy mass. According to these models the lightest objects formed first, and then merged into more massive structures, in a *bottom-up scenario*;

- *Hot Dark Matter (HDM) models*: in this scenario the Dark Matter particles are lighter and they are still relativistic at decoupling. These models predict $M_J(t_{eq}) \sim 10^{16} M_\odot$, which is larger than the mass of galaxy clusters. In this case, the most massive structures of the Universe are also the oldest ones, and the other formed for fragmentation, in a *top-down scenario.*

Current observations of galaxies and galaxy clusters support CDM models, since many clusters have not reached equilibrium yet. A schematic depiction of the growth of Dark Matter halos by consecutive mergers is given by *merger trees* (Lacey, C. & Cole, S., 1993) in Figure 1.3.

1.2.2 Non-linear theory

Linear theories are only valid until $\delta \sim 1$, but collapsed structures in the present day Universe are characterised by $\delta \sim 10^2 - 10^3$, so it is impossible to describe their formation and evolution without turning to a *non-linear theory.*

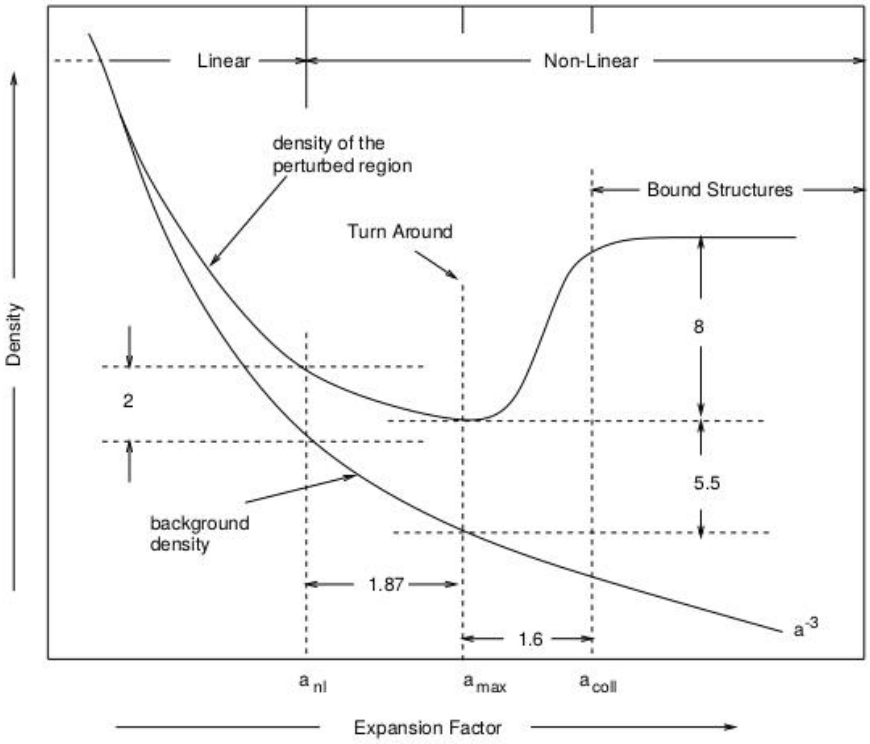


Figure 1.4: Density evolution of a spherical region in the non linear regime. The background Universe is described as a flat and matter-dominated Universe, while the perturbation is described as a closed Universe. Figure from Padmanabhan, T. (2003).

In the non-linear regime it is not feasible to find analytic solutions for the collapse of perturbations, except for the simple case of a spherical collapse. This example does not describe real structures, which are more complex, but it gives some insight into the collapse process.
A synthetic description of the different stages is portrayed in Figure 1.4, from Padmanabhan, T. (2003). In this scenario the background Universe is flat and matter-dominated, while the perturbation is described as a closed Universe of matter.
Firstly, the overdense region expands more slowly than the background Universe, until the turn-around point is reached at $a = a_{max}$, where $\delta(a_{max}) \sim 4.6$, that is definitely in the non-linear regime. At this point the collapse starts and goes on until the structure reaches virialization, at $t = t_{coll}$ when $\delta \sim 400$. At the same moment a linear theory would yield $\delta(t_{coll}) \sim 1.68$.
This example clearly shows how linear theories are not suitable for treating the formation of structures, even in the simplest case.

N-body simulations

N-body simulations are numerical simulations widely used to study the evolution of cosmic structures in the non-linear regime. They require the definition of the number of particles used as field tracers, the characterisation of the initial conditions (i.e.

the initial position of the particles, that should try to reproduce the Universe's properties at high redshift) and the implementation of the physical laws that the particles will follow.
In the case of gravitational collapse of Dark Matter halos, the most important interaction is the gravitational one. However, hydrodinamical simulations are the most accurate simulations when studying phenomena such as galaxy formation or baryonic matter flows, but they are very expensive from a computational point of view. Alternatively one can simulate the distribution of galaxies by placing them in Dark Matter halos obtained by a N-body simulation, following a Halo Occupation Distribution model (Zheng, Z. et al., 2005).
This approach has been followed in several simulations, including one of the most important ones of the recent years, the *Millennium simulation* (Springel, V. et al., 2005), carried out in 2005. This simulation used $N \sim 10^{10}$ particles to trace the distribution of the matter distribution in a large cube of Universe and its evolution with cosmic time.
In Figure 1.5 two snapshots of the simulation are compared, showing the distribution of Dark Matter on large scales at $z = 5.7$ (left) and at $z = 0$ (right).

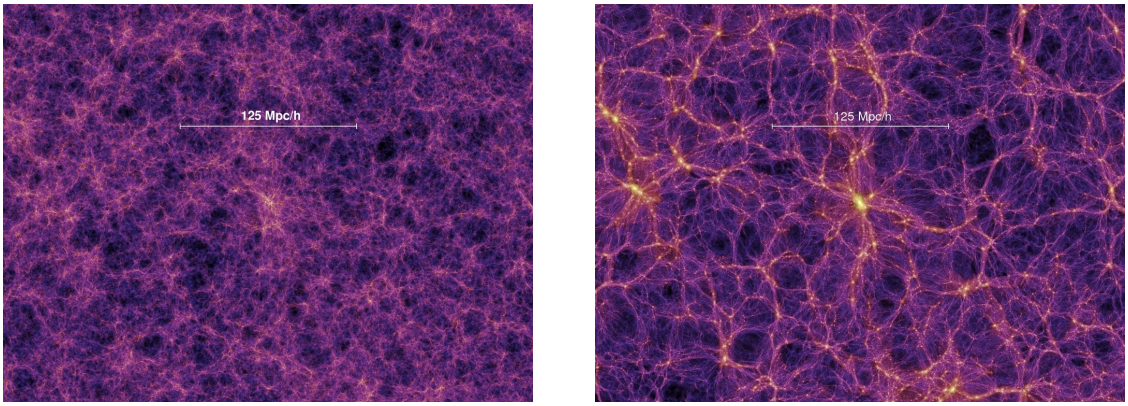


Figure 1.5: Distribution of Dark Matter on large scales at $z = 5.7$ (left) and at $z = 0$ (right), as calculated by the Millennium simulation. Figure from https://wwwmpa.mpa-garching.mpg.de/galform/virgo/millennium/.

As shown in this Figure, the Dark Matter distribution on large scales forms the so-called *cosmic web*, made of filamentary structures connecting denser clumps of matter, where galaxy clusters form.

1.3 Galaxy clusters

Galaxy clusters are the youngest and the largest gravitationally-bound structures in the Universe. According to the bottom-up scenario described in the previous section, they are thought to be formed through a sequence of mergers and accretion of smaller systems.
The gravitational potential of the cluster is defined by the distribution of Dark Matter, which makes up for $\sim 80 - 85\%$ of their mass, while the major part of the

ordinary matter is in the form of hot plasma, the Intracluster Medium (ICM), which accounts for $\sim 10 - 15\%$ of the total mass. Baryonic mass in the form of stars in galaxies only contributes to the total cluster mass for a small percentage. Typically $M_{tot} \sim 10^{14} - 10^{15} M_{\odot}$, with the lowest mass end being called *galaxy groups*.
There are different tracers for the total mass distribution of the cluster, and a combination of all of them gives the best possible description of it. In the central regions studies regarding the stellar kinematics of the Brightest Cluster Galaxy (BCG) and strong gravitational lensing events are most effective, while hydrostatic equilibrium of the hot X-ray gas and weak gravitational lensing phenomena allow to recover the mass distribution in the outer regions, even beyond the *virial radius*.
The main features of galaxy clusters' components will be discussed in the following sections.

1.3.1 Dark Matter

Dark Matter can only be detected by its gravitational effect on other matter and light passing near it. In fact, Gravitational Lensing has proven particularly effective for the definition of the large scale distribution of Dark Matter in clusters, as well as in individual galaxies within them.
The radial distribution of Dark Matter is commonly described by the *Navarro-Frenk-White* (NFW) profile (Navarro, J. F., Frenk, C. S., White S. D. M., 1996), firstly introduced as a universal profile adequate to model density profiles of Dark Matter halos over a wide range of masses. Its analytical expression is given by

$$\rho(r) = \frac{\rho_s}{\left(\frac{r}{r_s}\right)\left(1 + \frac{r}{r_s}\right)^2} \tag{1.56}$$

where r_s is the scale radius and $\rho_s = \rho_{DM}(r = r_s)$. At $r < r_s$ $\rho(r) \propto r^{-1}$, while at $r > r_s$ $\rho(r) \propto r^{-3}$. So the NFW profile is stepper than an isothermal profile ($\rho(r) \propto r^{-2}$) in the outer regions and it is shallower in the inner ones, as shown in Figure 1.6.
It should be noted that equation (1.56) represents an average profile of Dark Matter halos in equilibrium, but there is considerable scatter in the possible shapes of the profiles of particular clusters (see Meneghetti et al., 2014; Newman et al., 2011).
From the definition of the NFW profile follows that of *concentration* $c_\Delta = r_\Delta / r_s$, where r_Δ represents the size of the halo, defined as the radius enclosing a mean overdensity Δ above the critical density of the Universe $\rho_{crit}(z)$. The most adequate way to describe a cluster would be to use its virial radius r_{vir}, but the value Δ_{vir} is redshift and cosmology dependent. To avoid this dependence the value $\Delta = 200$, at first proposed by Navarro, J. F., Frenk, C. S., White S. D. M. (1996), has been commonly used in literature.
The mass profile that corresponds to the NFW density profile is

$$M(r) = 16\pi\rho_s r_s^3 \left[\ln\left(1 + \frac{r}{r_s}\right) - \frac{r/r_s}{1 + r/r_s}\right] \tag{1.57}$$

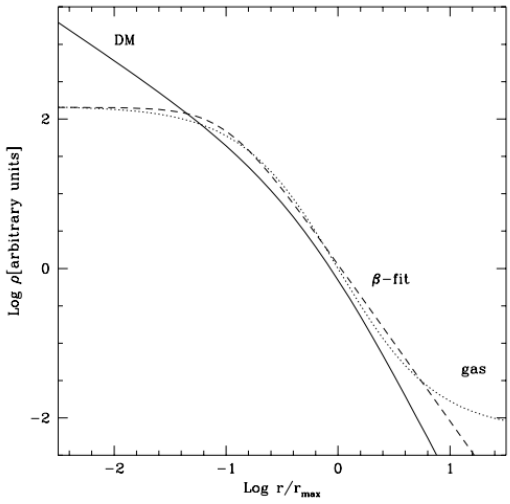


Figure 1.6: Comparison between the Dark Matter density radial profile given in Equation (1.56) (solid line) and the one of an isothermal sphere of gas (dotted line). The dashed line represents the β-model often employed for the description of the intracluster medium radial profile (Cavaliere, A. & Fusco-Femiano, R., 1976).

1.3.2 Galaxies

Galaxy clusters contain from tens to thousands of galaxies, that differ from isolated galaxies in morphology. In fact, while clusters mainly host Early Type Galaxies (ETGs) (Dressler, A., 1980), i.e. elliptical galaxies, spiral and irregular galaxies are more common in the field, as shown in Figure 1.7.
Also, cluster members usually have low Star Formation Rate (SFR) (Hogg, D. W. et al., 2004): they tend to be red if compared to field galaxies, as it can be seen in Figure 1.8.
Every galaxy cluster hosts a Brightest Cluster Galaxy (BCG) at its center. BCGs are a distinct class of galaxies, that include the most luminous and massive galaxies in the Universe. In fact, they can be 10 times more luminous than field galaxies, and reach $M \sim 10^{13} M_{\odot}$, comparable to the mass of small groups (Katayama, H. et al., 2003). Their formation history is probably different from that of other galaxies. Three models try to explain their origin: star formation from cooling flows expected in the centers of clusters (Fabian, A. C., 1994), cannibalism or accretion of existing galaxies (Ostriker, J. P. & Hausman, M. A., 1977), galaxy merging in the early history of the formation of clusters (Merritt, D., 1985).
From the kinematics of galaxies one can infer the cluster total mass. Assuming virial equilibrium, it is possible to relate the kinetic energy of the system K to its gravitational potential energy U: $2K + U = 0$. K and U write, respectively

$$K \approx \frac{1}{2} M \sigma_v^2 \tag{1.58}$$

and

$$U \approx \frac{GM^2}{R} \tag{1.59}$$

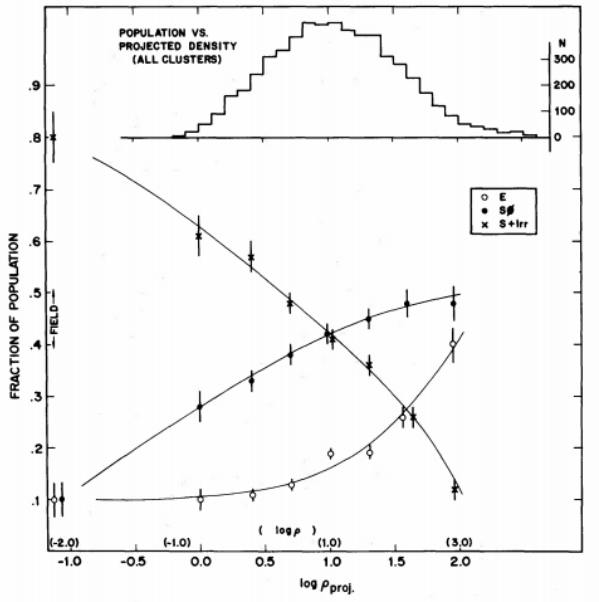


Figure 1.7: The fraction of elliptical (E), lenticular (S0), spiral and irregular (S + I) galaxies as a function of the log of the projected density (in galaxies/Mpc). The upper histogram shows the number distribution of the galaxies over the bins of projected density. The sample includes 55 clusters in the local Universe. Figure from Dressler, A. (1980).

for a cluster. Here M is the total mass of the cluster, σ_v is the velocity dispersion of the cluster members (typically, $\sigma_v \sim 10^3\, km/s$) and R is the cluster radius. From these equations follow (Rosati, P. et al., 2002):

$$M \approx \frac{R\sigma_v}{G} \approx \left(\frac{R}{1h^{-1}Mpc}\right)\left(\frac{\sigma_v}{10^3 km/s}\right)\ 10^{15}h^{-1}M_\odot. \tag{1.60}$$

1.3.3 Intracluster medium

As previously said, most of the ordinary matter in clusters is in the form of a hot plasma. Assuming that gas and galaxies share the same dynamics, it can be proven (Rosati, P. et al., 2002) that:

$$k_B T_{ICM} \approx \mu m_P \sigma_v^2 \approx 6\left(\frac{\sigma_v}{10^3 kms^{-1}}\right)^2 keV \tag{1.61}$$

where k_B is the Boltzmann constant, μ is the mean molecular weight of the gas and m_P is the proton mass. Typically $T_{ICM} \sim 10^6\, K$. This material is also very tenuous, and its density increases towards the center of the cluster. Because of these properties, the ICM emits X-rays via the Bremsstrahlung emission process, which emits with a brightness proportional to the density of the ICM squared.
In Figure 1.9 is an example of the Coma cluster as observed by XMM-Newton in the X-rays: here it is possible to appreciate how the center of the cluster emits more light than the outer regions in this band of the spectrum.

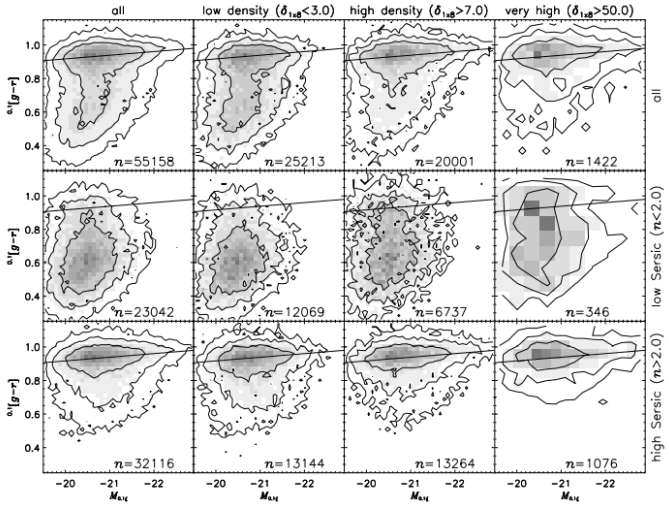


Figure 1.8: Color of the galaxies [g-r] as a function of their absolute magnitude in the i band M_i. The columns show subsamples cut in overdensity (the environment density increases from left to right), while the rows show subsamples cut in Sersic index (n increases from top to bottom, distinguishing early-type from late-type galaxies). Figure from Hogg, D. W. et al. (2004).

Figure 1.9: Coma galaxy cluster in X-ray (diffuse emission) and optical light (galaxies), as seen by XMM-Newton's European Photon Imaging Camera (EPIC) and the Sloan Digital Sky Survey (SDSS). Credits to ESA/XMM-Newton/SDSS/J. (Sanders, J. S. et al., 2020)

Assuming spherical symmetry, the condition of hydrostatic equilibrium writes

$$\frac{dp}{dr} = -\frac{GM(<r)\rho(r)}{r^2} \tag{1.62}$$

where p is the gas pressure. When considering the equation of state of a perfect gas, Equation (1.62) yields

$$M(<r) = -\frac{k_B T r}{G\mu m_p}\left(\frac{d\ln\rho}{d\ln r} + \frac{d\ln T}{d\ln r}\right). \tag{1.63}$$

From X-ray observations it is feasible to find both the temperature and density profiles of the cluster, so this is a common way to measure the gravitational mass of these structures. Nevertheless, the assumption of hydrostatic equilibrium does not hold for some clusters, and other contributions to the total pressure should be taken into account (Eckert, D. et al., 2019).
A different way of studying the hot gas in clusters of galaxies is through the observation of decrements in the intensity of the Cosmic Microwave Background associated with the *Sunyaev-Zeldovich effect* (Rephaeli, Y., 1995).
Some of the photons of the CMB suffer Compton scattering by the hot electrons while passing through the gas cloud. Because of these scatterings the spectrum of the CMB is shifted to slightly higher energies (Longair, S. M., 2011). There is expected to be a decrease in the intensity of the CMB in the Rayleigh-Jeans of the spectrum (at $h\nu \ll kT_{CMB}$), while in the Wien region ($h\nu \gg kT_{CMB}$) there should be a slight excess in T_{CMB}, as shown in Figure 1.10.
The magnitude of the distortion is determined by the *Compton scattering optical depth* through the region of hot gas

$$y = \int \left(\frac{k_B T_e}{m_e c^2}\sigma_T N_e dl\right) \tag{1.64}$$

where N_e, T_e and m_e represent the numerical density, temperature and mass of the electrons respectively and σ_T is the Thomson scattering cross section.
The reduction in the Rayleigh-Jeans region is given by

$$\frac{\Delta I_\nu}{I_\nu} = -2y. \tag{1.65}$$

An important property of the Sunyaev-Zeldovich effect in Cosmology is that, if the host galaxies have the same properties at all redfhits, the observed decrease is independent of redshift. Combining Sunyaev-Zeldovich and bremsstrahlung emission observations, it is possible (Longair, S. M., 2011) to estimate Hubble's constant without the necessity of using a hierarchy of distance indicators.

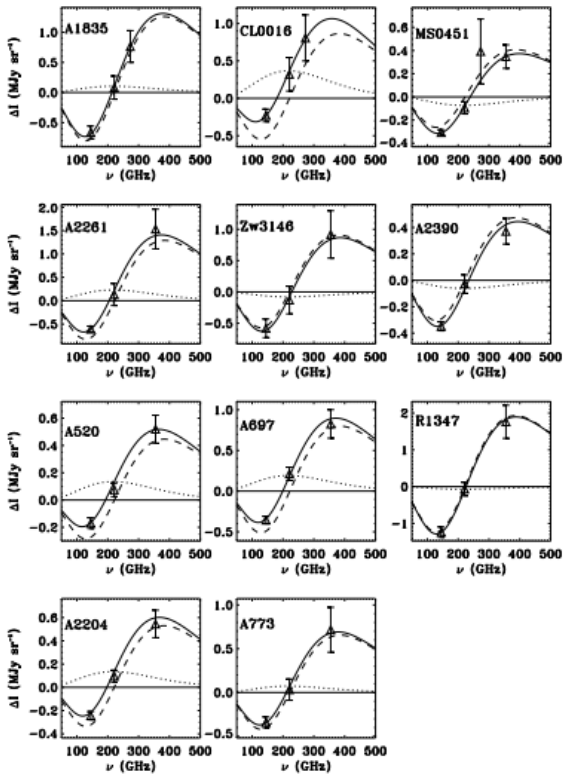


Figure 1.10: The measured SZ spectrum for eleven clusters. In each plot the solid line is the best-fit SZ model, the dashed line is the thermal component of the SZ effect and the dotted line is the kinematic component of the SZ effect. Figure from Benson, B. A. et al. (2004).

Chapter 2

Gravitational Lensing Theory

The presence of a massive object may deflect the path followed by photons passing nearby: this phenomenon is called *Gravitational Lensing*.
The massive body that acts as a deflector is called *lens* and it may be point-like, as a star, or extended, as a galaxy or a galaxy cluster. On the other hand, the luminous object, whose light is deflected, is called *source* and it could be a background galaxy or a distant quasar.
Gravitational lensing studies have several purposes in Astrophysics. One of the most common applications is the measurement of the mass of the lens and its radial distribution (e.g. Lagattuta et al. 2019; Caminha et al. 2017). Gravitational lensing observations have also proved valuable to constrain the value of H_0 (e.g. Koopmans et al. 2003; Wong et al. 2020) and to investigate cosmological models in general (e.g. Stark et al. 2007; Vanzella et al. 2020a; Grillo 2012). Finally, the flux magnification that derives from the deflection of the light of distant sources, allows to observe faint objects at high redshifts (Vanzella et al. 2020b).
In this Chapter we will give an overview of the basics of gravitational lensing theory, based on Meneghetti, M. (2018).

2.1 An Introduction to Gravitational Lensing

2.1.1 Deflection angle

The idea that photons may be deflected by a mass located along their path was originally expressed by Newton in the context of the Corpuscolar Theory of Light (Newton, I., 1704), and it was later confirmed and explained by Einstein's theory of General Relativity.
As seen in Section 1.1, according to this theory, the geometrical properties of space-time depend on its energy-matter content: light paths are bent by the gravitational effect of masses situated between the source of light and the observer.
In order to calculate the *deflection angle* of the light path, we make the assumption that the lens is weak, valid in virtually all astrophysical cases. This approximation corresponds to the request that the Newtonian gravitational potential of the lens Φ is such that $\Phi/c^2 \ll 1$. Equivalently, one could ask for the lens to be small compared to the dimension of the optical system, that includes observer, lens and source.

Under this approximation, the Minkowski metric, that describes unperturbed space-time, is modified by a small perturbation, and the line element writes

$$ds^2 = \left(1 + \frac{2\Phi}{c^2}\right) c^2 dt^2 - \left(1 - \frac{2\Phi}{c^2}\right) (d\vec{x})^2 \tag{2.1}$$

Light travels on null geodesics, for which $ds^2 = 0$, thus the light speed in the gravitational field is

$$c' = \frac{d\vec{x}}{dt} = c\sqrt{\frac{1 + 2\Phi/c^2}{1 - 2\Phi/c^2}} \approx c\left(1 + \frac{2\Phi}{c^2}\right) \tag{2.2}$$

Since $\Phi \leq 0$, we have $c' \leq c$.
Hence, we can describe the space-time as a medium with *effective refraction index*

$$n = \frac{c}{c'} = \frac{1}{1 + 2\Phi/c^2} \approx 1 - \frac{2\Phi}{c^2} \geq 1. \tag{2.3}$$

Employing Fermat's principle it can be shown (Schneider, P., Ehlers, J., Falco, E. E., 1992) that the deflection angle

$$\widehat{\vec{\alpha}}(b) = \frac{2}{c^2} \int_{-\infty}^{+\infty} \nabla_{\perp} \Phi dz \tag{2.4}$$

where it has been assumed that the light ray follows the $\vec{e}_z$ direction at the beginning of its path and passes the lens at $z = 0$ with impact parameter b.
This result is valid if the spatial scales considered are smaller than the distances between source, lens and observer and if the time scale is short enough to make the universe's expansion negligible.
In case of a point mass equation (2.4) reads

$$|\widehat{\vec{\alpha}}|(b) = \frac{4GM}{c^2 b} \tag{2.5}$$

where M is the mass of the lens.
Since the dependence of the deflection angle on the mass M is linear, the deflection angle of a group of lenses M_i, $1 \leq i \leq N$, can be superposed:

$$\widehat{\vec{\alpha}}(\vec{\xi}) = \sum_i \widehat{\vec{\alpha}}_i(\vec{\xi} - \vec{\xi}_i) = \frac{4G}{c^2} \sum_i M_i \frac{\vec{\xi} - \vec{\xi}_i}{|\vec{\xi} - \vec{\xi}_i|^2}. \tag{2.6}$$

Here $\vec{\xi}_i$ are the positions of the lenses, while $\vec{\xi}$ marks the position where the deflection angle is calculated.
In astrophysics it is always justified to use the *thin screen approximation*, that describes the lens as a surface mass density

$$\Sigma(\vec{\xi}) = \frac{4G}{c^2} \int_{-\infty}^{+\infty} \rho(\vec{\xi}, z) dz \tag{2.7}$$

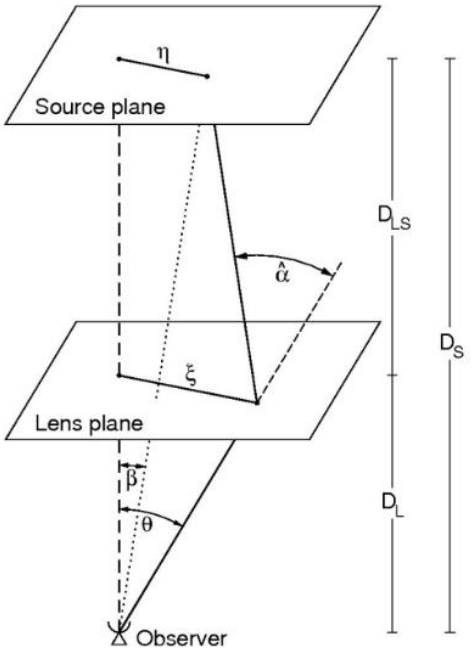


Figure 2.1: Typical configuration of a gravitational lensing system: source and lens are assumed to lie on two different planes. Image from Bartelmann & Schneider (2001).

where ρ is the three-dimensional density of the lens, $\vec{\xi}$ defines a position on the lens plane and $\vec{e}_z$ is the direction perpendicular to it.
Since the typical distances of observer, lens and source are much larger than the physical size of the lens, it is legitimate to use this approximation. The geometrical configuration of a gravitational lensing system is shown in figure 2.1. The sources are also assumed to lie on a plane.
In this approximation the deflection angle can be calculated as

$$\widehat{\vec{\alpha}}(\vec{\xi}) = \frac{4G}{c^2} \int \frac{(\vec{\xi} - \vec{\xi}')\Sigma(\vec{\xi}')}{|\vec{\xi} - \vec{\xi}'|^2} d^2\xi' \tag{2.8}$$

2.1.2 Lens equation

Gravitational lensing effects depend on the relative positions and distances between the observer, the lens and the source. We'll consider a lens at angular distance D_L (or, equivalently, redshift z_L), that deflects the light emitted by a source at angular distance D_S (or redshift z_S).
If the true position of the source is $\vec{\beta}$, it will appear as if it was located in $\vec{\theta}$ because of the deflection angle $\widehat{\vec{\alpha}}$, as shown in Figure 2.1.
If $\vec{\theta}$, $\vec{\beta}$, $\widehat{\vec{\alpha}}$ are small, the *lens equation* relates the true and apparent positions of the source:

$$\vec{\theta} D_S = \vec{\beta} D_S + \widehat{\vec{\alpha}} D_{LS}. \tag{2.9}$$

Here D_{LS} represents the distance between the lens and the source.
Moreover, introducing the *reduced deflection angle*

$$\vec{\alpha}(\vec{\theta}) \equiv \frac{D_{LS}}{D_S}\widehat{\vec{\alpha}}(\vec{\theta}) \tag{2.10}$$

equation (2.9) reads

$$\vec{\beta} = \vec{\theta} - \vec{\alpha}(\vec{\theta}). \tag{2.11}$$

Equation (2.9) is usually written in a dimensionless form. Considering the length scale ξ_0 on the lens plane and the corresponding length scale $\eta_0 = \xi_0 D_S/D_L$ on the source plane, we can define the two vectors

$$\vec{x} \equiv \frac{\vec{\xi}}{\xi_0}, \qquad \vec{y} \equiv \frac{\vec{\eta}}{\eta_0}$$

and the lens equation writes

$$\vec{y} = \vec{x} - \vec{\alpha}(\vec{x}) \tag{2.12}$$

where

$$\vec{\alpha}(\vec{x}) = \frac{D_L D_{LS}}{\xi_0 D_S}\widehat{\vec{\alpha}}(\xi_0\vec{x}). \tag{2.13}$$

2.1.3 Lensing potential

By projecting the three-dimensional Newtonian potential Φ on the lens plane one can define the *effective lensing potential*

$$\hat{\Psi} = \frac{D_{LS}}{D_S D_L}\frac{2}{c^2}\int \Phi(D_L\vec{\theta}, z)dz \tag{2.14}$$

where $D_{LS}/D_S D_L$ is a scaling factor. This quantity also has a dimensionless form

$$\Psi = \frac{D_L^2}{\xi_0^2}\hat{\Psi} \tag{2.15}$$

The lensing potential satisfies two fundamental properties:

1. The gradient of $\hat{\Psi}$ returns the deflection angle:

$$\nabla_x \Psi(\vec{x}) = \vec{\alpha}(\vec{x}) \tag{2.16}$$

2. The Laplacian of $\hat{\Psi}$ corresponds twice the *convergence*:

$$\Delta_x \Psi(\vec{x}) = 2\kappa(\vec{x}) \tag{2.17}$$

where the convergence is defined as a dimensionless surface mass density:

$$\kappa(\vec{x}) = \frac{\Sigma(\vec{x})}{\Sigma_{cr}} \tag{2.18}$$

where

$$\Sigma_{cr} = \frac{c^2}{4\pi G}\frac{D_S}{D_L D_{LS}} \tag{2.19}$$

is the *critical surface density*, a quantity that depends on the distances of lens and source and characterises the lensing system.

2.1.4 Magnification and distortion

One of the most peculiar effects of gravitational lensing in the distortion of the shapes of the background sources. For example, it is common for galaxies to appear in the form of elongated arcs in galaxy clusters.
The shape of the images can be determined by solving the lens equation for all the points within the source. In particular, if the source is smaller than the angular size on which the physical properties of the lens change, these positions can be locally linearized and the distortion of images is described by the *Jacobian matrix*

$$A \equiv \frac{\partial \vec{y}}{\partial \vec{x}} = \delta_{ij} - \frac{\partial \alpha_i(\vec{x})}{\partial x_j} = \delta_{ij} - \frac{\partial^2 \Psi(\vec{x})}{\partial x_i \partial x_j} = \delta_{ij} - \Psi_{ij} \tag{2.20}$$

where x_i is the i-component of $\vec{x}$ and δ_{ij} is the Kronecker delta.
This matrix can be split into two parts:

$$A \equiv \left(A - \frac{1}{2} \operatorname{Tr} A \cdot I \right) + \left(\frac{1}{2} \operatorname{Tr} A \cdot I \right) \tag{2.21}$$

The first part is an antisymmetric, trace-free matrix called the *shear matrix*:

$$\left(A - \frac{1}{2} \operatorname{Tr} A \cdot I \right) = \begin{pmatrix} -\frac{1}{2}(\Psi_{11} - \Psi_{22}) & -\Psi_{12} \\ -\Psi_{12} & \frac{1}{2}(\Psi_{11} - \Psi_{22}) \end{pmatrix}. \tag{2.22}$$

The elements of this matrix lead to the definition of the *shear*, the pseudo-vector $\vec{\gamma} = (\gamma_1, \gamma_2)$ on the lens plane

$$\begin{aligned} \gamma_1(\vec{x}) &= \frac{1}{2}(\Psi_{11} - \Psi_{22}) \\ \gamma_2(\vec{x}) &= \Psi_{12} = \Psi_{21} \end{aligned} \tag{2.23}$$

The eigenvalues of the shear matrix are $\pm\sqrt{\gamma_1^2 + \gamma_2^2} = \pm\gamma$. Thus, it is possible to rotate the coordinate system by an angle ϕ such that

$$\begin{pmatrix} \gamma_1 & \gamma_2 \\ \gamma_2 & -\gamma_1 \end{pmatrix} = \gamma \begin{pmatrix} \cos 2\phi & \sin 2\phi \\ \sin 2\phi & \cos 2\phi \end{pmatrix} \tag{2.24}$$

On the other hand, the isotropic part of the Jacobian matrix is related to the convergence (Meneghetti, M., 2018), so A becomes

$$A = (1 - \kappa) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \gamma \begin{pmatrix} \cos 2\phi & \sin 2\phi \\ \sin 2\phi & -\cos 2\phi \end{pmatrix}. \tag{2.25}$$

This equation clarifies how convergence and shear contribute to the distortion of images. The convergence induces an isotropic distortion, rescaling the images of the same factor in all directions, while the shear stretches the intrinsic shape of the source along the privileged direction identified by the eigenvectors of A.
For instance, the distortion experienced by a circular source is pictured in Figure 2.2, where the contributions of shear and convergence are separated.

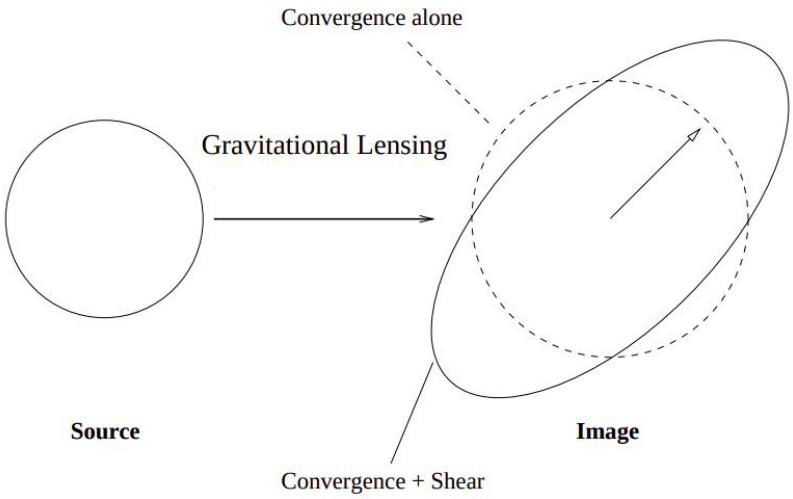


Figure 2.2: The distortion caused by convergence and shear on a circular source. Figure from Umetsu, K. (2010).

Let r be the radius of the circular source: the semi-major and semi-minor axes of the distorted image will be

$$a = \frac{r}{\lambda_t}, \qquad b = \frac{r}{\lambda_r} \tag{2.26}$$

where

$$\begin{aligned} \lambda_t &= 1 - \kappa - \gamma \\ \lambda_r &= 1 - \kappa + \gamma \end{aligned} \tag{2.27}$$

are the tangential and radial eigenvalues of A, respectively.
Another important consequence of gravitational lensing is the *magnification* of the source's images.
The process of deflection of photons by the lens does not involve emission nor absorption, so their number is conserved, and their frequency of emission ν is not altered either. Overall, this means that the surface brightness I_ν of the source is preserved. However, the effect of the lens changes the solid angle under which the source is observed, hence the source may appear magnified or demagnified.
Given the definition of the Jacobian matrix in Equation (2.20), the magnification can be calculated as the inverse of the determinant of A:

$$\mu \equiv \frac{d\theta^2}{d\beta^2} = (\det A)^{-1} = \frac{1}{(1-\kappa)^2 - \gamma^2} \tag{2.28}$$

where $d\theta^2$ is the solid angle intrinsically subtended by the source and $d\beta^2$ is the observed solid angle.
The inverse of the eigenvalues of A measure the amplification in the tangential and

radial directions, respectively:

$$\begin{aligned} \mu_t &= \frac{1}{\lambda_t} = \frac{1}{1-\kappa-\gamma} \\ \mu_r &= \frac{1}{\lambda_r} = \frac{1}{1-\kappa+\gamma} \end{aligned} . \tag{2.29}$$

The curves on the lens planes where $\lambda_t = 0$ and $\lambda_r = 0$ ideally correspond to infinite magnification and are respectively called the *tangential* and the *radial critical lines*. These curves are mapped by the lens equation (2.11) into the *caustics* on the source plane.
When faint sources transit across the caustic of a powerful lens, such as a galaxy cluster, their flux is strongly magnified and it is possible to observe them, even if they would otherwise be invisible.

2.1.5 Time delay surface

The deflection of light rays causes a time delay between the emission of radiation by the source and the signal reception by the observer. This delay can be decomposed into two contributions:

1. *Geometrical time delay*
   The different path length followed by the deflected light rays and the unperturbed ones partially causes the time delay:

$$t_{geom} = \frac{1}{2c}\frac{D_L D_S}{D_{LS}}(\vec{x}-\vec{y})^2 \tag{2.30}$$

2. *Gravitational time delay*
   Photons traveling through the gravitational field of the lens are slowed down, and this leads to an increase in the time delay:

$$t_{grav} = -\frac{D_L D_S}{D_{LS}}\frac{1}{c^2}\Psi(\vec{x}) \tag{2.31}$$

Summing the two contributions, the total time delay can be calculated as follows:

$$t(\vec{x}) = \frac{(1+z_L)}{c}\frac{D_S \xi_0^2}{D_L D_{LS}}\left[\frac{1}{2}(\vec{x}-\vec{y})^2 - \Psi(\vec{x})\right]. \tag{2.32}$$

Since the gradient of the effective lensing potential is related to the deflection angle, the lens equation can be reformulated as

$$(\vec{x}-\vec{y}) - \nabla\Psi(\vec{x}) = \nabla\left[\frac{1}{2}(\vec{x}-\vec{y})^2 - \Psi(\vec{x})\right] = 0. \tag{2.33}$$

Equation 2.33 implies that solving the lens equation and searching the stationary points of the delay surface defined in Equation 2.32 are equivalent.

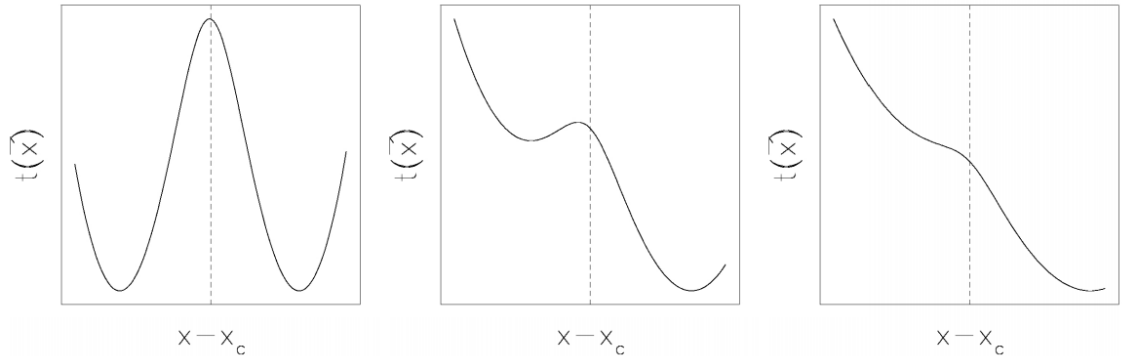


Figure 2.3: Time delay function of an axially symmetric potential $\Psi(x) \propto \sqrt{x^2 + x_c^2}$ for different source positions. This is the projection of the time delay surface along the symmetry axis. Figure from Meneghetti, M. (2018).

The Hessian matrix of this surface corresponds to the Jacobian matrix:

$$T = \frac{\partial^2 t(\vec{x})}{\partial x_i \partial x_j} \propto (\delta_{ij} - \Psi_{ij}) = A \tag{2.34}$$

The Hessian matrix describes the curvature of the time delay surface, and it is proportional to the inverse of the magnification. In particular, magnification is higher along the direction with a smaller curvature. It is possible to distinguish between three types of images:

- *Type I images*, with $\det A > 0$, $\mathrm{Tr}\, A > 0$: images arise at the minima of $t(\vec{x})$ and they have positive magnification;
- *Type II images*, with $\det A < 0$: images arise at the saddle points of $t(\vec{x})$ and they have negative magnification;
- *Type III images*, with $\det A > 0$, $\mathrm{Tr}\, A < 0$: images arise at the maxima of $t(\vec{x})$ and they have positive magnification.

It should be noted that images having negative magnification are not demagnified (this condition corresponds to $|\mu| < 1$), but the parity of the image is flipped compared to the one of the source.
For instance, let's consider an axially symmetric potential and the possible source configurations in Figure 2.3. If the source is perfectly aligned with the lens, $t(\vec{x})$ has a maximum at the lens center, while its minima are located on a ring centered on the maximum. In this situation the source is mapped to a type I ring-shaped image, the so-called *Einstein Ring*, an example of which is in Figure 2.4, and to a central type III image.
When the source is away from the optical axis, the delay surface distorts: the ring breaks and a minimum and a saddle point form. Therefore, three images arise.
Moving the source further away, the maximum and the saddle point get closer and the local curvature of the surface in the radial direction becomes smaller: the images arising at these two points are elongated towards each other and a radial image forms.

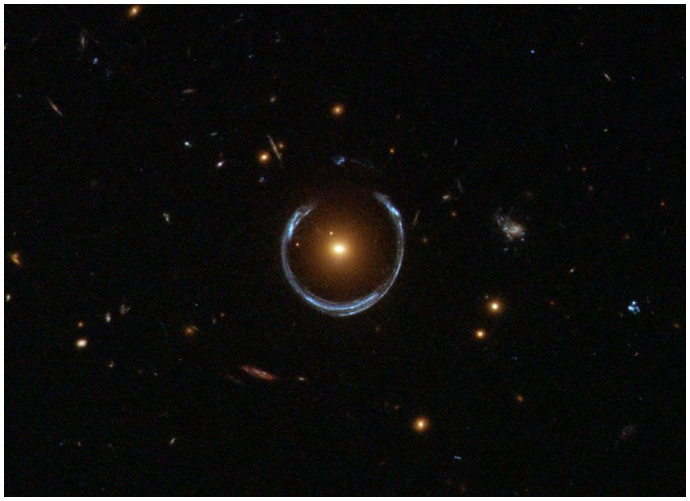


Figure 2.4: Horseshoe Einstein Ring observed with Hubble Space Telescope's Wide Field Camera 3. Image Credit: ESA/Hubble & NASA.

Since he curvature changes from negative to positive between the maximum and the saddle point, critical lines separate multiple-image pairs: if a lens does not have critical lines, it will not be able to form multiple images of the same source.

2.2 Lens Models

Analytic lens models are commonly used for several applications, such as the description of specific image configurations and the prediction of the rate of lensing events.
Different models are required to describe lensing phenomena on different scales. For instance, point lenses approximate the properties of compact objects like stars and planets, while elliptical models are better suited for the characterization of extended lenses, like clusters of galaxies.
In the following sections we will illustrate the main characteristics of some of the most common models used to describe extended lenses.

2.2.1 Axially symmetric models

The surface density of *axially symmetric models* is independent on the position angle with respect to the lens center. Therefore, the lensing equations can be reduced to a one-dimensional form. The deflection angle in this case writes

$$\alpha(x) = \frac{2\int_0^x x'\kappa(x')dx'}{x} = \frac{m(x)}{x} \tag{2.35}$$

where $m(x)$ is the dimensionless mass enclosed within the radius x. The substitution in the lens equation 2.12 yields

$$y = x - \frac{m(x)}{x}, \tag{2.36}$$

while the convergence and shear can be calculated as

$$\begin{aligned} \kappa(x) &= \frac{1}{2x}\frac{dm}{dx}, \\ \gamma(x) &= \bar{\kappa}(x) - \kappa(x) \end{aligned} \tag{2.37}$$

where $\bar{k}(x)$ is the mean value of the convergence within the radius x.
The tangential and radial critical lines are, respectively:

$$\begin{aligned} 1 - \bar{\kappa}(x) &= 0, \\ 1 - 2\kappa(x) + \bar{\kappa}(x) &= 0 \end{aligned} \tag{2.38}$$

as it can be calculated from the determinant of the Jacobian matrix.
Both these equations describe circumferences, that are respectively mapped into a point and into a circular radial caustic by the lens equation. If the source is located within the radial caustic, it produces three images, while it only produces one image otherwise. If source, lens and observer are perfectly aligned an Einstein Ring forms, as it was explained in Section 2.1.5.

Power Law Lens

Power Law lenses are characterised by the following set of equations:

$$\begin{aligned} \kappa(x) &= \frac{3-n}{2}x^{1-n}, \\ m(x) &= x^{3-n}, \\ \alpha(x) &= x^{2-n}, \\ \gamma(x) &= \frac{n-1}{2}x^{n-1}, \\ \det A &= (1 - x^{1-n})(1 - (2-n)x^{1-n}) \end{aligned} \tag{2.39}$$

where the exponent n may be varied to better describe different objects:

a) $n < 1$: the radial profile of the convergence grows with the radius, which is a non-physical behaviour;

b) $n = 1$: this is the case of a completely convergent lens, characterised by a constant κ profile. Every light ray is mapped into $y = 0$;

c) $1 < n < 2$: the radial profile of the convergence decreases with the radius;

d) $n = 2$: this particular case describes the *Singular Isothermal Sphere* (SIS), that represents an ideal gas in thermal and hydrostatic equilibrium confined by a circularly symmetrical potential;

e) $2 < n < 3$: the deflection angle diverges at the origin;

f) $n = 3$: this is the case of the *point lens*;

g) $n > 3$: $m(x)$ decreases with the radius, which is a non-physical behaviour.

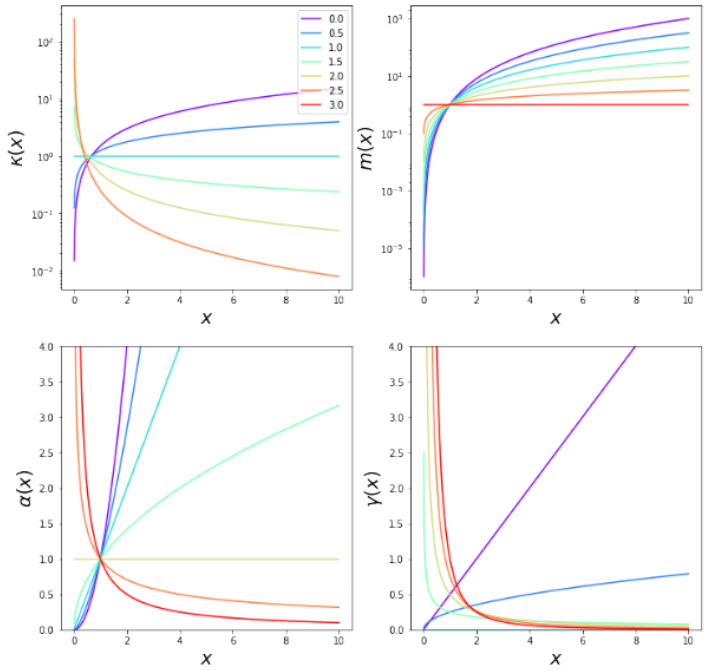


Figure 2.5: Radial profiles of the convergence (upper left), dimensionless mass (upper right), deflection angle (lower left) and shear (lower right) of a Power Law Lens for different values of n, $0 < n < 3$. Figure from Meneghetti, M. (2018).

The radial profiles of the deflection angle, the dimensionless mass, the convergence and the shear for several values of n are illustrated in Figure 2.5.
As shown in equation (2.29) the tangential and radial critical lines can be obtained as the points where the eigenvalues of the Jacobian matrix are null.
In this case, for the tangential critical line we get:

$$\lambda_t = 1 - x^{1-n} = 0 \quad \Rightarrow \quad x_{cr,t} = 1 \quad \Rightarrow \quad y_{cau,t} = 0. \tag{2.40}$$

The tangential critical line, i. e. the Einsten ring, is independent on x and is mapped into a point on the source plane.
On the other hand, for the radial critical line we find:

$$\lambda_r = 1 - (2-n)x^{1-n} = 0 \quad \Rightarrow \quad x_{cr,r} = (2-n)^{\frac{1}{n-1}}. \tag{2.41}$$

The dimensions of the radial critical line depend on n. As n increases the radial critical line becomes smaller and it is null for $n = 2$, while the opposite of this trend is true for the caustic. There are no radial critical lines for $n > 2$, therefore lenses described by a Power Law model with this exponent do not have radial critical lines, and this is relevant when studying the number of multiple images yielded in a given configuration, as seen in Section 2.1.5.

The Navarro Frenk & White density profile

The NFW density profile was introduced in Section 1.3.1 as a common one to describe the distribution of the Dark Matter within galaxy clusters.

If we take $\xi_0 = r_s$, the density profile 1.56 corresponds to the surface mass density

$$\Sigma(x) = \frac{2\rho_s r_s}{x^2 - 1} f(x) \tag{2.42}$$

where

$$f(x) = \begin{cases} 1 - \frac{2}{\sqrt{x^2-1}} \arctan \sqrt{\frac{x-1}{x+1}} & \text{if } x > 1 \\ 1 - \frac{2}{\sqrt{x^2-1}} \operatorname{arctanh} \sqrt{\frac{1-x}{x+1}} & \text{if } x < 1 \\ 0 & \text{if } x = 1 \end{cases} . \tag{2.43}$$

The lensing potential is

$$\Psi(x) = 4\kappa_s g(x) \tag{2.44}$$

where $\kappa_s \equiv \rho_s r_s \Sigma_{cr}^{-1}$ and

$$g(x) = \frac{1}{2} \ln^2 \frac{x}{2} + \begin{cases} 2 \arctan^2 \sqrt{\frac{x-1}{x+1}} & \text{if } x > 1 \\ -2 \operatorname{arctanh}^2 \sqrt{\frac{1-x}{x+1}} & \text{if } x < 1 \\ 0 & \text{if } x = 1 \end{cases} . \tag{2.45}$$

This implies the deflection angle

$$\alpha(x) = \frac{4\kappa_s}{x} h(x) \tag{2.46}$$

with

$$h(x) = \ln \frac{x}{2} + \begin{cases} \frac{2}{\sqrt{x^2-1}} \arctan \sqrt{\frac{x-1}{x+1}} & \text{if } x > 1 \\ \frac{2}{\sqrt{x^2-1}} \operatorname{arctanh} \sqrt{\frac{1-x}{x+1}} & \text{if } x < 1 \\ 1 & \text{if } x = 1 \end{cases} . \tag{2.47}$$

Finally, the convergence is given by

$$\kappa(x) = 2\kappa_s \frac{f(x)}{x^2 - 1} \tag{2.48}$$

that gives the dimensionless mass

$$m(x) = 2 \int_0^x \kappa(x') x' dx' = 4\kappa_s h(x). \tag{2.49}$$

2.2.2 Elliptical models

Circularly symmetric models are too simplistic to describe the properties of real extended lenses. The addition of two more parameters, the ellipticity and the position angle, that defines the orientation of the lens, yield slightly less idealized models. An axially symmetric model with surface density $\Sigma(x)$ may be generalized to an elliptical model by the substitution

$$x \to X = \sqrt{x_1^2 + f x_2^2} \tag{2.50}$$

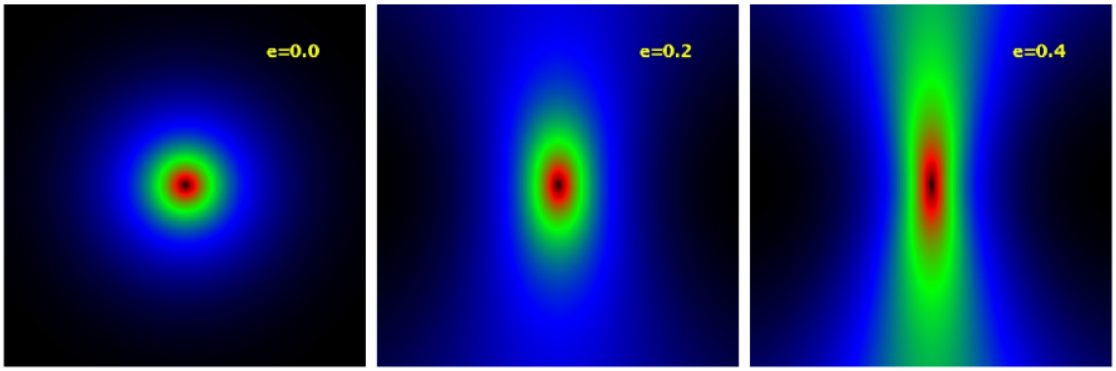


Figure 2.6: Deflection angle maps generated by a NFW profile with ellipticity $\epsilon = 0$ (i.e. axially symmetric profile, left), $\epsilon = 0.2$ (center) and $\epsilon = 0.4$ (right). Figure from Meneghetti, M. (2018) .

where $e = b/a$, a and b major and minor axis of the ellipse, respectively.
However, this approach may result into complicated calculations to derive the expression of the lensing potential.
It is often simpler to model a lens by means of an elliptical lensing potential $\Psi(x)$, following the same substitution in Equation 2.50.
The disadvantage of this approach is that the deflection angle field becomes dumbbell-shaped even for moderate values of ellipticity, which is often unwanted.
Figure 2.6 shows an example of how the deflection angle field changes according to different values of the ellipticity, in the case of the NFW density profile.

2.2.3 External perturbations

In order to obtain more realistic models, the effect of the objects surrounding the lens should be taken into account. In fact, lenses are often located in dense environments, where they are surrounded by other massive structures, such as galaxies in galaxy clusters. It is not feasible to describe each substructure as a mass distribution, because it would require too many parameters.
It is common practice to consider the effect of the surrounding lenses with an *external shear* represented by a potential Ψ_γ, that satisfies three conditions:

$$\begin{aligned} \gamma_1 &= \frac{1}{2}(\Psi_{11} - \Psi_{22}) = constant; \\ \gamma_2 &= \Psi_{12} = constant; \\ \kappa &= \frac{1}{2}(\Psi_{11} + \Psi_{22}) = constant. \end{aligned} \tag{2.51}$$

Since both $\Psi_{11} \pm \Psi_{22}$ must be constant, Ψ_{11} and Ψ_{22} have to be constants, and the potential will take the form

$$\Psi_\gamma = Cx_1^2 + C'x_2^2 + Dx_1x_2 + E. \tag{2.52}$$

The substitution of Equation 2.52 into the conditions 2.51 gives

$$\begin{aligned} \gamma_1 = \frac{1}{2}(\Psi_{11} - \Psi_{22}) = C - C' = \gamma_1, \\ \gamma_2 = \Psi_{12} = D = \gamma_2, \\ \kappa = \frac{1}{2}(\Psi_{11} + \Psi_{22}) = C + C' = \kappa. \end{aligned} \tag{2.53}$$

Imposing $\kappa = 0$, we obtain $C = C'$, hence $C = \frac{\gamma_1}{2}$. Thus, the external potential will be

$$\Psi_\gamma = \frac{\gamma_1}{2}(x_1^2 - x_2^2) + \gamma_2 x_1 x_2. \tag{2.54}$$

Similarly, one could place the lens on a *sheet of constant surface density*, with null shear, and Equation 2.51 would yield

$$\Psi_\kappa = \frac{\kappa}{2}(x_1^2 + x_2^2). \tag{2.55}$$

The addition of external perturbations modifies the lensing properties of the isolated lens.
In particular, if the perturbations are modeled by a constant external shear, they produce an additional deflection of the light rays

$$\vec{\alpha}_\gamma = \nabla\Psi_\gamma = \begin{pmatrix} \gamma_1 & \gamma_2 \\ \gamma_2 & -\gamma_1 \end{pmatrix} \vec{x}. \tag{2.56}$$

On the other hand, if they are modeled by a sheet of constant mass density, the addition to the deflection angle is of the type

$$\vec{\alpha}_\kappa = \nabla\Psi_\kappa = \kappa\vec{x} \tag{2.57}$$

and the lens equation is reads

$$\vec{y} = \vec{x}(1 - \kappa). \tag{2.58}$$

If the external sheet is critical (i.e. $\Sigma = \Sigma_{cr} \Rightarrow \kappa = 1$), the lens focuses all of the light rays in the origin, so $y = 0$ for all of the images.
Modeling external perturbations by means of an external shear is more common than doing it by means of the convergence, because the substructures around the main lens are able to induce an additional deflection on the light rays.

2.3 Lensing by galaxies and galaxy clusters

The gravitational lensing of background galaxies by galaxy clusters represents a solid method to recover the mass distribution of Dark Matter in the latter. In particular, the position of multiple images of the same source and their distortion give valuable information to constrain the mass distribution in the core regions.
As it was explained in the Section 2.2.3, however, the lens effect caused by the cluster potential well is modified by the individual galaxies within it, that induce perturbations in the large-scale mass distribution. These alterations have to be taken into account, and may provide additional information.
It is possible to distinguish between two main classes of events, according to the relative position of the observer, the lens and the source:

- *Strong lensing events* may occur when the observer, the lens and the source are well aligned along the line of sight, i.e. if the angular distance between the center of the lens and the source is small;

- *Weak lensing events* may be observed when the angular separation between the lens and the source projected on the sky is large.

These two regimes are characterised by different observables and properties, that will be discussed in the following Sections.

2.3.1 Strong Lensing

Strong lensing events are frequently observed in the central regions of galaxies and rich galaxy clusters, since they are favoured in dense environments. They occur when the lens develops extended critical lines, and the source, that may be point-like or extended, falls within the corresponding caustics or intersects them.
According to the geometrical configuration of the system, namely the relative positions of the source and the lens, the resulting event may show different characteristics.
Firstly, multiple images of the same background source may be observed. Since gravitational lensing is an achromatic effect, i.e. it acts in the same way at all wavelengths, images corresponding to the same source can be identified by their spectrum, whose features are preserved. The mass distribution of the inner regions of the lens can be constrained by the number of images and their displacement.
Secondly, if the source is extended, its images may be highly distorted by the differential deflection of the light in the tangential or in the radial direction. One of the most common types of distortion is the gravitational arcs observed in the cores of massive galaxy clusters.
In order to trace the mass distribution in the core of the lens, a *mass model* has to be calculated. This is usually done trying to reproduce the observational constraints, that can be of three different types.
Astrometric constraints are based on the measurements of the relative positions of the lens and the images, whose separation can usually be measured accurately. This information is related to the deflection angle, i.e. the the first derivative of the lensing potential. Moreover, *photometric constraints* can be retrieved evaluating the relative fluxes of different images, that are determined by the magnification on the lens plane, hence by the second derivatives of the potential. These measurements are more difficult to carry out and many factors, such as time variability in the source, lensing by substructures in the extended lens, have to be taken into account, so these constraints are not as accurate as the former ones. Finally, *time delay* measurements between multiple images can be used to constrain the mass distribution of the lens. In particular, they are mainly used when the lens is a galaxy, in which case they amount to months, while they are of the order of decades or centuries when dealing with galaxy clusters.
Building a lens model of an extended lens requires some effort, since they are complicated structures, perturbed by substructures and described by huge sets of parameters. According to this approach, the best model is found by defining the

combination of parameters that best describes the observed lensed features. Thus, the model will be optimized by minimizing the χ^2 function.
For instance, if one has constraints regarding the position of the multiple images corresponding to one source, the χ^2 function will be

$$\chi_i^2(\vec{p}) = \sum_{j=1}^{N_{im}} \frac{\vec{\theta}^j_{obs} - \vec{\theta}^j_{pred}(\vec{p})}{\sigma_{ij}^2} \tag{2.59}$$

where the index j refers to the multiple images of the same source i; $\vec{\theta}^j_{obs}$ is the observed position and $\vec{\theta}^j_{pred}$ is the position predicted by the model for a given image j; σ_{ij} is the uncertainty associated to the position; $\vec{p}$ is the set of parameters that characterizes a model and N_{im} is number of images produced by the source i.
An example of strong lensing analysis of a galaxy cluster, that aims to recover its mass distribution, may be found in Caminha et al. (2017). This work employed positional measurements of 82 multiple images belonging to 27 different families[1] to reconstruct the total mass distribution of MACS 1206. Figure 2.7 shows the multiple images divided according to whether they are produced by a tangential or radial critical line.
However, different modeling techniques have been developed over the years to map the Dark Matter distribution in the cluster lens. A description and comparison of various parametric, free-form and hybrid techniques can be found in the work by Meneghetti et al. (2017).

Galaxy-Galaxy Strong Lensing events

Galaxy-Galaxy Strong Lensing (GGSL) events occur when a background source falls within the caustic of a foreground galaxy. It is more likely to observe them in dense environments, such as galaxy clusters, whose high mass density boosts the strong-lensing cross-section of individual galaxies (Desprez et al., 2018).
The strong lensing features are expected to appear around the critical lines of the lens. Hence, while the angular separation of the distorted images of a source produced by a cluster lens is of the order of tens of *arcsec*, the multiple images generated by the cluster members are usually less than a few *arcsec* apart.
Figure 2.8 shows a collection of images of Galaxy-Galaxy Lenses found in the Cluster Lensing And Supernovae Survey with Hubble[2] (CLASH).
The analysis of this type of events may provide valuable information on several accounts. For instance, they are useful in the reconstruction of the mass distribution of both the hosting cluster at large scales and the lensing galaxies (Desprez et al. 2018; Bergamini et al. 2019, 2020) and they have been used to constrain cosmological parameters (see e.g. Collett & Auger 2014).
Even though GGSL events are rare, recent studies have shown that they are relatively abundant in the core of some massive clusters (few per cluster). In fact, Meneghetti et al. (2020) have found that the substructures in galaxy clusters produce about an order of magnitude more events than expected from cosmological

[1] A family of multiple images is the ensemble of the images produced by the deflection of the light of one single source.
[2] https://www.stsci.edu/~postman/CLASH/

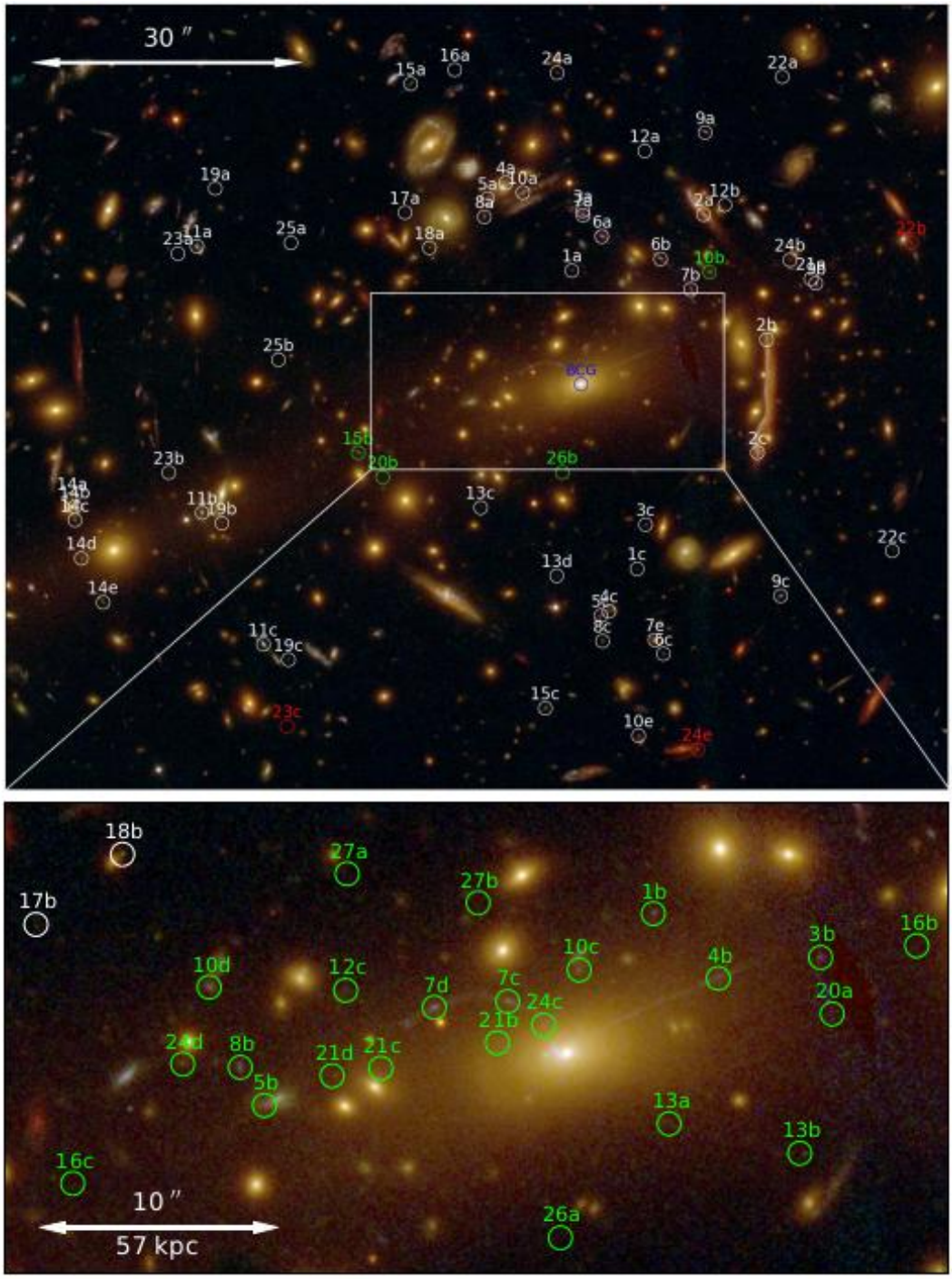


Figure 2.7: Spectroscopically confirmed families of multiple images in MACS J1206 overlaid on a color composite image based on 12 CLASH filters. White and green circles indicate, respectively, the tangential and radial images. Image from Caminha et al. (2017).

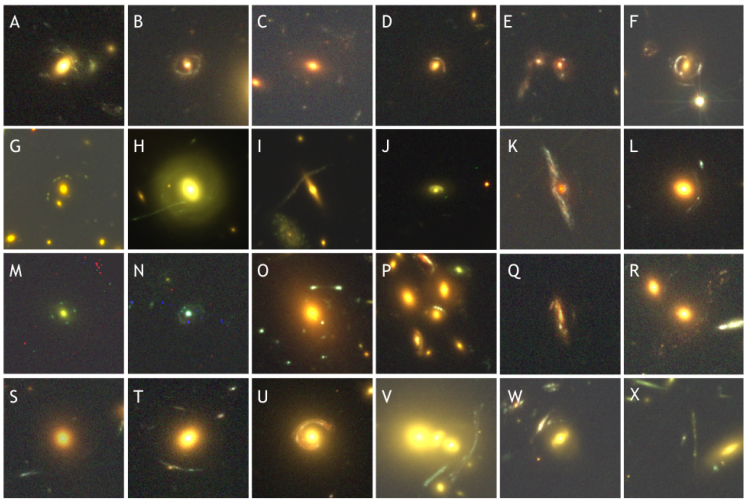


Figure 2.8: Galaxy-Galaxy Lenses found in the CLASH. Image from Desprez et al. (2018).

simulations. Figure 2.9 compares the probability of observing a GGSL event in simulated clusters and in real ones, as a function of the redshift of the source.
However, it is also expected that the number of GGSL systems will increase of several orders of magnitude in upcoming imaging surveys (Collett, 2015), such as those that will be carried out by the Euclid mission[3] and by the Nancy Grace Roman Space Telescope[4]. The identification of potential candidates will require the inspection of tens of millions of images.

2.3.2 Weak Lensing

The measurements regarding weak gravitational lensing effects are statistical and difficult to carry out. In fact, in this regime the shape of the sources may be only subtly altered and it is impossible to distinguish the distortions from the intrinsic shape of a single source. However, the distortion is coherent in a region surrounding the lens, so it can be detected by studying the shape distribution of an ensemble of galaxy images (Bartelmann & Schneider, 2001).
Although it is challenging to detect them, weak lensing effects allow to determine the mass distribution of lenses at large distances from the center of the cluster (see e.g. Giocoli et al. 2014; Okabe et al. 2010), making it possible to prove cosmological parameters and dark matter models, that predict different masses for clusters, as explained in Chapter 1.
Assuming that the orientation of distant, faint and irregularly-shaped sources is random, the average shape of a large number of them should be circular. However, as it was explained in Section 2.1.4, because of weak lensing, the circular source appears to be elliptical, with axes:

$$a = \frac{r}{1-\kappa-\gamma}, \qquad b = \frac{r}{1+\kappa+\gamma} \tag{2.60}$$

[3] https://www.euclid-ec.org/
[4] https://www.nasa.gov/content/goddard/nancy-grace-roman-space-telescope

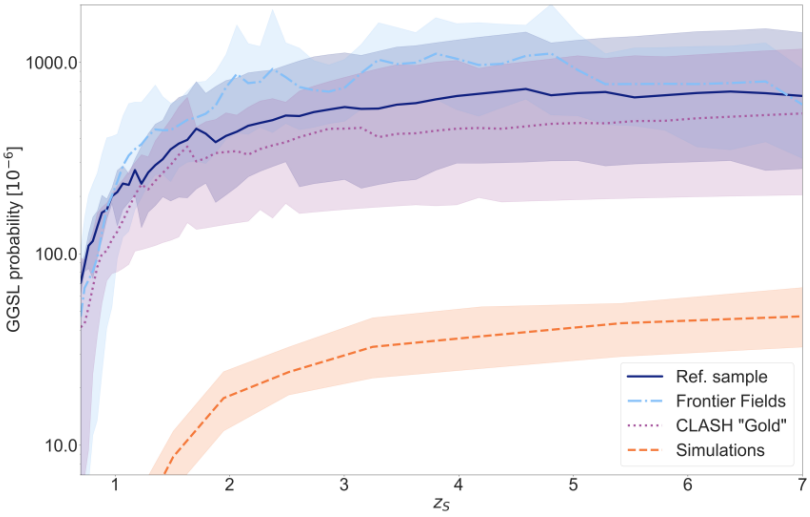


Figure 2.9: The GGSL events probability as a function of the redshift of the source. Image from Meneghetti et al. (2020).

and the ellipticity can be measured as

$$\epsilon = \frac{a-b}{a+b} = \frac{2\gamma}{2(1-\kappa)} \approx \gamma \tag{2.61}$$

where $\kappa, \gamma \ll 1$ in the weak lensing regime.
The ellipticity may also be defined by means of a second order tensor that describes the brightness moments on the source and on the lens planes:

$$\begin{aligned} Q_{ij} &= \frac{\int d^2\theta I(\theta) q_I[I(\theta)](\theta_i - \bar{\theta}_i)(\theta_j - \bar{\theta}_j)}{\int d^2\theta I(\theta) q_I[I(\theta)]}, \\ Q^{(s)}_{ij} &= \frac{\int d^2\beta I^{(s)}(\beta) q_I[I^{(s)}(\beta)](\beta_i - \bar{\beta}_i)(\beta_j - \bar{\beta}_j)}{\int d^2\beta I^{(s)}(\beta) q_I[I^{(s)}(\beta)]} \end{aligned} \tag{2.62}$$

where q_I is a weight function that selects the scale covered by the galaxy, $I(\theta)$ and $I^{(s)}(\beta)$ are the brightness functions on the lens and on the source planes respectively, and $\bar{\theta}$, $\bar{\beta}$ are the image centroids:

$$\bar{\theta} = \frac{\int d^2\theta I(\theta) q_I[I(\theta)]\theta}{\int d^2\theta I(\theta) q_I[I(\theta)]}, \qquad \bar{\beta} = \frac{\int d^2\beta I^{(s)}(\beta) q_I[I^{(s)}(\beta)]\beta}{\int d^2\beta I^{(s)}(\beta) q_I[I^{(s)}(\beta)]} \tag{2.63}$$

It can be proven that, writing the ellipticity as a complex value, with $|\epsilon| = \sqrt{\epsilon\epsilon^*} = \sqrt{\epsilon_1^2 + \epsilon_2^2}$, one finds

$$\epsilon = \frac{Q_{11} - Q_{22} + 2iQ_{12}}{Q_{11} + Q_{22} + 2(Q_{11}Q_{22} - Q_{12}^2)^{1/2}}. \tag{2.64}$$

In the same way, the intrinsic ellipticity is described by $Q^{(s)}_{ij}$. The observed ellipticity on the lens plane and the intrinsic ellipticity on the source plane are related through the lens equation, i.e. in the first order approximation $\beta = A\theta$:

$$Q^{(s)} = AQA^T = AQA. \tag{2.65}$$

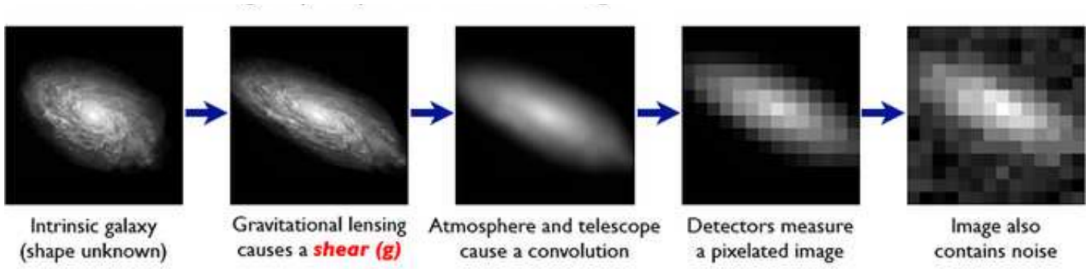


Figure 2.10: This sequence of images shows how the intrinsic shape of a galaxy is modified into the one observed by a ground-based telescope. Image from Meneghetti, M. (2018).

Combining equations (2.25), (2.64) and (2.65), one finds:

$$\epsilon^{(s)} = \begin{cases} \frac{\epsilon - g}{1 - g^* \epsilon} & \text{if } |g| \leq 1 \\ \frac{1 - g\epsilon^*}{\epsilon^* - g^*} & \text{if } |g| > 1 \end{cases} , \tag{2.66}$$

where

$$g = \frac{\gamma}{1 - \kappa} \tag{2.67}$$

is the *reduced shear*.
When averaging over a large sample of galaxies, it is expected that the average value of the intrinsic ellipticity will be null, so

$$\epsilon = \begin{cases} g & \text{if } |g| \leq 1 \\ \frac{1}{g^*} & \text{if } |g| > 1 \end{cases} . \tag{2.68}$$

Therefore, the ellipticity is related to the reduced shear. A detailed explanation of the procedure for the reconstruction of the mass distribution starting from the shear profile can be found in the review by Bartelmann & Schneider (2001), that also provides a thorough description of the effects of the density perturbations on the propagation of light on cosmological scales.
However, weak lensing is not the only cause of distortion in the shapes of galaxies images. Other factors, such as the atmosphere and the Point Spread Function[5] (PSF) of the instrument must be taken into account. Figure 2.10 schematically shows how the intrinsic shape of a galaxy may be distorted by different causes.
The Keiser Squires Broadhurst (KSB) method (Kaiser et al., 1995) provides an effective algorithm to isolate the signal generated by the weak lensing effect from the other contributions.

[5]The Point Spread Function describes the response of an imaging system to a point source.

Chapter 3

Machine Learning

Machine Learning (ML) is the capability of an Artificial Intelligence (AI) system to acquire their own knowledge, by extracting patterns from raw data (Goodfellow et al., 2016). ML methods have become increasingly popular in the last few decades because they allow to work on complicated problems, that would not be possible to tackle by means of other types of algorithm.
A great variety of tasks often performed in astrophysical analyses and, with a broader perspective, in scientific studies can be approached with a ML technique. Some of the most common ones are:

- *Classification*: the data is divided into n categories and the algorithm is either required to assign some input to the class it is part of or to calculate the probability that the input belongs to each one of the possible classes, in order to know which one is the most likely to be the correct one;

- *Regression*: given some input, the program is asked to predict a real number, that represents a specific quantity;

- *Clustering*: the aim of the algorithm is to divide the dataset into subsets of objects that are similar between them, but different from the objects belonging to the other clusters.

Moreover, ML algorithms can practically divided into two classes:

1. *Supervised learning algorithms* learn an arbitrary mapping from input data to output labels, observing a large number of training examples. In particular, they examine numerous instances of a vector $\vec{x}$ associated to some vector $\vec{y}$ and learn to predict $\vec{y}$ from $\vec{x}$, by estimating the conditional probability $p(\vec{y}|\vec{x})$[1];

2. *Unsupervised learning algorithms* learn meaningful patterns and characteristics of the structure of a given dataset without being provided with explicit labels. Specifically, these programs inspect a considerable amount of examples of a given vector $\vec{x}$ and attempt to learn the probability distribution $p(\vec{x})$ or some interesting properties of it.

[1]The conditional probability $p(\vec{y}|\vec{x})$ is the probability of $\vec{y}$ given that $\vec{x}$ is true.

For instance, classification and regression problems are usually dealt with algorithms of the first kind, while clustering problems are approached with algorithms of the second type.

With the growth of astronomical datasets in volume and complexity, the exploitation of ML techniques for data mining has also become necessary in Astrophysics (see e.g. Baron 2019 for an introduction, Huerta et al. 2019). The need to further develop reliable ML methods in the field of astronomical data analysis will only increase in the years to come, because of the amount of data expected to be gathered by future surveys, such as the ones listed in 2.3.1.

Several tasks have already been addressed with ML algorithms in the context of astrophysical research. Among these are image classification in Gravitational Lensing studies (see e.g. Schaefer et al. 2018; Davies et al. 2019), the assessment of the photometric redshift of luminous sources (see e.g. Sadeh et al. 2016), the dissection of degeneracies in cosmological models (see Merten et al. 2019), the identification of cluster members in galaxy clusters (see e.g. Angora et al. 2020) and the estimation of the dynamical mass of galaxy clusters (see e.g. Ntampaka et al. 2015; Kodi Ramanah et al. 2020).

This work employs a supervised learning method, that will be introduced in the next Sections for image classification, while unsupervised learning techniques will not be addressed. However, a valid introduction to this topic may be found in the work by Rojas (2009).

3.1 Supervised Machine Learning

Supervised ML algorithms are designed to learn an unknown mapping from known input-output pairs. Subsequently, this mapping can be applied to never seen before data of the same kind as the inputs analysed by the algorithm, in order to predict the corresponding and unknown output, without following specific instructions.

In particular, the relationship between the input and the output is provided by an element called *model*. Several types of ML models, such as Multilayer Perceptron (MLP, Hastie et al. 2009), Long Short-Term Memory (LSTM, Gers et al. 2002) and Convolutional Neural Networks (CNN, LeCun & Bengio 1998) are composed of layers. Each layer is defined by a set of parameters and has a precise role in the definition of the model.

Moreover, a particular instance of the dataset is called *example*. As it was previously said, the peculiarity of supervised learning is the model experiences a phase during which each example is associated to a corresponding *label*, that is the output sought by the model. For instance, in classification problems, the label would be the correct group that the example should be placed in, while in regression problems it would be the numerical value that the algorithm is required to predict.

The stage in which the model analyzes the input-output pairs and readjusts itself to improve its capability to predict the output is called *training*. Before being applied to new data, the model also goes through a *test* phase, in which its predictions on unlabeled data are compared to the known corresponding labels and its performance is assessed.

However, the implementation and training process of the models will be discussed in greater detail in Sections 3.2 and 3.3, while some of the metrics commonly employed to evaluate the performance of a supervised model in a classification problem will be described in Chapter 5.
Going back to the basic components of a ML algorithm, the input variables are usually referred to as *features*. Choosing features that adequately represent the characteristics of the data and provide means of solving the problem is an important part of the implementation of a ML algorithm. If the features do not describe the dataset appropriately, the execution of a certain task might become more difficult or impossible.

3.1.1 Deep Learning

Deep Learning (DL) refers to a class of ML techniques, that exploits complex models with multiple layers to tackle difficult problems.
The depth of the model is related to its capacity of extracting complex features from the data and of generalizing well on other data. In this regard, the automatization of feature extraction is the greatest advantage of DL methods with respect to ML methods.
In particular, DL techniques resolve the issue of learning a complicated mapping by dividing it into a series of nested simple mappings, each described by a different layer of the model (Goodfellow et al., 2016). In other terms, the function that maps the input into the output is decomposed into simpler functions, each of whom provides a new representation of the data. We will clarify this concept in Section 3.2.
Further observations and remarks on the theoretical reasons for DL algorithms and on their practical applications may be found in the work by Deng (2014), while a more comprehensive introduction to the topic is in the book by Goodfellow et al. (2016).

3.2 Neural Networks

Neural Networks (NNs) are ML and DL models, whose operating principle is summarily inspired by the functioning of the neurons in the human brain. In particular, they are constituted by simple processing units, also called neurons, nodes or *perceptrons* (Bishop, 2006), that are linked through connections and organised in layers.
While the architecture of the network, i.e. the ensemble of the layers that define it, is decided by the programmer in the implementation phase, the weights, that indicate the sensitivity of the connections between the individual nodes (Hebb, 1949), are adjusted by the learning algorithm in the training phase to improve the capability of the model to make the correct predictions.
NNs have at least three layers: the input layer, the output layer and one hidden layer. However, networks generally have several hidden layers, capable to extract high-level features from the data. In particular:

- The input layer, also called visible layer, is the one where the data is passed to the network;

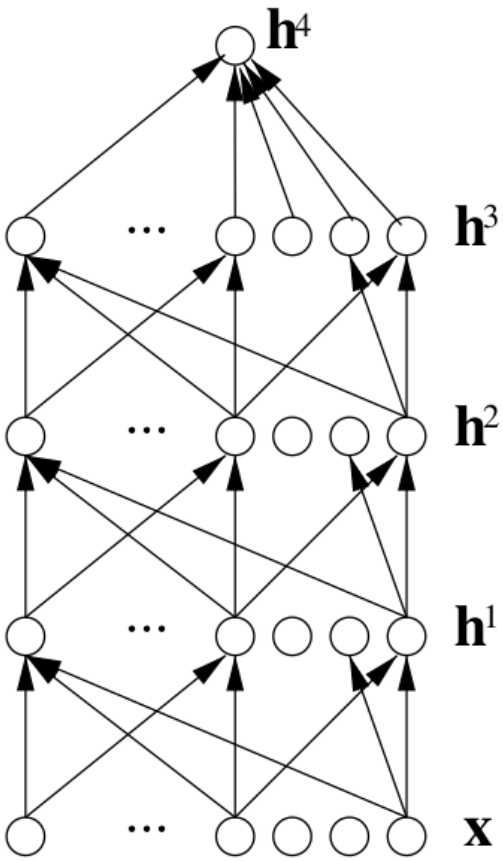


Figure 3.1: Architecture of a generic network with three hidden layers: in this example $\mathbf{x}$ is the input layer, $\mathbf{h}^i, i = 1, 2, 3$ are the hidden layers and $\mathbf{h}^4$ is the output layer. The scheme also shows how the units of each layer are connected to the ones of the lower and higher level layers. Image from Bengio (2009).

- The hidden layers have the purpose of extracting abstract features from the data. Their number and the number of nodes within each layer is decided a priori, while their values are modified throughout the training stage, after a random initialization;

- The output layer is the final layer of the network and it provides some additional transformation from the features, finally fulfilling the required task. The labels associated to the training data directly specify what the behaviour of this layer should be: the learning algorithm will adjust the hidden layers in order to approximate the desired output in the best possible way.

Moreover, every model is characterised by two dimensions: the *depth* and the *width*, namely the total number of layers and the number of neurons within the same layer, respectively.
Figure 3.1 shows the architecture of a multilayer Neural Network with three hidden layers and the structure of the connections between the units that belong to different layers.

Each layer is composed of several nodes connected to the ones of the previous layer through weighted connections, that describe how the input is propagated through the network.
In fact, the output of the k-th layer $\mathbf{h}^k$[2] can be calculated using the output of the previous layer $\mathbf{h}^{k-1}$:

$$\mathbf{h}^k = f(\mathbf{b}^k + W^k \mathbf{h}^{k-1}) \tag{3.1}$$

where $\mathbf{b}$ and W^k are, respectively, the vector of offsets (biases) and the weight matrix associated to the layer: their dimension is defined by the number of units in the layer. The first layer is given by the input: $\mathbf{x} = \mathbf{h}^0$, while the last layer $\mathbf{h}^L$ is used to make a prediction. The notation $\mathbf{z}^k = \mathbf{b}^k + W^k \mathbf{h}^{k-1}$ will be adopted later in this Chapter.
Finally, f is an activation function: it adds non-linearity to the network, that would otherwise only be characterised by linear operations. Some examples of often employed activation functions are listed in Table 3.1, while more information on this topic may be found in the work by Szandała (2021).

Table 3.1: List of activation functions commonly used in Neural Networks and their equation.

Activation function	Equation
Rectified Linear Unit (ReLU)	$f(x) = \max(0, x)$
Leaky ReLU	$f(x, \alpha) = \begin{cases} \alpha x, & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$
Sigmoid	$\text{sigmoid}(x) = \dfrac{1}{1 + \exp(-x)}$
Hyperbolic tangent	$\tanh(x) = \dfrac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$

Sometimes, in the last layer of the network a different activation function is used. For instance, the softmax function is usually used in the output layer of a NN whose task is to classify the input into different categories.

$$\mathbf{h}_i^L = \frac{e^{\mathbf{b}_i^L + W_i^L \mathbf{h}^{L-1}}}{\sum_j e^{\mathbf{b}_j^L + W_j^L \mathbf{h}^{L-1}}} \tag{3.2}$$

where the sum in the denominator is over all the neurons in the output layer L and i represents a generic neuron. The softmax output $\mathbf{h}_i^L$ can be interpreted as an estimator of $P(Y = i|\mathbf{x})$, where Y is the class associated with the input $\mathbf{x}$ (Bengio, 2009).
The models in which the input is propagated through the hidden layers to the output layer in the way described by Equation 3.1 are also called *feed-forward Networks*.

[2]From this moment on, vectors will be indicated as **bold** characters, to improve clarity.

3.3 Training process

Once the architecture of the model has been decided and implemented, the network has to be trained before it can be applied to any data: the aim of this procedure is to determine the best possible values for the weights and the offsets.
In fact, as it was previously said, in the implementation of the network only the number of parameters per layer is defined, but their values are readjusted in the training phase, according to the effectiveness shown by the model in predicting the correct output.
The training process is hence iterative: each time all of the examples in the training set are processed by the network defines an *epoch*. Generally, a great number of epochs is necessary for the model to converge to the best configuration of parameters. The weights and biases are updated at the end of each epoch, according to the performance of the Network, that is assessed by means of a *loss function*, that combines the predictions of the Network and the correct outputs $\mathcal{L}(\mathbf{h}^l, y)$.
The most relevant loss function in the context of this work is the Binary Cross-Entropy (Goodfellow et al., 2016), that is especially useful in binary classification problems, where the label associated to each example is either 0 or 1:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y(\mathbf{x}_i) \cdot \log[y_p(\mathbf{x}_i)] + (1 - y(\mathbf{x}_i)) \cdot \log(1 - y_p(\mathbf{x}_i)) \tag{3.3}$$

where N is the number of training examples, y is the ground truth and y_p is the probability that the i-th example has label 1 as predicted by the network.
Some other examples of loss functions are listed in Table 3.2. Specifically, the Categorical Cross-Entropy is used in classification problems where the input might belong to one of $M > 2$ classes, while the Mean Squared Error and the Mean Absolute Error are commonly employed in regression problems.

Table 3.2: Loss functions used in the training process of Neural Networks. The decision of the loss function to use depends on the task that the Network has to perform.

Loss function	Equation
Categorical Cross-Entropy	$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_j(\mathbf{x}_i) \cdot \log[y_j(\mathbf{x}_i])$
Mean Squared Error (MSE)	$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y(\mathbf{x}_i) - y_p(\mathbf{x}_i))^2$
Mean Absolute Error (MAE)	$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\lvert y(\mathbf{x}_i) - y_p(\mathbf{x}_i)\rvert$

The aim of the training algorithm is to minimize the loss function, that is usually fairly complicated and might have local minima. In other words, the solution of

the learning problem is given by the combination of weights that minimizes the loss function.
One of the most common learning algorithms is the *backpropagation algorithm* (see e.g. Rojas (2009)), that allows to update the weights of the network in combination with an *optimization method*. In particular, the backpropagation algorithm refers to the method used for the computation of the gradient of the loss function with respect to the weights of the network, while the optimization method is used to actually update the weights and perform the learning.
The objective of the backpropagation algorithm is to compute the gradient of the loss function with respect to the weights and biases of the network, since the gradient gives an indication of how these parameters should be changed to minimize the difference between the predictions of the network and the known output.
In particular, this computation is performed by using the chain rule and is split into two steps, as described in the work by Nielsen (2015):

1. Forward propagation: the input $\mathbf{x}$ is propagated through the hidden layers, whose units compute an output value according to Equation 3.1. In the training procedure, the output of the final layer, i.e. the prediction of the Network, is used in combination with the ground truth to evaluate the loss function $\mathcal{L}$.

2. Backpropagation: the gradient of the loss function with respect to the output of the network is computed and the network is run backwards.

   In detail, we can define the error of the neuron i in the layer k as

   $$\delta_i^k = \frac{\partial \mathcal{L}}{\partial z_i^l}. \tag{3.4}$$

   Firstly, the error is evaluated on the output layer L as

   $$\delta_i^L = \frac{\partial \mathcal{L}}{\partial h_i^L} f'(z_i^L). \tag{3.5}$$

   The first term of this product takes into account how the loss function is affected by the output of the i-th neuron, while the second one considers how the activation function changes at z_i^L.

   Equation 3.5 can also be written in a vectorial form:

   $$\delta^L = \nabla_{\mathbf{h}} \mathcal{L} \odot f'(\mathbf{z}^L) \tag{3.6}$$

   where the operator $\odot$ stands for the element-wise product of the two vectors.

   In the learning procedure, this gradient is converted into a gradient with respect to the weights at each node and is propagated to the adjacent lower level hidden layer:

   $$\delta^k = ((W^{k+1})^T \delta^{k+1}) \odot f(\mathbf{z}^k) \tag{3.7}$$

   With this equation it is possible to compute the error at each layer k, until the input layer is reached. The rate of change of the loss function with respect to a generic weight w can finally be evaluated as

   $$\frac{\partial \mathcal{L}}{\partial w} = a_{in} \delta_{out} \tag{3.8}$$

where a_{in} is the activation of the neuron input and has been calculated in the forward step, while δ_{out} is the error of the neuron output and is evaluated in the backward step.

An analogous equation may be derived for the biases of the Network (Nielsen, 2015).

It should be noted that the gradient of the loss function passed to the optimizer is an average estimation of the gradients calculated for the training examples at the end of each epoch. In fact, the training dataset is usually split into several small *batches*, that include a given number of examples and is not passed to the network as a whole.

3.3.1 Optimization

The optimization is the method employed to update the weights and biases using the computed gradient. In this context, the *learning rate* is probably the most important hyper-parameter[3] for the success of the training process. It is used to define the step length of each update in the negative gradient direction. If it is too high, the model rapidly converges towards the minimum of the loss function, but it might not reach it exactly, while if it is too small, the model might get stuck into a local minimum or require too many epochs to converge.
Here follows a list of some common optimizers:

- The Gradient Descent (GD) updates the weights using an increment that is a fraction of the magnitude of the gradient of the loss function:

$$\mathbf{w}_{e+1} = \mathbf{w}_e - \gamma \nabla \mathcal{L}(\mathbf{w}_e) \tag{3.9}$$

  where e is the epoch considered and γ is the learning rate.

  When the training set is too large, it is difficult to compute the derivatives for each example. In this typical situation, a convenient variant of the GD is the Stochastic Gradient Descent (SGD), that estimates the derivative on a random subset of the whole dataset.

- Adaptive Gradient Algorithm (AdaGrad, Duchi et al. 2011) takes into account that in high-dimensionality problems a given value of the learning rate might be too high in a certain direction, while being too small in another one.

  Since it is quite unpractical to choose different learning rates for each possible dimension, this algorithm adaptively scales the learning rate in each dimension. The weights will be updated according to

$$w_{i,e+1} = w_{i,e} - \frac{\gamma}{\sqrt{\epsilon + G_{i,e}}} \frac{\partial \mathcal{L}}{\partial w_{i,e}} \tag{3.10}$$

[3]Hyper-parameters are parameters whose value is chosen manually and is not further adjusted during the training procedure.

where ϵ is a hyper-parameter used to avoid the division of zero and

$$G_{i,e} = \sum_{j=1}^{e} \left(\frac{\partial \mathcal{L}}{\partial w_{i,j}} \right)^2 = \sum_{j=1}^{e} g_{i,j}^2. \tag{3.11}$$

takes into account the variation of the loss function with respect to a particular weight in the past epochs.

The effective learning rate is hence dependent on the training itself. In particular, smaller updates are performed on the parameters associated with frequently occurring features, while larger updates are carried out on parameters associated with infrequent features. For this reason, this optimization method is well-suited for dealing with sparse data.

- Root Mean Square propagation (RMSprop, Hinton et al. 2012) is very similar to AdaGrad, but it also takes into account that during the training process, the summation in Equation 3.11 monotonically grows and, consequently, the effective learning rate decreases and eventually approaches 0.

  In order to address this issue, RMSprop recursively defines the sum of gradients as a decaying average of the past gradients:

$$v_e = E[g_i^2]_e = \eta v_{e-1} + (1 - \eta) g_{i,e}^2 \tag{3.12}$$

  where η is a hyper-parameter, usually set to 0.9, and v_e is used in place of $G_{i,e}$ in Equation 3.10.

- Adadelta (Zeiler, 2012) is very similar to RMSprop: it was also developed to address the problem of the diminishing learning rate of the AdaGrad optimizer.

  The main difference between RMSprop and Adadelta is that the latter does not require the definition of a default value for the learning rate in Equation 3.10. In fact, the weight update is performed following

$$w_{i,e+1} = w_{i,e} - \sqrt{\frac{D_e + \epsilon}{v_e + \epsilon}} \frac{\partial L}{\partial w_{i,e}} \tag{3.13}$$

  where

$$D_e = E[\Delta w_i^2]_e = \eta D_{e-1} + (1 - \eta) \Delta w_{i,e}^2. \tag{3.14}$$

  This algorithm basically uses the difference between the current weight and the newly updated one instead of the learning rate.

  Both v_e and D_e are initialized to 0.

- Adaptive moment estimation (Adam, Kingma & Ba 2017, Reddi et al. 2019) uses the first and second derivatives of the gradient of the loss function to update the effective learning rate.

$$w_{i,e+1} = w_{i,e} - \frac{\gamma}{\sqrt{\widehat{v}_e} + \epsilon} \cdot \widehat{m}_e \tag{3.15}$$

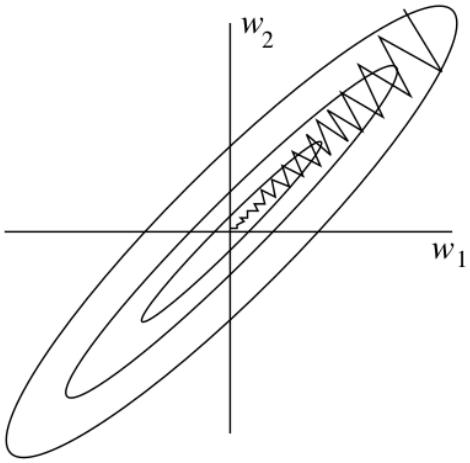


Figure 3.2: This figure shows the path followed by the learning algorithm to find the minimum of the loss function, here projected onto the parameter space. In this simplistic case the network is defined by two weights, w_1, w_2. The length of each step depends on the learning rate. Image from Rojas (2009).

where β_1, β_2 are two hyper-parameters and

$$\widehat{v}_e = \frac{v_e}{1-\beta_1^2} \qquad \widehat{m}_e = \frac{m_e}{1-\beta_2^2} \tag{3.16}$$

are the bias corrections and

$$\begin{aligned} m_e &= \beta_2 m_{e-1} + (1-\beta_2) g_{i,e} \\ v_e &= \beta_2 v_{e-1} + (1-\beta_2) g_{i,e}^2. \end{aligned} \tag{3.17}$$

The same equations are used to update the biases.
Figure 3.2 shows how following the gradient direction leads the optimization algorithm to find the minimum of the loss function in the simplistic case of a network with just two weights w_1, w_2.

3.3.2 Validation and Regularization

One of the most important properties of a NN is *generalization*, namely the capability of the network to correctly apply the knowledge it has gained from the training data on data it has never processed before.
In particular, to determine whether the model has good generalization capability, its predictions in the training phase are not only evaluated on the training set, but also on an independent dataset called validation set.
The validation set is usually composed of a small percentage of the training set (about 10-15%) and it must be representative of the properties of this dataset.
While the performance of the network on the training set determines how the weights and biases are updated, following the procedure outlined in the previous sections,

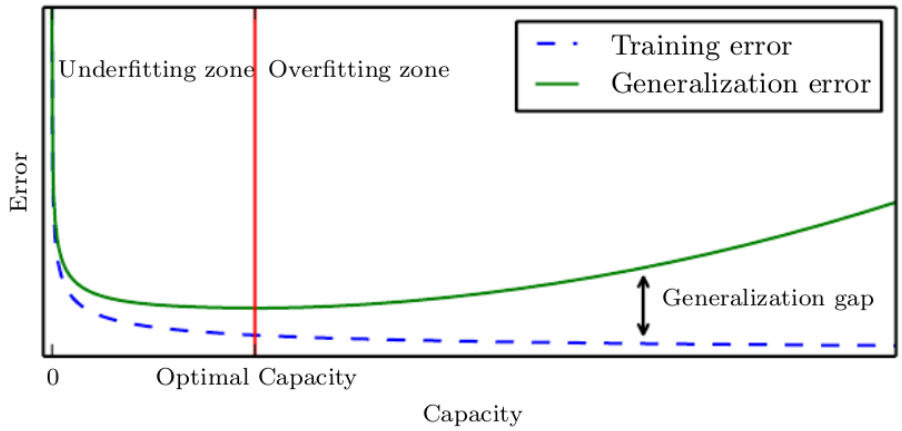
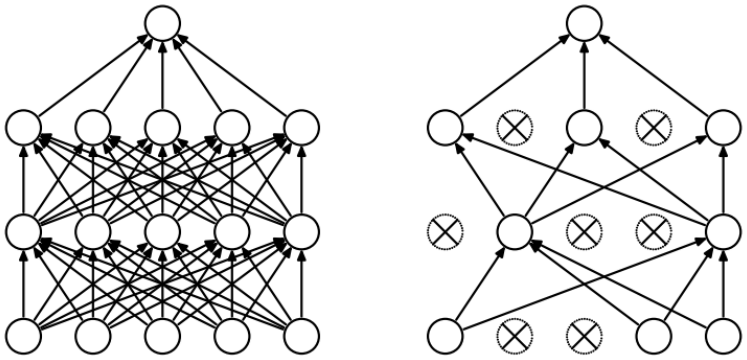


Figure 3.3: Relationship between the model capacity's and the error on the training (dashed blue line) and the validation (solid green line) set. The red vertical line corresponds to the optimal capacity of the model for this generic task: simpler models will underfit, while more complicated models will overfit. Image from Goodfellow et al. (2016).

the results on the validation set are used to decide whether the learning rate should be decreased or the training stopped. However, they are not used to calculate the loss function and hence do not influence the parameters' update.
Therefore, when training a ML algorithm, one has two main goals: the first one is to reduce the loss function on the training set and the second one is to reduce the gap between the training and the validation errors.
A very complex and deep model will likely perform well on the training set and the training error will decrease steadily in the training phase. However, there is a chance that it memorizes some properties that are not representative of the data, but of specific examples. In this case, the model will not improve in predicting the output of the validation set, because it will look for features that are not present in this data. This problem is called *overfitting*, and is characterized by a small value of the loss function on the training set, but a high one on the validation set.
On the other hand, if the model is too simple it will not be able to abstract complex properties from the training set, and will not perform well on either this or the validation set. In this case, the model is *underfitting*.
Both these problems may be addressed by trying to change the model's *capacity*, namely its ability of fitting a wide variety of functions (Goodfellow et al., 2016). The capacity of a model should be appropriate in regard to the complexity of the task it is required to perform. Figure 3.3 shows how a variation in the model's capacity affects the performance of the network on both the training and the validation set.
When training a deep network it is important to take precautions against overfitting, that might be favored by the complexity of the model. It is called *regularization* the ensemble of small modifications to the learning algorithm made in order to achieve a better performance on the validation set, without affecting the error on the training set (Goodfellow et al., 2016).
In this regard, several strategies can be opted for in the implementation of the network's architecture, in the data processing or in the definition of the learning



(a) Standard Neural Net (b) After applying dropout.

Figure 3.4: Neural Network with two layers before (a) and after (b) dropout has been applied. Crossed units in image (b) have been dropped. Image from Srivastava et al. (2014).

algorithm. Some of the most common are listed here:

1. An effective way of reducing overfitting is using a larger training dataset. In practice, this is not always feasible because one has a limited amount of data. However, it is sometimes possible to create fake data using *data augmentation* techniques.

   This strategy has proven particularly useful in image recognition: new images can be obtained by scaling, flipping, translating or rotating the existing ones.

2. *Dropout* randomly drops units from the network during training (Srivastava et al., 2014). This means that the units are temporarily removed from the network, along with their incoming and outcoming connections. The units to drop are chosen randomly, while the probability p that each unit will be dropped independently from the others is fixed (often, $p = 0.5$).

   Figure 3.4 shows how the application of dropout affects the connections between the units of two hidden layers of a network.

   Applying dropout to a neural network amounts to sampling a "thinned" network from it, that consists of all the units that survived dropout. For each presentation of each training batch, a new thinned network is sampled and trained. Therefore, training a model with dropout can be seen as training a collection of thinned networks.

   At test time, a single model without dropout is used. The weights of this network are scaled-down versions of the trained weights. If a unit is retained with probability p during training, the outgoing weights of that unit are multiplied by p at test time.

3. *L2 regularization* is a widely used regularization method. It implies the addition of a penalty term to the loss function for every weight w_i of the Network,

namely

$$\mathcal{L}' = \mathcal{L} + \frac{1}{2}\lambda \sum_{i=1}^{N} w_i^2 \tag{3.18}$$

where λ is called regularization strength and N is the number of weights in the Network.

Equation 3.18 shows that the weights with larger values will contribute in greater part to the total loss. All the weights are pushed to smaller values, but none of them will likely be constrained to be exactly null. However, the parameters with larger values will be penalized more, since a negative term is added to the optimizer.

For example, after the application of this regularization term, Equation 3.9 writes

$$w_{e+1,i} = w_{e,i} - \gamma \frac{\partial}{\partial w_i}(\mathcal{L}(\mathbf{w}_e) + \frac{1}{2}\lambda w_{e,i}^2) = w_{e,i} - \gamma \frac{\partial}{\partial w_i}(\mathcal{L}(\mathbf{w}_e)) - \gamma \lambda w_{e,i}. \tag{3.19}$$

Similar to this method is the *L1 regularization*: the difference is in the term added to the loss function

$$\mathcal{L}' = \mathcal{L} + \lambda \sum_{i=1}^{N} |w_i| \tag{3.20}$$

with the same notation as before. In this case, the derivative of the penalty term is a constant value, that is subtracted to the weight at each update and might constrain it to be precisely zero.

3.4 Convolutional Neural Networks

Convolutional Neural Networks (LeCun & Bengio, 1998) are a particular kind of Neural Networks specialized for processing data characterized by a grid-like topology (Goodfellow et al., 2016). This is the case, for instance, of images, that can be treated as grids of pixels: the following Section will be focused on this type of data.
The peculiarity of the CNNs is in the exploitation of the convolutional layers, that compute the convolution between their input and a *kernel*: the output of this operation is often referred to as *feature map*.
In this context, the kernels correspond to the units of a generic Network: the parameters that define them are learned in the training procedure. In particular, each convolutional layer is composed by a series of kernels, whose number is a hyperparameter to define during the implementation of the CNN. Deep architectures hierarchically learn high-level representations of the data: for example, in the analysis of an image, the first layers may identify lines and edges, while deeper layers can detect shapes.
One of the first CNN architectures to be presented has been the LeNet-5 in the work by Lecun et al. (1998), who applied it to digits recognition. The architecture of this Network is representative of the basic structure of modern CNNs and is shown in Figure 3.5.

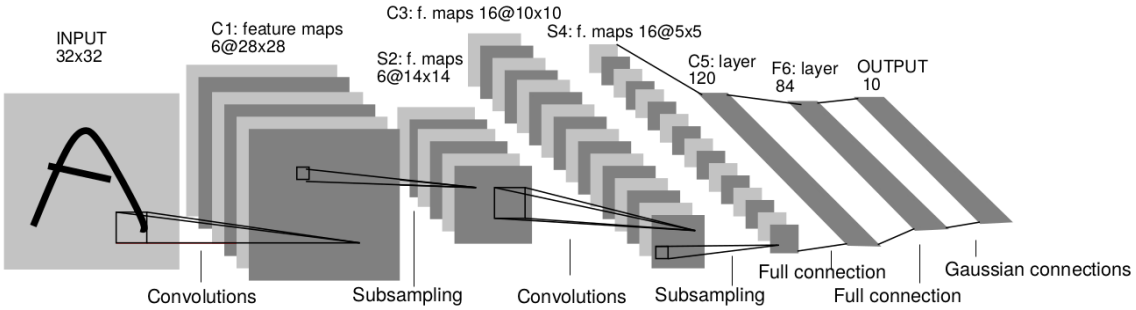


Figure 3.5: The architecture of LeNet. The input image is processed through convolutional and pooling layers. The feature maps are concatenated before being processed by two fully connected layers and then are passed to the output layer. Image from Lecun et al. (1998).

The input is an image, that is convoluted with a series of kernels in the first convolutional layer, so that a series of feature maps is produced. The feature maps are later subsampled by a pooling layer and new convolutions are performed. This procedure is repeated for a certain number of layers and, before being fed to the output layer, the feature maps are concatenated and processed by three fully-connected layers, that work as the layers of regular NNs.
Now we will describe the convolution and pooling operations in greater detail.
The convolution between two functions is a linear operation that is be defined by an integral. However, the convolution between multi-dimensional arrays, such as images and kernels, is better described as a discrete multiplication between matrices. Typically, the kernel is smaller than the image and the resulting output is a very sparse matrix with many elements that are null.
The output S of the convolution between an image I and a kernel K of dimension $(M \times N)$ can be computed as

$$S(i,j) = (K * I)(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} I(i-m, j-n) K(m,n). \tag{3.21}$$

where (m,n) represents a generic point on the kernel grid, while (i,j) represent a point on the image grid.
The employment of convolutions has several advantages (Goodfellow et al., 2016):

1. *Sparse interactions*: each unit in a classical neural network is connected to the one of the previous layer. In a CNN the kernel is smaller than the input of the layer, and the number of connections is also reduced. The most important consequence of this property is that the number of parameters that defines the Network is reduced, and so is the memory requirement. Figure 3.6a shows how one output unit, s_3, is connected to just three input units of the input layer, instead of all of them.

   The units of one layer that are directly connected to one unit of the following layer are also known as its *receptive field*.

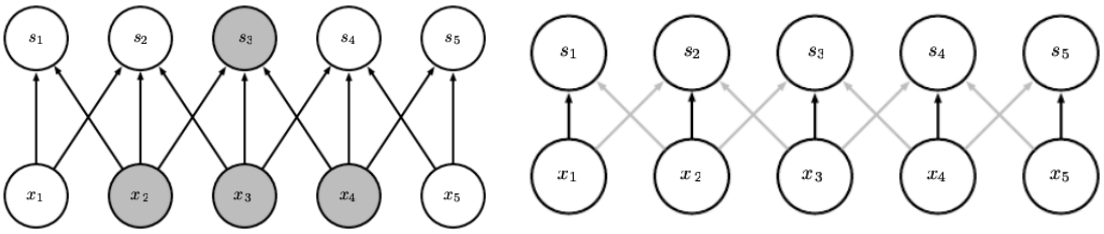


Figure 3.6: Representation of sparse connectivity (a) and parameter sharing (b) in two adjacent layers of a CNN. s_i denote the output units, while x_i are the input units. The arrows represent the connections between these units. Specifically, in image (b) the black arrows represent the same connection, that is repeated spatially over the input units. Images from Goodfellow et al. (2016).

2. *Parameter sharing*: each kernel in a convolutional layer of a CNN is applied to every position of the input, hence the algorithm does not learn a set of parameters for every location, but just one set. This property further decreases the memory requirements of the model. Figure 3.6b depicts how one parameter (black arrow) is used in several positions of the input.

3. *Equivariance to translation.* This property of CNNs is a consequence of parameter sharing: spatially repeating the same kernel over the input grid allows the network to become equivariant to little translations of the input.

   However, CNNs are not intrinsically equivariant to other forms of transformations, such as rotations: this problem can be addressed with other strategies.

In a convolutional layer several convolutions are performed in parallel, and then passed to an activation function, that adds non-linearity.
It is common to add a pooling layer in-between convolutional layers.
The pooling operation substitutes the value of the output at a given position with a summary statistic of the adjacent values. The most common pooling function is max pooling (Zhou & Chellappa, 1988), but other statistics, such as average or weighted average may also be used.
Including pooling layers in a CNN's architecture is important for two main reasons. Firstly, they make the model invariant[4] to little modifications of the input. This is clear from Figure 3.7: the input units are shifted of one pixel, but just two of the output units are affected by this modification after max pooling has been applied. Secondly, using pooling layers reduces the size of the feature maps and, by doing so, the number of parameters and computation in the Network. Generally, the pooling filters have dimension (2×2) or (3×3) and are applied to the feature maps with stride (2×2) to downsample them.

[4]It should be noted equivariance and invariance to translations are not equivalent properties. Being *equivariant* to translations means that if the input is translated, the output will be translated in a similar proportion, while being *invariant* to translations indicates that a translation in the input does not amount to any modification the output whatsoever.

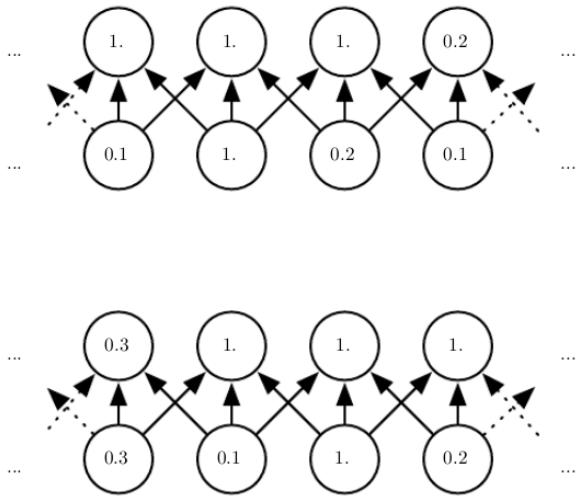


Figure 3.7: This image shows how max pooling makes the CNN invariant to little modifications of the input. In this case, the input has been shifted of one pixel to the right (second row in top and bottom panel, respectively), hence all of the units have a different value, but just half the values of the max pooling output are changed (first row in top and bottom panel, respectively). Image from Goodfellow et al. (2016).

The following Sections describe the architecture and properties of three popular CNNs (VGGNet (Simonyan & Zisserman, 2015), GoogLeNet (Szegedy et al., 2015), ResNext (Xie et al., 2017)) that have become a benchmark in the scientific community in the past few years and have inspired the models implemented in this thesis work.

3.4.1 Visual Geometry Group Network

The Visual Geometry Group Network (VGGNet) has been presented in the work by Simonyan & Zisserman (2015). This is one of the first works to exhaustively evaluate the relationship between the depth of a model and its performance. In fact, the authors built different Network configurations, with up to 19 layers and compared them.
However, the most relevant innovation proposed in this work is the use of small convolutional filters, with a receptive field of 3×3: this allowed the construction of such deep models in the first place. Previous works usually employed filters of bigger size, with a receptive field of 5×5 or 7×7.
With equal number of layers, the introduction of small filters keeps the number of trainable parameters in the CNN smaller than the one of networks that make use of larger filters. Therefore, introducing smaller filters in the architecture allows the implementation of deeper architectures in the first place, since it optimizes the memory requirements.
Moreover, the concatenation of multiple kernels of size 3×3 has the same resulting receptive field of larger filters. For instance, a stack of two convolutional layers

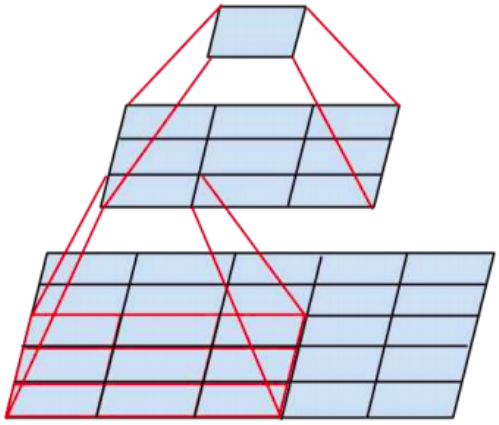
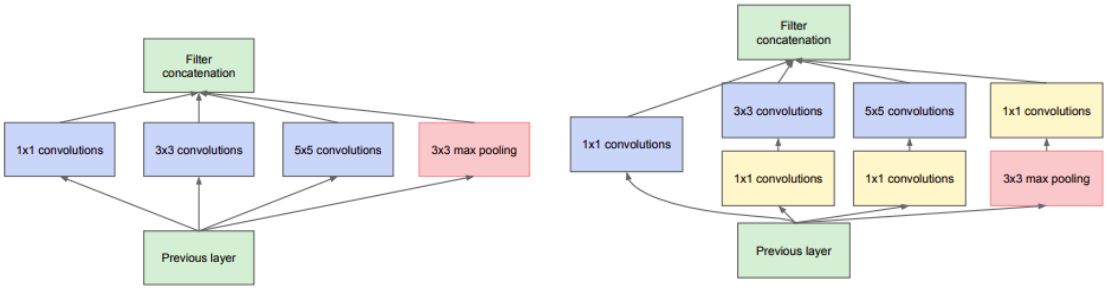


Figure 3.8: Stacking to filters of size 3×3 has the same effective receptive field as using a 5×5 filter, while keeping the number of parameters required by the model lower. This property allows the implementation of deeper architectures. Image from (Szegedy et al., 2016).

3×3 has the effective receptive field of a 5×5 layer, as it is shown in Figure 3.8, but requires fewer parameters. In fact, in the first case $2 \cdot 3^2 = 18$ parameters would be necessary to describe the operation, while in the second case the number of parameters would increase to $5^2 = 25$. In this example, using 5×5 filters would hence be $25/9 = 2.78$ times more expensive than using the smaller ones.
It is notable that filters with size 1×1 can also be used in CNNs, but their main advantage is to reduce the dimension of the input without actually changing its features. However, they are not particularly useful in the detection of the characteristics of the feature maps, as 3×3 filters are the smallest able to capture the notions of left and right, up, down and center (Simonyan & Zisserman, 2015).

3.4.2 Inception Networks

The reasons for the Inception Networks architectures were first outlined in the work by Szegedy et al. (2015), who apply the idea of Network in Network of Lin et al. (2013) to CNNs.
Trying to improve the performance of a CNN by enlarging its architecture, namely the depth and width, comes with a cost. In fact, the number of parameters in the model increases with the model's size, favouring overfitting and increasing the requirements of computational resources.
Szegedy et al. (2015) propose to tackle this problem in an original way. Instead of developing the network in size, they suggest looking for the filter construction that locally operates optimally and repeat it spatially, taking advantage of the local invariance of CNNs. In other words, their idea is to apply filters with different size on the same input, making the model learn features on different scales in the same feature maps.
Specifically, this idea is implemented through the *inception module*, in Figure 3.9. In the simplest configuration, displayed in Figure 3.9a each module applies filters



(a) Inception module, naive version

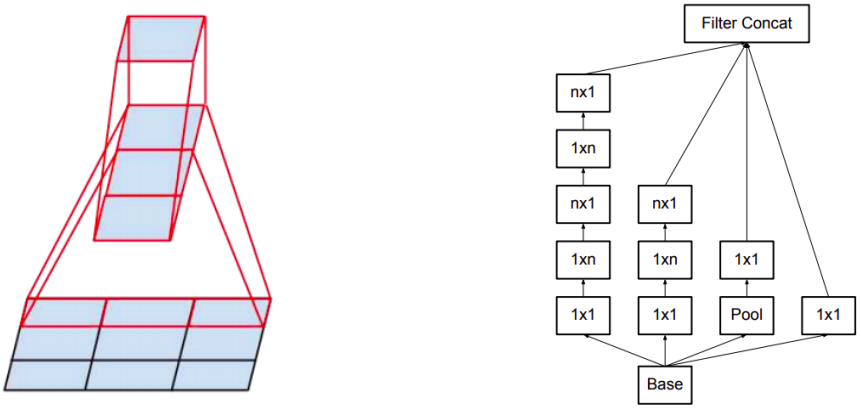
(b) Inception module with dimension reductions

Figure 3.9: The inception module in its naive (a) and in its final (b) implementation. The input of the module is processed by filters of different size in parallel, and the outputs of these convolutions are concatenated and used as input of the following layer. Image from Szegedy et al. (2015).

of several size (1×1, 3×3, 5×5) and a pooling function to the same input and concatenates their output, passing the result of this operation as input to the following layer.
However this implementation can be improved by applying 1×1 filters before 3×3 and 5×5 filters in order to reduce the dimension of the input and, by doing so, decreasing the computational cost of the operations. This version of the inception module, in Figure 3.9b is the one actually used in the network's implementation.
An Inception Network is series of such modules stacked upon each other, with max pooling layers occasionally used to reduce the resolution of the feature maps.
GoogLeNet, the particular Inception Network described in the original work by Szegedy et al. (2015) is 22 layers deep. Other than the inception module, this Network also presents another innovation. Given the good effectiveness of shallower networks, the authors have added auxiliary classifiers to intermediate layers, that act as regularizers and help the propagation of the gradient in the lower layers. In the training phase, their loss is weighted by 0.3 and summed to the loss of the final output layer.
A further improvement of the original inception module design is presented in the work by Szegedy et al. (2016). Specifically, the 5×5 filters are replaced by two 3×3 filters stacked together in order to decrease the amount of parameters required by the model. Moreover, the possibility to substitute a $n \times n$ filter (in this case, 3×3) with an asymmetric convolution between a filter of size $n \times 1$ followed by a filter of size $1 \times n$ is also introduced. This expedient, graphically and schematically shown in Figure 3.10a and 3.10b respectively, further reduces the memory requirements of the model.

3.4.3 Residual Networks

He et al. (2016) introduce *residual learning* as a way to make the training process of Deep Networks easier and more efficient. In fact, the authors notice that the



(a) (b)

Figure 3.10: Image (a) illustrates how stacking two asymmetric convolutions of size 3×1 and 1×3 gives the same result as a 3×3 convolution. Image (b) shows a scheme of the architecture of the inception module after the introduction of the asymmetric convolutions. Image from Szegedy et al. (2016).

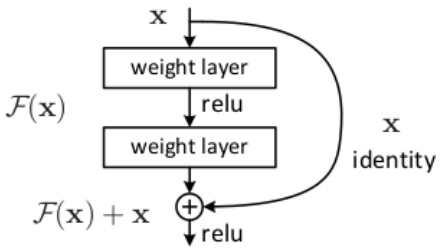
addition of additional layers does not always correspond to a better performance of the network and experiment a new learning framework.
The basic idea of this type of architecture, called Residual Networks (ResNet), is that it is easier for a certain layer (or a few stacked layers) to learn a residual function with respect to the input, other than the complete and more complicated mapping. In practice, this is implemented using a residual block that exploits shortcut connections, as shown in Figure 3.11a. The input of the block $\mathbf{x}$ is simultaneously propagated through the layers within the block and stored without being changed. The function that the block is expected to learn can be conceived as

$$\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x} \tag{3.22}$$

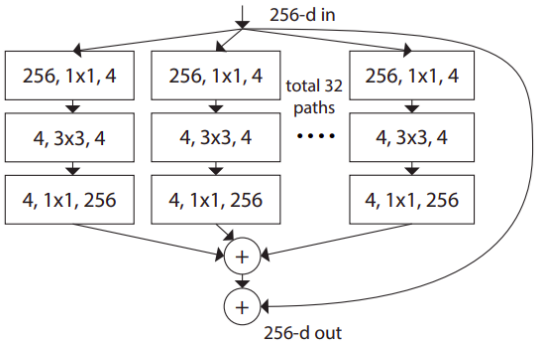
where $\mathcal{H}(\mathbf{x})$ is the original function and $\mathcal{F}(\mathbf{x})$ is the residual function. Thus, the original function can be computed as $\mathcal{F}(\mathbf{x}) + \mathbf{x}$.
For instance, if a convolutional layer was supposed to learn the identity mapping, the corresponding residual block would have to learn all zeros, that is an easier task to achieve. Of course, the same argument is true for other, more complicated and realistic mappings.
The output vector $\mathbf{y}$ of the residual block is hence calculated as

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x} \tag{3.23}$$

As He et al. (2016) prove, a model that uses this building block and has the same number of layers of a classical CNN, will perform better.
The evolution of the ResNet is the ResNeXt, presented in the work by Xie et al. (2017). This network architecture is based on the ResNeXt block, that aggregates a set of transformations, as the Inception Networks do.



(a) ResNet block



(b) ResNeXt block

Figure 3.11: This figure compares the architectures of the simple residual block and of the residual block after the addition of the cardinality parameter. Image (a) from He et al. (2016) and image (b) from Xie et al. (2017).

The aggregated transformations can be presented as

$$\mathcal{F}(\mathbf{x}) = \sum_{i=1}^{C} \mathcal{T}_i(\mathbf{x}) \tag{3.24}$$

where $\mathcal{T}_i(\mathbf{x})$ can be an arbitrary function and C is a hyper-parameter called *cardinality*, that represents the size of the set of transformations to be aggregated.
In particular, the aggregated transformation in Equation 3.24 can serve as the residual function in Equation 3.23. In this case, the output $\mathbf{y}$ of the ResNeXt block is given by

$$\mathbf{y} = \mathbf{x} + \sum_{i=1}^{C} \mathcal{T}_i(\mathbf{x}). \tag{3.25}$$

The basic structure of the Resnext block is displayed in Figure 3.11b.
The approach of the ResNeXt architecture is also referred to as Network in Neuron, because it replaces the linear transformation $w_i x_i$ operated by the simple neuron with a non-linear function.
This block allows to expand the network in a new dimension, other than the depth and the width. A thorough comparison of the effect that increasing the depth, width or cardinality has on the model's performance can be found in the original work by Xie et al. (2017).

Chapter 4

The Datasets

The identification of the optimal network architecture for a certain task completion is just one of the fundamental steps in the implementation of a supervised ML model. It is, however, equally important to carefully choose the dataset used in the training phase, since the features extracted from it will determine how the weights and biases of the model are adjusted and defined. In fact, the final configuration of these parameters will be adopted when applying the model to other, unknown datasets, whose properties need to be well represented by the training set employed.
The models implemented in this thesis work have been trained and tested on two different datasets, that mainly contain images of GGSL events produced by isolated galaxies.
The two datasets will be described in greater detail in Sections 4.1 and 4.2.

4.1 Lens Finding Challenge Dataset

The Strong Gravitational Lens Finding Challenges[1] are a series of open competitions organized by the Bologna Lens Factory[2]. They aim to stimulate the research for reliable methods for the detection of strong gravitational lensing events in future surveys.
The results of the Challenge 1.0 were presented in the work by Metcalf et al. (2019) and compared the effectiveness of different techniques in the classification of the images: visual inspection, arc-finder algorithms and ML methods, such as Support Vector Machines (see e.g. Burges 1998) and CNNs. The latter have proven to perform particularly well and have the great advantages of automatizing the feature extraction process and allowing a fast analysis of great amounts of images.
The simulations implemented for this Challenge are described in detail in Metcalf et al. (2019), therefore we will only focus on some of the most important steps.
First of all, the simulations started with the creation of a catalog of Dark Matter halos, obtained by constructing a light-cone within the Millennium Observatory project (Overzier et al., 2013), starting from the Millennium simulation (Boylan-Kolchin et al., 2009).

[1] http://metcalf1.difa.unibo.it/blf-portal/gg_challenge.html
[2] http://metcalf1.difa.unibo.it/blf-portal/index.html

Table 4.1: Main characteristics of the VIS and NISP instruments.

Instrument	Capability	λ range (nm)	Pixel size (arcsec)
VIS	Visual imaging	550 - 900	0.1
NISP	Near-Infrared imaging photometry	Y (920 - 1146), J (1146 - 1372), H (1372 - 2000)	0.3

Afterwards, the lensing code Gravitational Lensing with Adaptive Mesh Refinement[3] (GLAMER) (Metcalf & Petkova, 2014; Petkova et al., 2014) was applied to the halos in this catalog to map all the caustics located in different source planes on several lens planes.
The background sources lensed in the simulation were taken from the Hubble Ultra Deep Field (Beckwith et al., 2006). In particular, they were placed randomly within three times the distance between the point of a given caustic that was furthest from the caustic's center and the caustic's center: this choice allows the simulation of diverse lens configurations in addition to clear events.
Moreover, the visible lens galaxies were simulated within these DM halos by defining an analytic model for their surface brightness. Specifically, the parameters used to simulate them were provided by the Millennium Observatory project, using the semi-analytic galaxy formation models of Guo et al. (2011). Within GLAMER, the surface brightness map was converted into a mass map, then added to the halos previously described to obtain the total mass distribution of the lens. Finally the deflections caused by this map were calculated by Fast Fourier Transform and added to the halos' contributions for the ray tracing.
Instead, the dataset used in this thesis work consists of the mock images created for the Challenge 2.0. They were generated following a similar procedure as those for the Challenge 1.0, but the initial catalog of foreground and background galaxies was taken from the Euclid Flagship simulation[4] (see `http://metcalf1.difa.unibo.it/blf-portal/galaxy-galaxy.html`).
The training set is composed of 100000 images mimicking the quality of the observations expected by the Visual Imager[5] (VIS) and the Near-Infrared Spectrometer and Photometer[6] (NISP) instruments, that the Euclid space telescope will be equipped with. The main properties of these instruments are in Table 4.1.
The images are available in four different bands: the VIS band and the NISP (H, Y, J) bands. The dimensions of the VIS and NISP images are 200×200 and 66×66 pixels, respectively. Given the resolution of the images, also reported in Table 4.1, these correspond to $20'' \times 20''$.
Along with the images the training set also consists of a catalog of properties that describe the lens candidates:

- `ID`: image identification number;

[3] `http://glenco.github.io/glamer/`
[4] `https://sci.esa.int/web/euclid/-/59348-euclid-flagship-mock-galaxy-catalogue`
[5] `https://sci.esa.int/web/euclid/-/euclid-vis-instrument`
[6] `https://sci.esa.int/web/euclid/-/euclid-nisp-instrument`

- `x_crit`, `y_crit`: coordinates of the center position of the critical curve (in rad);
- `source_ID`: source identification number;
- `z_source`, `z_lens`: redshift of source and of lens, respectively;
- `mag_source`, `mag_lens`: magnitude of source and of lens, respectively;
- `ein_area`: Einstein area of the largest critical curve (in rad^2);
- `n_crit`: number of critical curves;
- `r_source`: distance of the source from the center of the caustic (in rad);
- `crit_area`: area of the caustic (in rad^2);
- `n_pix_source`, `n_pix_lens`: number of pixels where the source and the lens, respectively, are observable (i.e. above 1σ);
- `source_flux`, `lens_flux`: flux of the source and of the lens in those pixels in units of σ;
- `n_source_im`: number of multiple images of the source;
- `mag_eff`: effective magnification of the source, taking into account the magnification of all images;
- `sb_contrast`: average surface brightness contrast between the lens and the source in pixels above threshold;
- `color_diff`: difference of colour between the lens and the source;
- `n_gal_3`, `n_gal_5`, `n_gal_10`: number of companions with $m < 25$ within 3, 5, 10 $arcsec$ from the source;
- `stellar_mass`, `halo_mass`: stellar and halo mass of the main lens;
- `n_sources`: number of sources added in the simulation of the image.

The images in the training set are not labeled as lenses or non lenses a priori. Following the criteria of the Challenge 2.0, the objects are considered lenses if they satisfy the following conditions:

$$\begin{cases} \texttt{n\_source\_im} > 0; \\ \texttt{mag\_eff} > 1.6; \\ \texttt{n\_pix\_source} > 20. \end{cases} \tag{4.1}$$

The ensemble of unambiguous non lenses is constituted by the the 10004 galaxies that do not have a corresponding background source and are associated to `n_sources` $= 0$ in the training set. In many other cases the source is intrinsically very faint and/or is weakly magnified and the object is not recognisable as a lens. For this reason, the parameters `n_pix_source` and `mag_eff` are also considered in the classification criteria 4.1. Of course, depending on the sensitivity of the model, the classification of the borderline objects might vary, while the most clear ones should be unequivocally classified as belonging to the correct category.
In this work the dataset was initially cleaned by dropping the images with a source at $z > 7$, thus leaving a catalog of 99612 objects. Afterwards, the criteria listed in

Equations 4.1 were adopted to divide the dataset into the subsets of lenses and non lenses, that hence consist of 49060 and 50552 objects, respectively.
A randomly selected sample of images belonging to each of these two categories is displayed in Figure 4.1 and 4.2. Moreover, Figure 4.3 shows the distribution of some of the parameters previously enumerated for the whole dataset and for the two subsets separately.

4.2 The Euclid VIS dataset

The second dataset[7] we use to train the Networks is composed by 20000 images, divided into two different subsets, one of 10000 lenses and one of 10000 non lenses, respectively.
The images were simulated using GLAMER, following a similar procedure as the one described in Metcalf et al. (2019) for the creation of the dataset of the Challenge 1.0. Moreover, they mimic the Euclid telescope's observational capabilities, as the dataset of the Challenge 2.0. However, in this case the images are only available in the VIS band. The size of the images is 200×200 pixels.
Each subset of lenses and non lenses is also complemented by two catalogs, with a total of four catalogs.
The first one is `euclid_image_catalog.csv`: it contains the same information described in the previous Section for the Lens Finding Challenge dataset. In addition, it contains two entries referred to the source's Spectral Energy Distribution (SED):

- `source_sed_bzp`: index of the SED used for the source, among the templates available in BPZ (see Section B.2 for more details about this software);
- `source_sed_cosmos`: estimated index of the source's SED among the templates of COSMOS[8].

All the objects that belong to the class of the non lenses are characterized by `n_sources` = 0. Consequently, several parameters, such as `n_pix_source`, `z_source`, `flux_source` are equal to 0 or NaN[9] for all of the members of this subset.
Figure 4.4 displays the distributions of some of the parameters available to describe the galaxies in the dataset.
The second catalog, `euclid_lens_gal_catalog.csv`, contains the values of following parameters referred to the main lens:

- `lens_number`: index of the lens;
- `mag_lens`: magnitude of the lens;
- `Rh`: scale height of the disk;
- `Reff`: effective radius of the lens;
- `Btot`: ratio of the bulge luminosity to the total luminosity;

[7] Available at http://metcalf1.difa.unibo.it/TrainingData/
[8] https://cosmos.astro.caltech.edu/
[9] Not a Number.

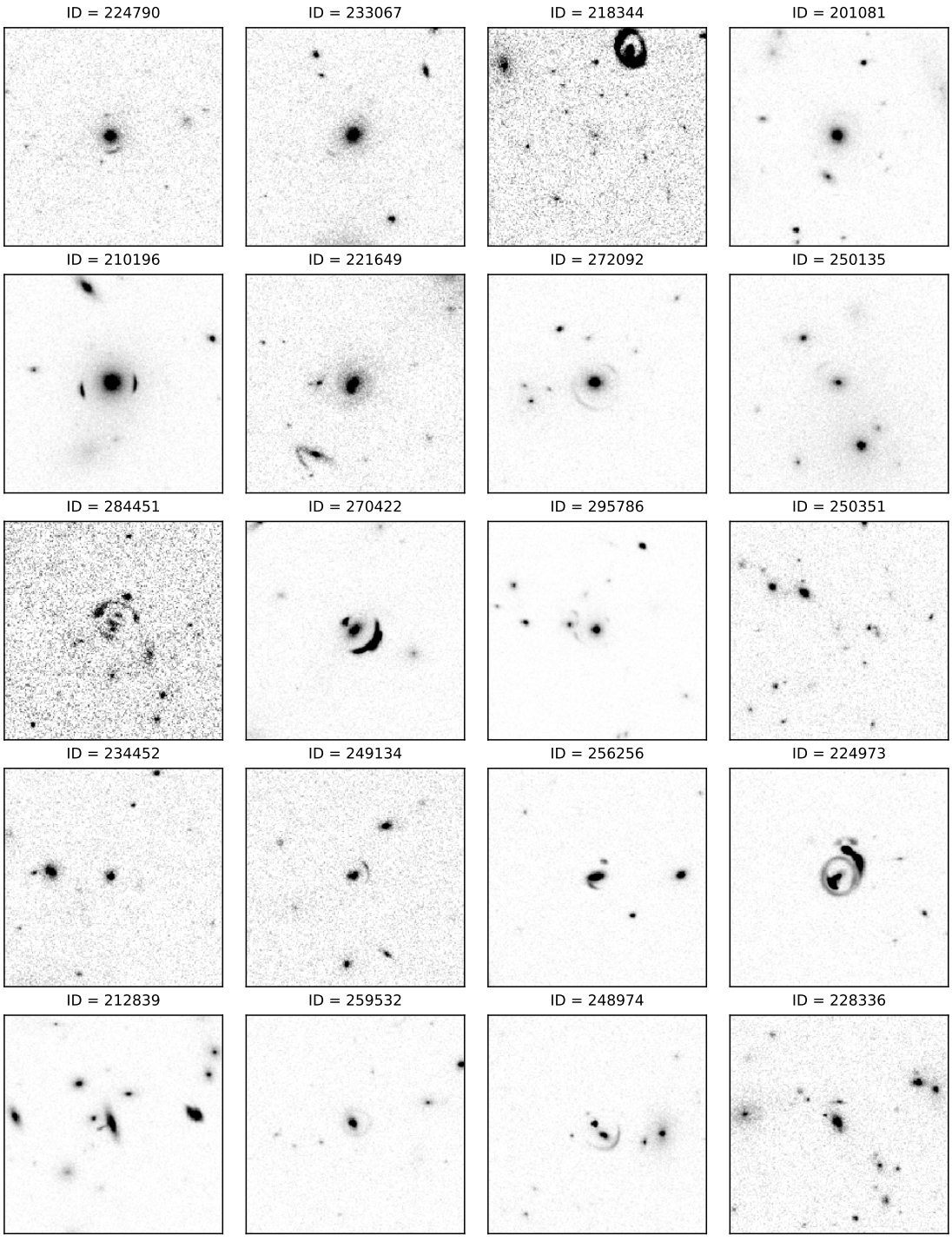


Figure 4.1: Randomly selected sample of 20 images classified as lenses, in the VIS band.

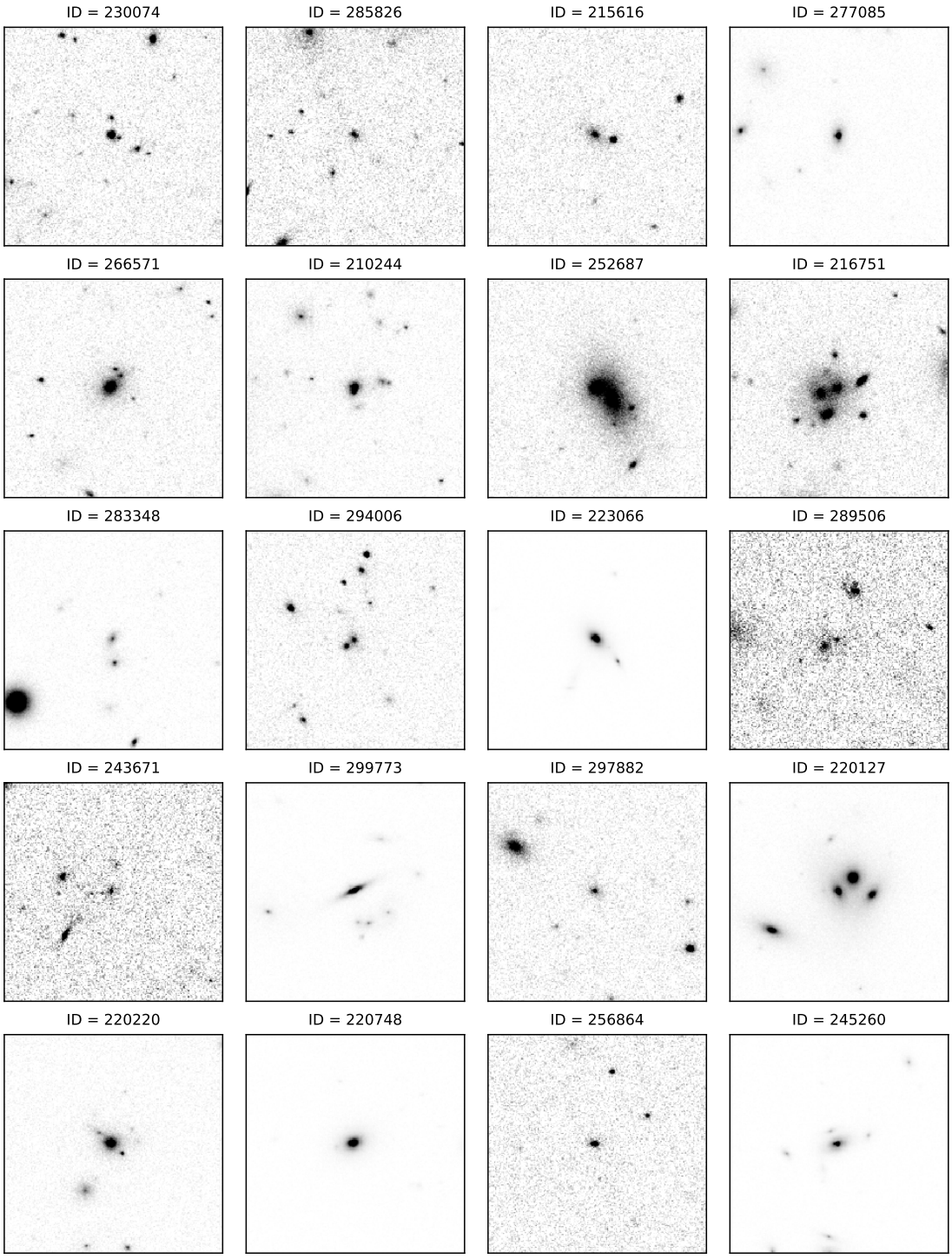


Figure 4.2: Randomly selected sample of 20 images classified as non lenses, in the VIS band.

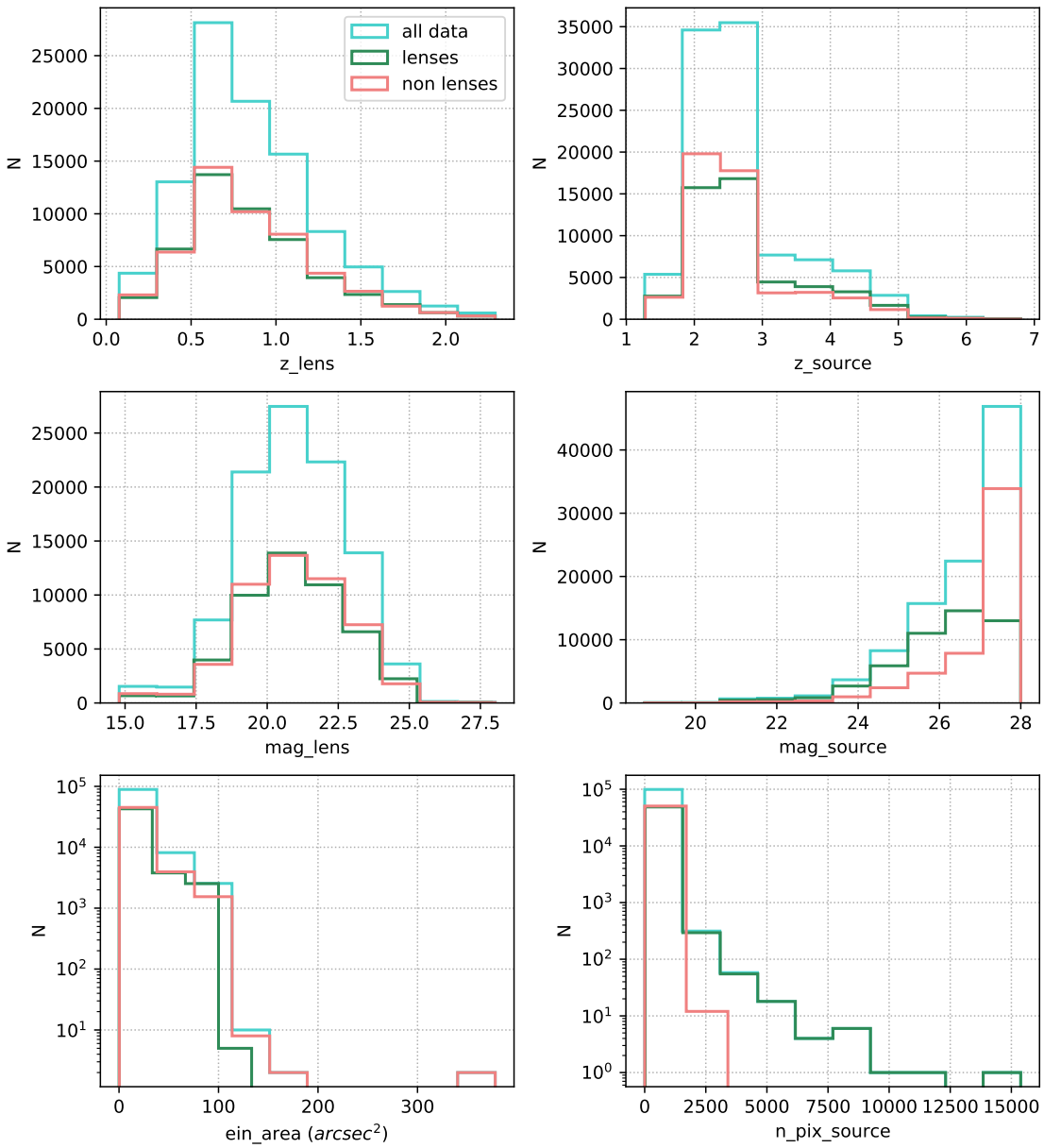


Figure 4.3: Distribution of several properties of the Lens Finding Challenge dataset: the redshift of the lenses (top left) and sources (top right), of the magnitude of lenses (central left) and sources (central right), Einstein area of the critical curve (bottom left) and number of pixels in which the source is visible (bottom right). The distributions are shown for the whole dataset (sky blue), for the lenses (green) and for the non lenses (pink).

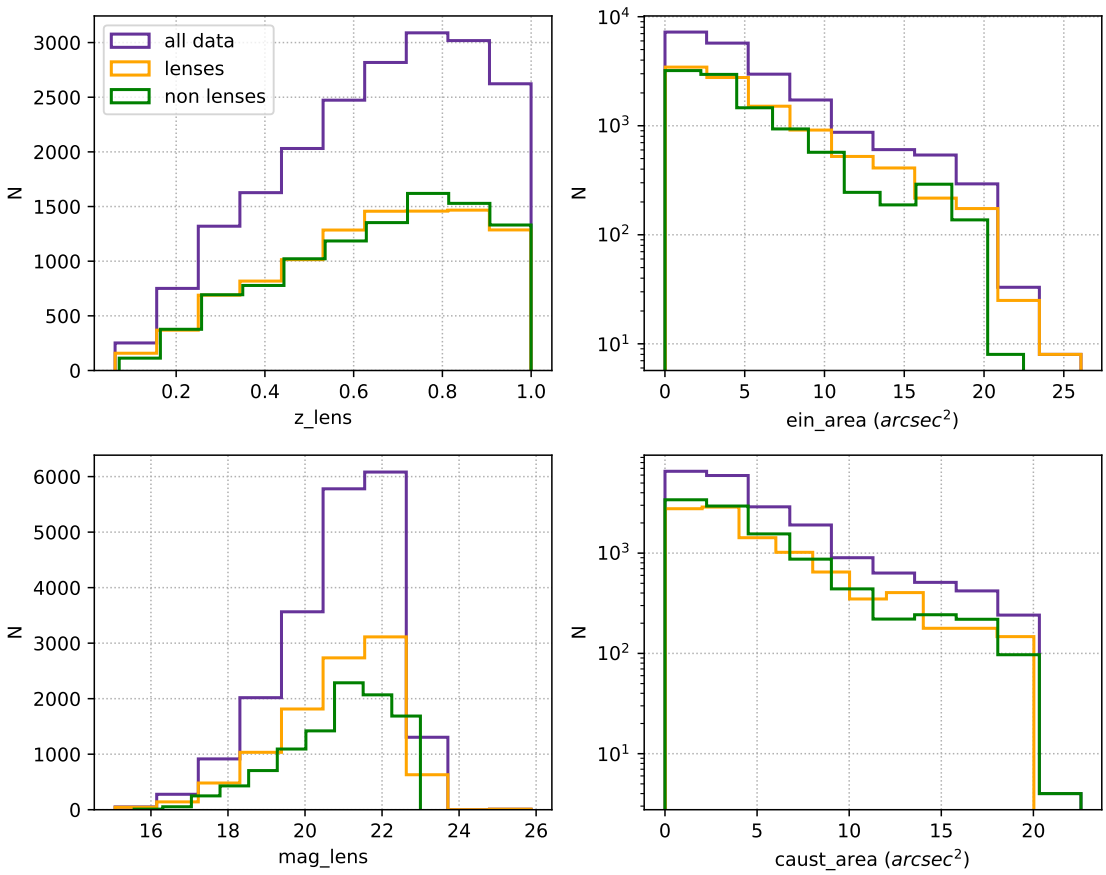


Figure 4.4: Distribution of the redshift of the lens (top left), the Einstein area (top right), the magnitude of the lens (bottom left) and the area of the main caustic (bottom right) of the objects of the dataset. The distributions are shown for the whole dataset (purple), and for the subsets of lenses (orange) and non lenses (green).

- `MagBulge`: magnitude of the bulge ;
- `SphIndex`: Sérsic index (Sérsic, 1963) used to model the bulge;
- `Z`: redshift of the lens.

Before using the data, we clean it from the sources with z >7, obtaining a catalog of 9931 lenses and 10000 non lenses. Figures 4.5 and 4.6 show 20 objects randomly selected from the subsets of lenses and non lenses of this dataset.

4.3 Differences between the datasets

We can make a comparison between the images in the two datasets by analyzing the lenses in Figures 4.1 and 4.5 and the non lenses in Figures 4.2 and 4.6. It is evident from these examples that the classification of the images is more straightforward in the Euclid VIS dataset than in the Lens Finding Challenge dataset.

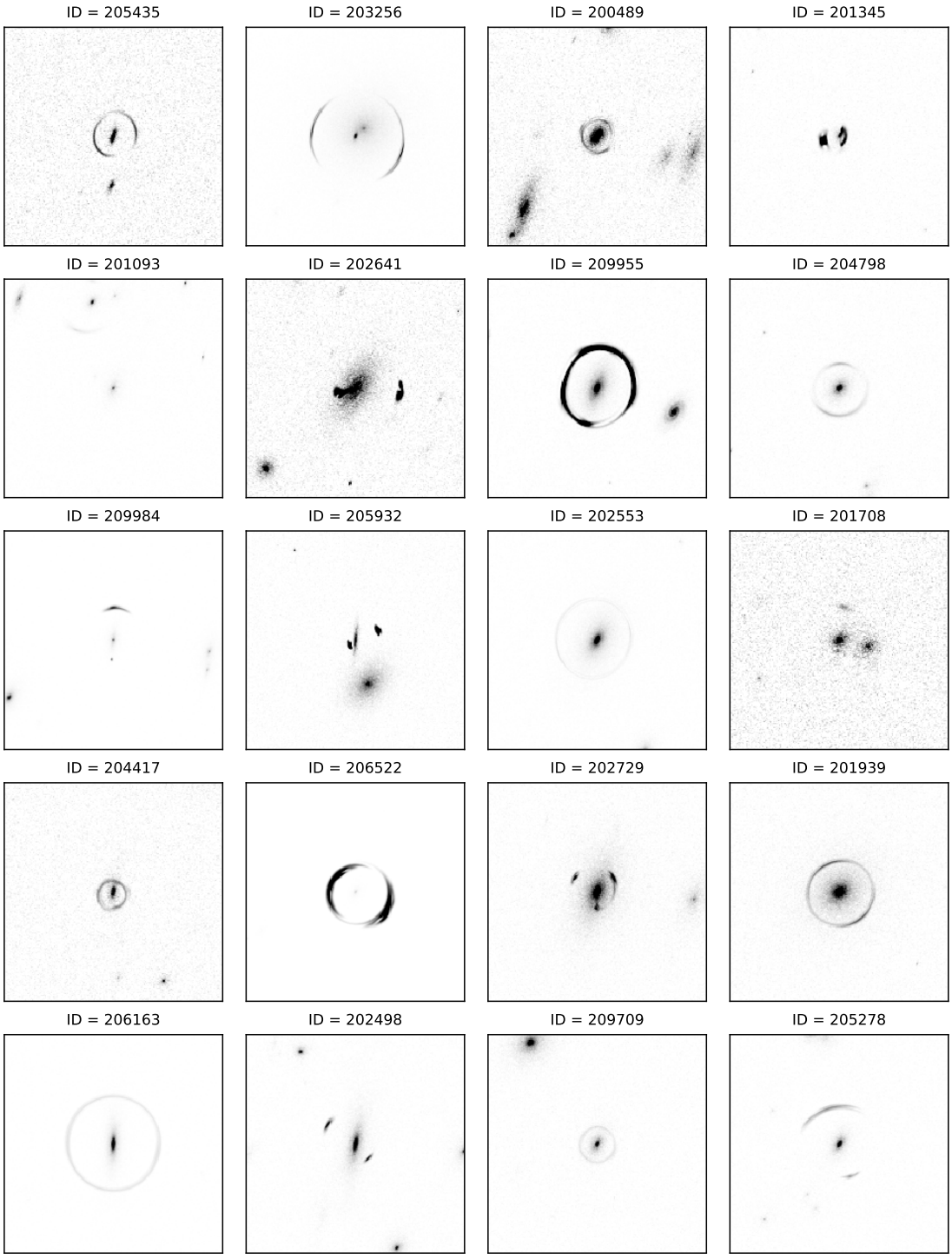


Figure 4.5: Randomly selected sample of 20 lenses from the Euclid VIS dataset.

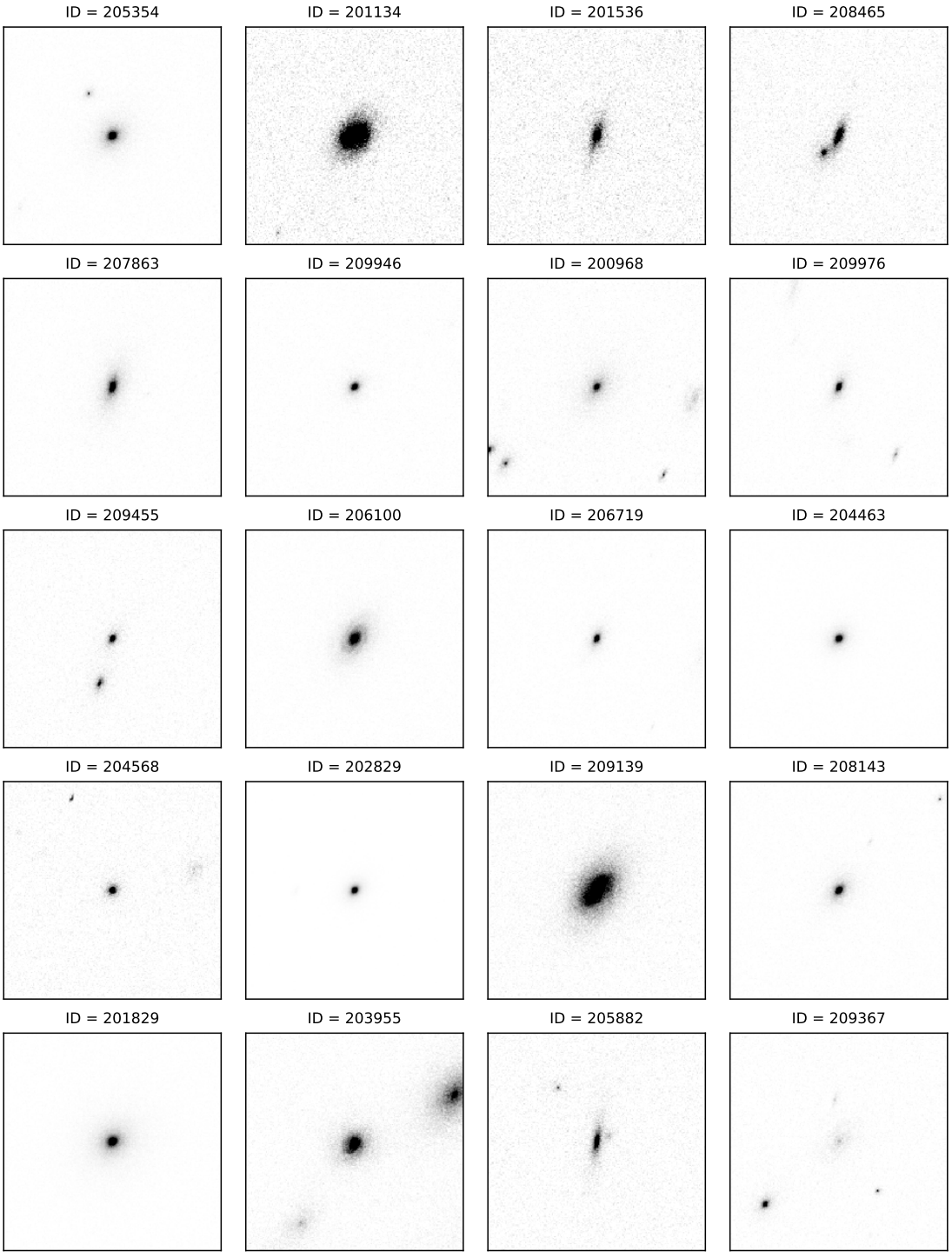


Figure 4.6: Randomly selected sample of 20 non lenses from the Euclid VIS dataset.

In particular, the lenses in the Euclid VIS dataset display clear arcs and rings in the majority of the cases, and fainter features in rarer instances. On the other hand, the lenses in the Lens Finding Challenge dataset are characterized by a more diverse morphology, that includes several images in which the detection of the lensing features is more challenging. The same consideration may be highlighted for the non lenses images: since the ones in the Euclid VIS dataset never show background sources in addition to the lens galaxy, their classification is easier.
We may also compare the distributions of some of the most relevant properties of the lens and source galaxies simulated in two datasets. In particular, Figure 4.7 shows the distribution of `ein_area`, `halo_mass`, `n_pix_source`, `mag_eff`, `z_source` and `mag_source` in the two datasets.
In particular:

1. The Einstein area of the lenses in the Lens Finding Challenge dataset is on average greater than that of the lenses in the Euclid VIS dataset.

   Considering that the Einstein area and radius are related through

   $$\theta_E = \sqrt{\frac{A_E}{\pi}}, \tag{4.2}$$

   we find $\theta_{E,max} \sim 3''$ in the Euclid VIS dataset and $\theta_{E,max} \sim 10''$ in the Lens Finding Challenge dataset.

2. The range of masses covered by the halos of the lenses in the two datasets is different. While the lenses in the Euclid VIS dataset are characterized by $M_{lens} \sim 10^{12} - 10^{14} M_{\odot}$, with the largest part of the lenses being in the interval $M_{lens} \sim 10^{12} - 10^{13} M_{\odot}$, the lenses in the Lens Finding Challenge dataset are characterized $M_{lens} \sim 10^{13} - few\, 10^{14} M_{\odot}$, with a peak at $M_{lens} \sim few\, 10^{13} M_{\odot}$.

   Therefore, the lenses in the Euclid VIS dataset are mainly galaxies of different size, while the Lens Finding Challenge also includes several examples of groups of galaxies.

3. The amount of pixels where the source is visible can be considered as an estimator of how evident the lensing features are in the images. For this reason, even though it is not an intrinsic property of the sources nor of the lenses, it is useful to compare its distribution in the two datasets.

   We expect the sources to be more distorted and clearer in the images characterized by a high value of `n_pix_source` and to be less observable in the opposite case. Although the distribution of this parameter covers the same range of values for the simulated lensing events in the two datasets, the source is less visible, on average, in the images of the Euclid VIS dataset.

4. The magnification of the sources is also different in the two datasets. Although some of the sources in the Euclid VIS dataset are demagnified by the lenses, their number is negligible with respect to that of the sources that are highly magnified, with $\mu_{max} \sim 10^4$. On the other hand, none of the sources in the Lens

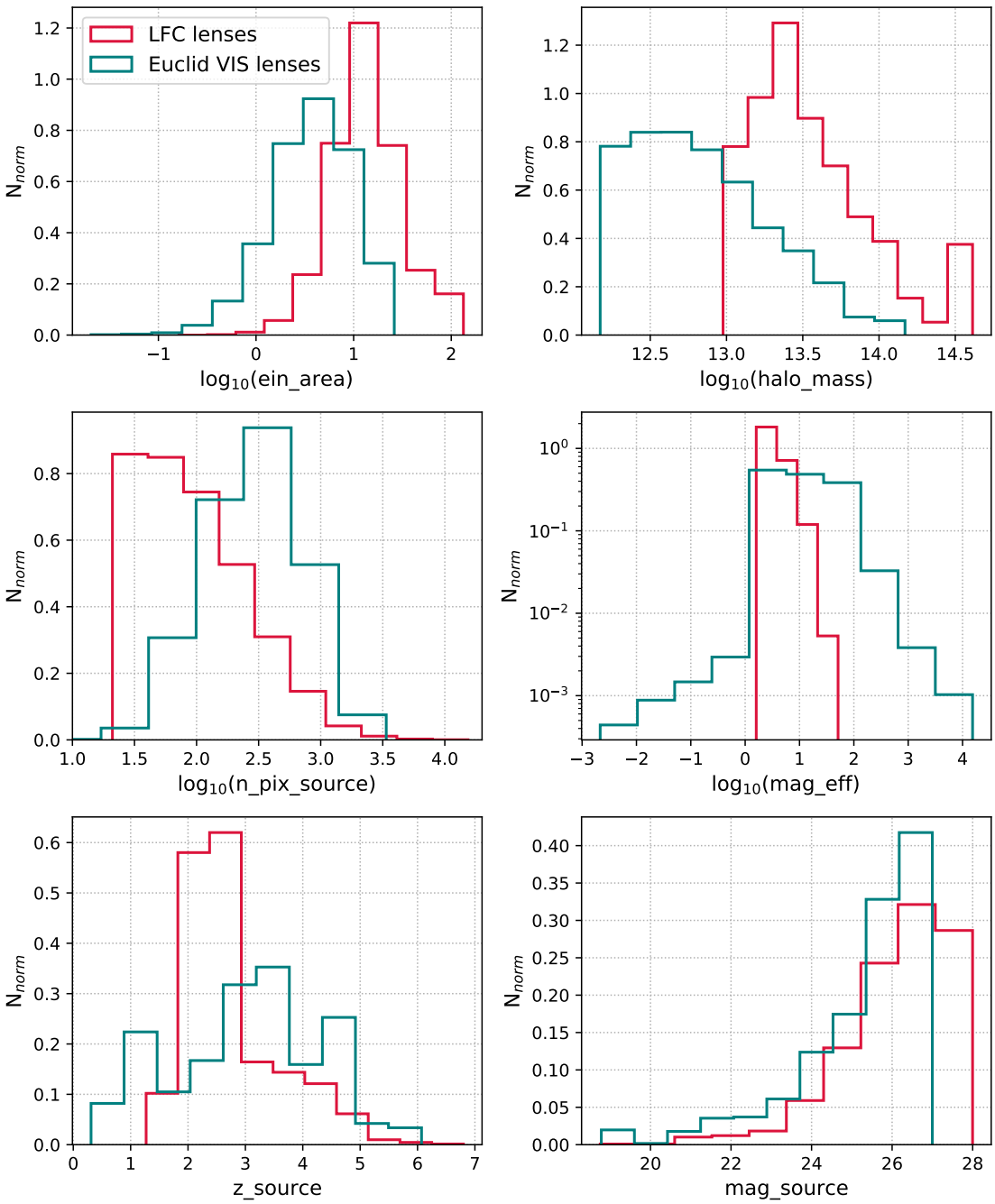


Figure 4.7: Comparison between the properties of the lenses in the Lens Finding Challenge (red) and in the Euclid VIS (blue) datasets. From top left to bottom right, the distributions of the Einstein area, halo mass, amount of pixels with a visible source, effective magnification, redshift of and magnitude of the background sources are displayed.

Finding Challenge dataset are demagnified, but the maximum magnification reached is smaller, since $\mu_{max} \sim 10^2$.

5. The range of redshift covered by the sources is similar in the two datasets. However, the sources in the Lens Finding Challenge dataset are mostly located at $z \sim 2 - 3$, while those in the Euclid VIS dataset are more homogeneously distributed in the range $z \sim 2 - 5$.

6. The distribution of the magnitude of the sources in the two datasets is quite similar. The main difference we might notice is that in the Euclid VIS dataset $m_{VIS,max} \sim 27$, while in the Lens Finding Challenge dataset $m_{VIS,max} \sim 28$, i.e. some of the sources in the second dataset are fainter than any of the objects in the first one.

The ensemble of the characteristics of sources and lenses described contributes to explain the observed differences in the images of the two datasets.

Chapter 5

Experiments and Results

We have trained and tested our models on the datasets described in the previous Chapter. We have paid particular attention to understand the impact of borderline objects on the efficiency of the classifiers.
This Chapter is divided into two parts. Section 5.1 describes the preliminary steps applied to the datasets before the training of the networks, while Section 5.2 introduces the metrics employed for the assessment of the performance of the models. Sections 5.3 and 5.4 describe the experiments and the main results obtained using the Lens Finding Challenge dataset, while 5.5 describes the results of the experiments conducted on the Euclid VIS dataset.
The architectures of our networks and the technical specifications related to the trainings are discussed in Appendix A.

5.1 Data Preprocessing

We divide each dataset described in the previous Chapter into three subsets: the training set, the validation set and the test set.
The training set represents the greatest part of the whole dataset, since it includes 70% percent of the data, while the validation set contains 5% of the data and the remaining 25% constitutes the test set.
In particular, the objects of the original dataset are randomly assigned to one of these subsets. Before passing them to the networks, however, we verify that the validation and test sets are well described by the training set. We do this by inspecting the distributions of several parameters that define the characteristics of the lenses and sources in the dataset, among those listed in the previous Chapter. More specifically, we consider the redshift, magnitude and Einstein area of the lenses, the redshift and magnitude of the sources and the amount of pixels where the source is visible.
Figure 5.1 displays the distributions of these properties in the three subsets selected in the case of the complete Lens Finding Challenge dataset. As it shown here, the distribution of these parameters is the same for the three samples, thus confirming that the training set is representative of the other two sets.
Once the dataset is split, we randomly select the 20% of the images in the training set and augment them. In particular, the operations we perform on the original image are: a rotation by 90°, 180° and 270° and an up-down and left-right flipping.

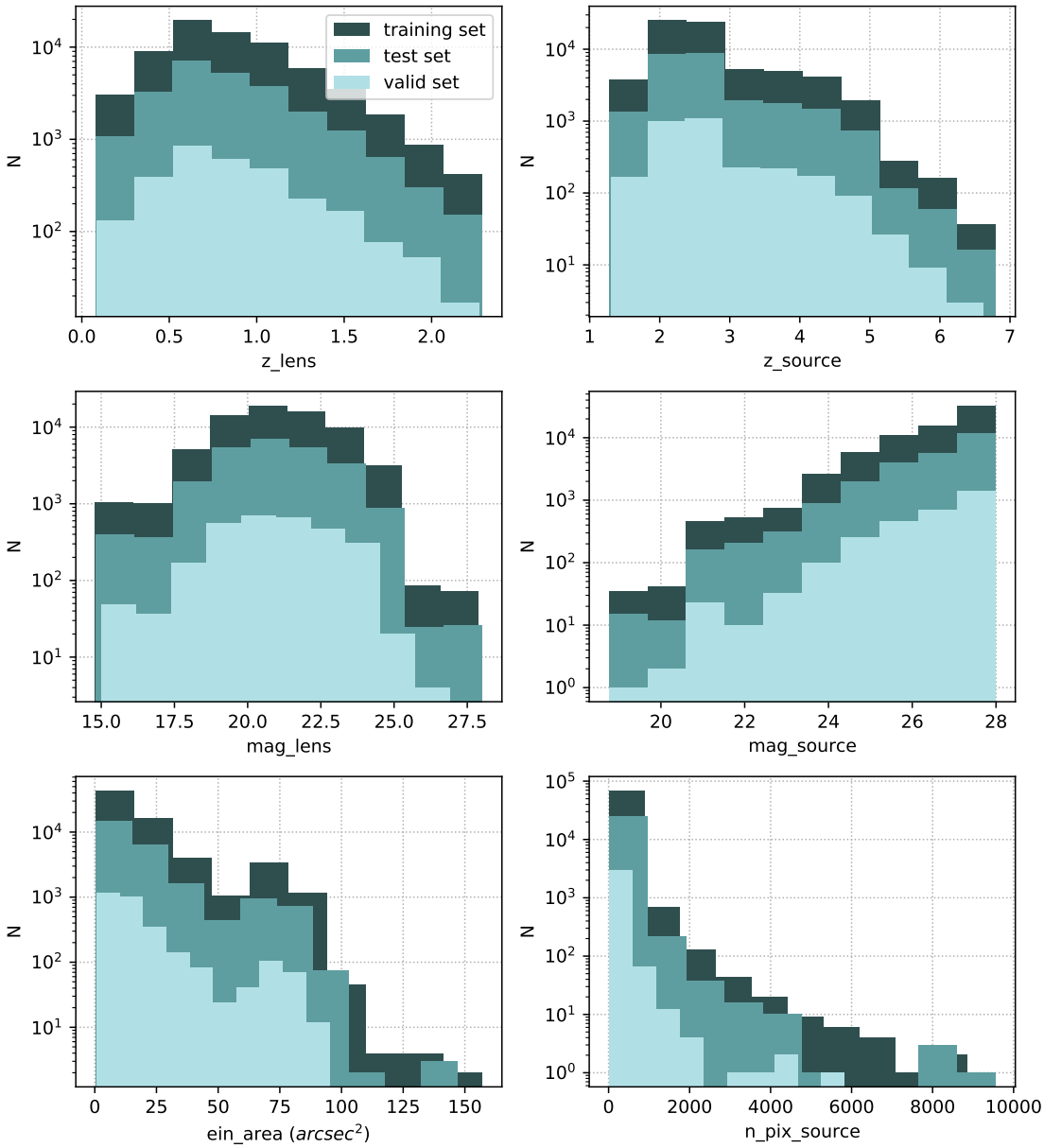


Figure 5.1: Distribution of the redshift of the lenses(top left) and sources (top right), of the magnitude of the lenses (central left) and sources (central right), of the Einstein area of the lenses (bottom left) and the amount of pixels where the source is observable (bottom right) in the training (dark blue), test (blue) and validation (sky blue) sets of the Lens Finding Challenge dataset.

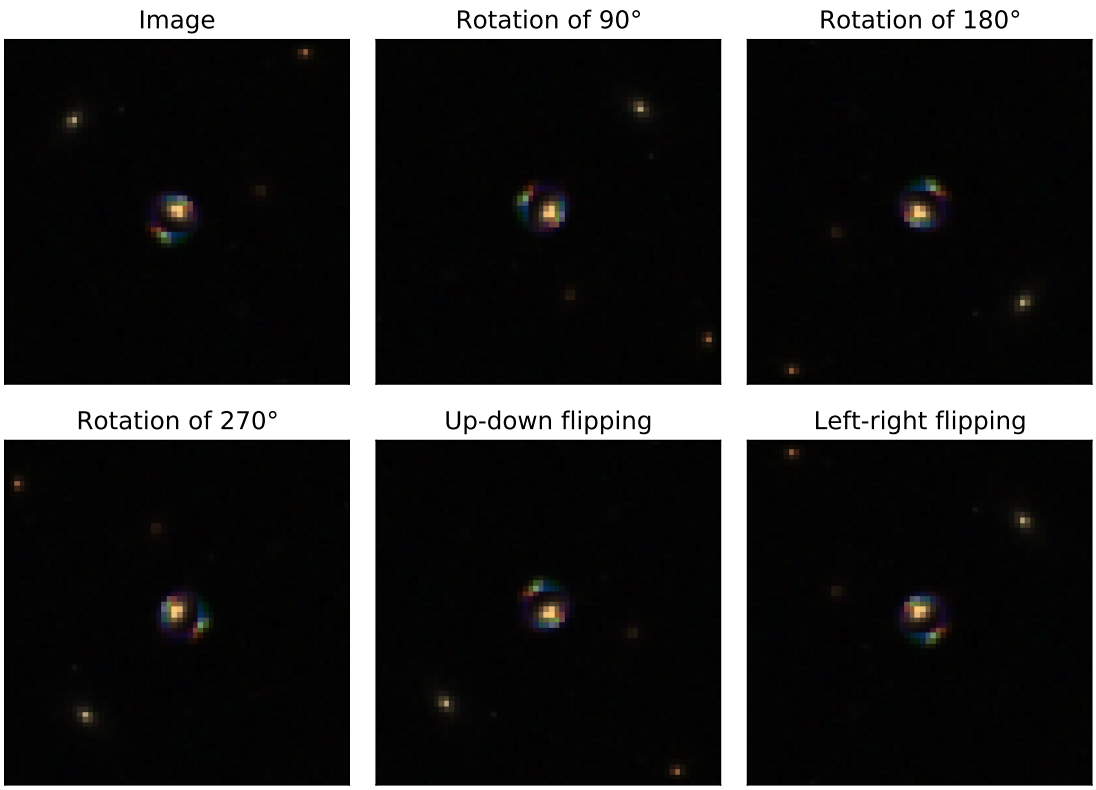


Figure 5.2: This figure shows the result of the operations of augmentation carried out on the 20% of the training set. The original image (top left) is rotated of 90° (top center), of 180° (top left) and of 270° (bottom left) and it is flipped up-down (bottom center) and left-right (bottom right).

Figure 5.2 shows, as an example, the result of these transformations on a randomly selected image of the Lens Finding Challenge dataset. The application of these operations has the twofold purpose of increasing the amount of images in the training set and of making the models invariant to the operations carried out.
Afterwards, we normalize every image I as:

$$I_{norm} = \frac{I - \mu_{train}}{\sigma_{train}} \tag{5.1}$$

where I_{norm} is the normalized image, μ_{train} is the mean of the training set and σ_{mean} is the standard deviation of the training set. This computation is performed pixel-wise. The mean and standard deviation are computed considering the elements in the training set and also used in the normalization of the images in the test and validation sets.
The reason for this type of normalization is that the computation of the gradients in the training stage of the models is easier if the features of the dataset are in a similar range. Moreover, scaling the inputs in this way makes the parameter sharing more efficient (Goodfellow et al., 2016).
We carry out the steps described in this Section by employing several functions

Table 5.1: Confusion matrix in the case of a binary classification problem. The rows indicate the true class of the objects, while the columns indicate the predicted one. The main diagonal elements contain the amount of correctly classified objects: the True Positives (TP) and the True Negatives (TN).

	Predicted class: Positive	Predicted class: Negative
Actual class: Positive	**TP**	**FN**
Actual class: Negative	**FP**	**TN**

available in the `NumPy`[1] package, treating the images as matrices of pixels.

5.2 Performance evaluation

We evaluate the performance of our models by means of four statistical estimators that can be directly derived from the *confusion matrix* (Stehman, 1997). Every element of the confusion matrix C_{ij} indicates the amount of objects belonging to the class i and classified as members of class j.
Table 5.1 displays the structure of the confusion matrix in a binary classification problem, in which each object can belong to one of two categories. In particular:

- C_{00} is the number of objects that belong to the class True and are classified as True, namely the *True Positives* (TP);
- C_{11} is the number of objects that belong to the class False and are classified as False, namely the *True Negatives* (TN);
- C_{01} is the number of objects that belong to the class True, but are classified as False, namely the *False Negatives* (FN);
- C_{10} is the number of objects that belong to the class False, but are classified as True, namely the *False Positives* (FP).

These quantities can be combined to obtain four important estimators of the quality of the classification:

1. The *precision* or *purity* of a given class is the ratio between the number of objects correctly classified as belonging to the class and the sum of all of the objects that are classified as belonging to the class.

   For example, the precision of the class Positive is computed as

$$P = \frac{TP}{TP + FP}. \tag{5.2}$$

[1] https://numpy.org/

This estimator measures the ability of the model not to classify an object as being part of a certain category, if it belongs to another one.

2. The *completeness* or *recall* of a given class is the ratio between the number of objects correctly classified as belonging to the class and the amount of objects that are actually part of the class.

   For example, the recall of the class Positive is computed as

   $$R = \frac{TP}{TP + FN}. \tag{5.3}$$

   This estimator measures the ability of the classifier to identify all the objects that belong to a certain class.

3. The F_β *score* estimator combines the precision and recall to compute the score. Depending on the value of β, one of the two quantities might be weighted more than the other:

   $$F_\beta = (1 + \beta)^2 \frac{P \cdot R}{\beta^2 P + R}. \tag{5.4}$$

   For $\beta = 1$ we obtain the F1-score, that equally weighs the precision and recall.

4. The *accuracy classification score* is the ratio between the number of correctly classified objects and the amount of objects in the dataset:

   $$A = \frac{TP + TN}{TP + TN + FP + FN}. \tag{5.5}$$

In addition to these metrics, we also compute the *Receiver Operating Characteristic* (ROC) (Hanley, 1982) for each one of our tests. The ROC estimates the performance of the classifier as a function of the discrimination threshold, i.e. the probability at which a certain class is chosen over another one. In particular, the ROC curve is obtained by plotting the True Positive Rate (TPR) versus the False Positive Rate (FPR) at various classification thresholds.
The area under the ROC curve (AUC or AUROC) summarizes the information conveyed by the ROC: the closer the AUC is to 1 the better is the performance of the classifier, while it is useless when the AUC = 0.5.
In practice, the output of our models in the test phase is the probability that each of the instances in the test set belongs to one of the two classes, "lens" or "non lens". From this probability, we can establish what the correct classification would be according to our model's predictions. We use the functions contained in the package `sklearn`[2] of Python to compute the estimators just listed, using the truth table and the predictions as input.

[2] https://scikit-learn.org/stable/

5.3 Training on the Lens Finding Challenge Dataset dataset

The application of the criteria 4.1 for dividing the images of the Lens Finding Challenge dataset into the classes of lenses and non lenses ensures that the objects in the first category are at least characterized by the presence of one image of the background source.
However, due to several factors, such as the possible weak magnification caused by the lens and the high magnitude of the background sources, the multiple images of the sources are only weakly observable in a significant fraction of the images. Moreover, other typical features caused by the lensing phenomenon, namely the presence of ring-shaped and arc-shaped sources, are evident in only a subset of the images in the category considered. For instance, some of the images displayed in Figure 4.1 and 4.2 are quite similar, despite being classified as members of different categories.
The presence of borderline objects in the dataset is clearly influential in the performance of the networks, since their correct classification is more difficult to obtain. Even though CNNs are quite versatile in extracting from a training set the characteristics that will be the most useful for assigning new data to the appropriate categories, the recurring occurrence of non immediately recognizable examples might have a significant impact on the ability of the network of detecting the relevant features.
We aim to investigate the capability of the models described in Appendix A to identify lensing features in borderline objects and to verify what are the physical properties of the lenses and sources simulated in the images that are correctly classified. For this purpose, we train our models on different portions of the dataset, that gradually include a larger amount of images and a greater fraction of objects whose classification might be ambiguous.

5.3.1 Data Selection

We consider six selections of data, named from S1 to S6, that consist of approximately 2000, 10000, 20000, 40000, 60000 and the total amount of images respectively. The criteria we adopt to progressively include the less easily classifiable images in our selections take into consideration the characteristics of the images that might be exploited by the networks to classify the objects as members of the correct category. In the case of the images that belong to the class of the non lenses, the lack of a background source, or the absence of an image of it, make the classification of the objects more likely to be correct.
Therefore, we initially consider the objects in the the dataset that are characterized by `n_sources` $= 0$, thus managing to obtain up to approximately 10000 images of non lenses. Afterwards, we broaden our sample by including the images where a background source has been added, but does not correspond to a visible image, i.e. the images that are characterized by `n_source_im` $= 0$. Finally, we extend our selection to the other objects that are classified as non lenses according to the criteria 4.1. Figure 5.3 shows the composition of the non lenses subsets thus obtained in the

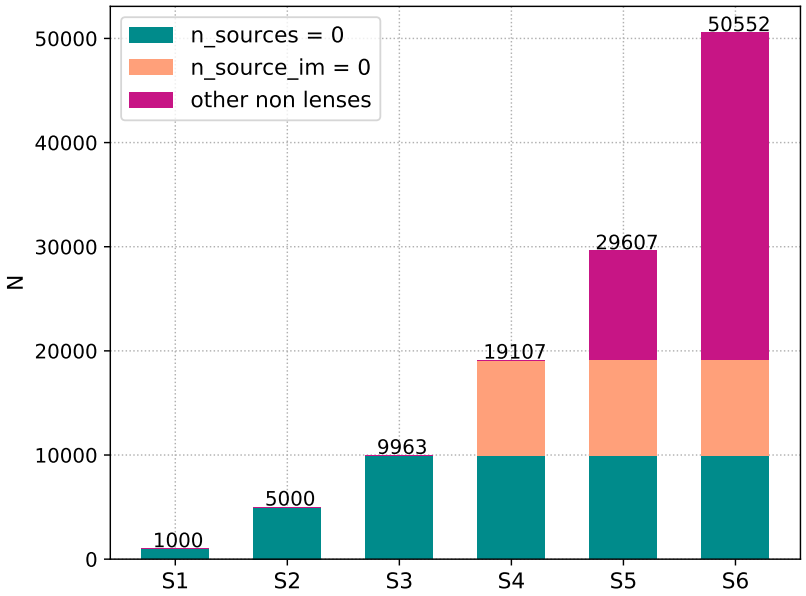


Figure 5.3: This bar chart shows the composition of the subsets of non lenses obtained accordingly to the criteria listed in Table 5.2. The selections S1, S2, S3 only consist of images with `n_sources` = 0; selection S4 also includes objects with `n_source_im` = 0; selections S5 and S6 are partly composed of the other non lenses.

different selections.

In the case of the subset of images classified as lenses, the definition of an effective criterion to identify the most clear examples in the dataset is more important as well as more challenging. In fact, the mere presence of an image of the source does not guarantee a straightforward classification of the image, since several factors contribute to the actual clarity of the observable features. Among them are, for example, the magnitude of the source and the extent of the image produced by the lens, as it was previously mentioned.

After testing the importance of several of the parameters in the catalog associated to the images for discriminating the most evident lenses, we deem `n_pix_source` to be an appropriate choice. As it was explained in the previous Chapter, this parameter describes the amount of pixels where the source is observable. The complete sample of the lenses is characterized by the minimum value `n_pix_source` = 20, as specified in the system 4.1. For each one of the selections we raise this threshold to a different value, that depends on the amount of images we seek to isolate: the higher is the value employed, the smaller will be the amount of images selected and the more evident the lensing features displayed.

The histogram in Figure 5.4 illustrates how the selection of the lenses is gradually broadened. In particular, as it is evident from the distribution, the greatest part of the lenses in the dataset is characterized by relatively small values of `n_pix_source`. In fact, roughly 1000 lenses are identified in the selection S1 considering the objects with `n_pix_source` in the range between 950 and 15000, while the remaining 48000 objects are characterized by a value of `n_pix_source` in the interval between 20 and

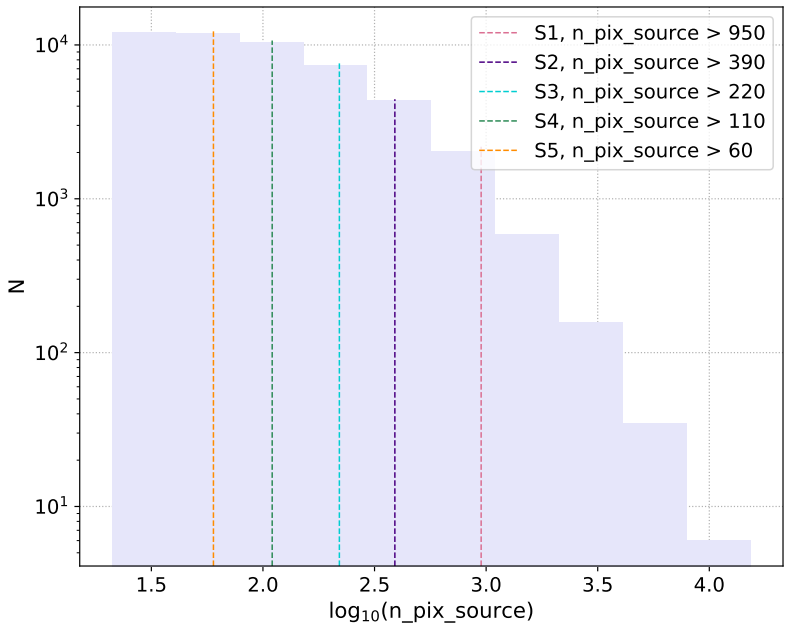


Figure 5.4: This histogram displays the distribution of `n_pix_source` for the complete sample of lenses in the dataset. The vertical lines correspond to the thresholds adopted to achieve the different selections: they are progressively less strict from the right to the left.

950 and sequentially included in the largest samples.
The thresholds established for the creation of the selections described so far also take into account the necessity to have comparable amounts of images in each of the two classes, so that the examples analyzed by the networks in the training phase are balanced.
Table 5.2 contains a detailed summary of the characteristics of the images that are part of the different selections, including the amount of images identified in each of the classes and the criteria and thresholds applied. It should be noted that the selections have been carried out in such a way that the smaller samples are always included in the larger ones. This is obvious in the case of the lenses, because of the criterion followed, but has also been implemented for the non lenses.
Once we have identified the objects that will be part the selections, we divide the datasets into training, validation and test sets according to the percentages give in Section 5.1. For these experiments we only consider the images simulated in the VIS band: the training is described in Appendix A.2.

5.3.2 Results

The results of the classification experiments on the Lens Finding Challenge dataset are summarized in Table 5.3. We list there the precision, recall, F1-score, accuracy and AUC obtained from the application of the three models to the selections

Table 5.2: This Table summarizes the criteria adopted to select the images included in the selections of lenses and non lenses used for our experiments with the Lens Finding Challenge dataset. While the identification of the lenses is solely based on the variation of a threshold value for the parameter `n_pix_source`, the identification of the non lenses is primarily based on the possible presence and visibility of a background source.

Selection	**Lenses**		**Non Lenses**		Total amount of images
	Criteria	Amount of images	Criteria	Amount of images	
S1	`n_pix_source` >950	1084	Randomly selected objects with `n_sources` = 0	1000	2084
S2	`n_pix_source` >390	4995	Randomly selected objects with `n_sources` = 0	5000	9995
S3	`n_pix_source` >220	9977	Whole sample of objects with `n_sources` = 0	9963	19940
S4	`n_pix_source` >110	19191	Whole sample of objects with `n_source_im` = 0	19107	38298
S5	`n_pix_source` >60	29704	Randomly selected objects in the whole sample	29607	59311
S6	Whole sample of lenses	49060	Whole sample of non lenses	50552	99612

considered.
Figures 5.5, 5.6, 5.7 show the confusion matrices obtained by testing the VGG-like Network, the Inception Network and the Residual Network, respectively, on the six selections of data. Moreover, Figure 5.8 displays the ROC curves realized from the predictions of the different models.
By comparing the results of the experiments carried out with our models we can highlight a series of interesting results regarding the relationship between the presence of borderline objects in the dataset and the overall performance of the networks. First of all, it is clear that the ability of the networks to correctly classify the objects in the dataset tends to deteriorate as the percentage of borderline objects included in the dataset increases. This is evident from Figure 5.9, that shows the trend of the accuracy with the selections and from Figure 5.10, that shows the trends of the other metrics considered. This last Figure shows specifically that the degradation of the performance does not only concern the classification of the images that are part of the class of the lenses, but it involves the objects of both the categories.
In particular, the three models succeed in the classification of the objects in the selections S1, S2 and S3, as it is shown from the metrics calculated, that range from ~ 0.85 to ~ 0.95 with only slight differences between different networks. On the other hand, they perform gradually worse on the selections S4, S5 and S6: the accuracy evaluated on this last selection is ~ 0.65 for the three models.
We will now focus on the selection S6, that includes the complete subsets of lenses and non lenses. Figures 5.11 and 5.12 show some of the images in the test set of this selection that are misclassified: Figure 5.11 shows 12 images that are classified as non lenses while being part of the lens category and Figure 5.12 shows 12 images that are classified as lenses while being part of the non lens class. It should be noted that the objects represented in these Figures are incorrectly classified by all the three models, therefore they should be characterized by the features that the models generally find harder to attribute to the correct class.
We can make two comments about these images:

1. The images mistakenly classified as non lenses in Figure 5.11 are in large part not even recognizable as lenses by visual inspection. Although being classified as lenses according to the criteria 4.1, many of these objects do not show evident lensing features. Therefore, if the classification was to be carried out on real observations, we would not expect the models to be able to identify them as lenses. In some of the images, however, the presence of arc-shaped sources is evident. Nevertheless, their classification is incorrect: this issue is discussed in greater detail later in this Section.

2. The images erroneously classified as lenses in Figure 5.12 are often characterized by the presence of more than one source in addition to the lens galaxy. The coexistence of several objects in the images might be mistaken for the presence of multiple images of the same source by the networks. Moreover, in some of these images it might be noticed the presence of spiral galaxies, whose arms might be interpreted as faint arc-shaped or ring-shaped features by the models. In particular, the misinterpretation of these features might be exacerbated by the presence of several images in the subset of the lenses in

Table 5.3: Summary of the performance of the VGG-like Network, the Inception Network and the Residual Network in the classification of the objects of the six selections of data. The precision, recall and F1-score are evaluated on the class of the non lenses (0) and of the lenses (1) separately.

VGG-like Network												
	S1		S2		S3		S4		S5		S6	
	0	1	0	1	0	1	0	1	0	1	0	1
Precision	0.89	0.98	0.95	0.98	0.95	0.95	0.89	0.92	0.81	0.85	0.65	0.65
Recall	0.98	0.91	0.98	0.95	0.95	0.95	0.92	0.88	0.86	0.81	0.68	0.62
F1-score	0.93	0.95	0.96	0.96	0.95	0.95	0.90	0.90	0.83	0.83	0.66	0.64
Accuracy	0.94		0.96		0.95		0.90		0.83		0.65	
AUC	0.72		0.74		0.81		0.78		0.68		0.60	
Inception Network												
	S1		S2		S3		S4		S5		S6	
	0	1	0	1	0	1	0	1	0	1	0	1
Precision	0.91	0.97	0.94	0.96	0.92	0.92	0.85	0.88	0.73	0.76	0.62	0.70
Recall	0.96	0.93	0.97	0.93	0.92	0.92	0.88	0.84	0.77	0.72	0.80	0.48
F1-score	0.94	0.95	0.95	0.95	0.92	0.92	0.86	0.86	0.75	0.74	0.70	0.57
Accuracy	0.94		0.95		0.92		0.86		0.74		0.65	
AUC	0.87		0.82		0.80		0.76		0.69		0.62	
Residual Network												
	S1		S2		S3		S4		S5		S6	
	0	1	0	1	0	1	0	1	0	1	0	1
Precision	0.85	0.99	0.94	0.91	0.91	0.8	0.81	0.92	0.70	0.75	0.64	0.64
Recall	0.99	0.87	0.91	0.94	0.90	0.91	0.93	0.79	0.78	0.66	0.66	0.61
F1-score	0.92	0.93	0.93	0.93	0.91	0.90	0.87	0.85	0.73	0.70	0.65	0.62
Accuracy	0.92		0.93		0.90		0.86		0.72		0.64	
AUC	0.88		0.82		0.81		0.78		0.68		0.61	

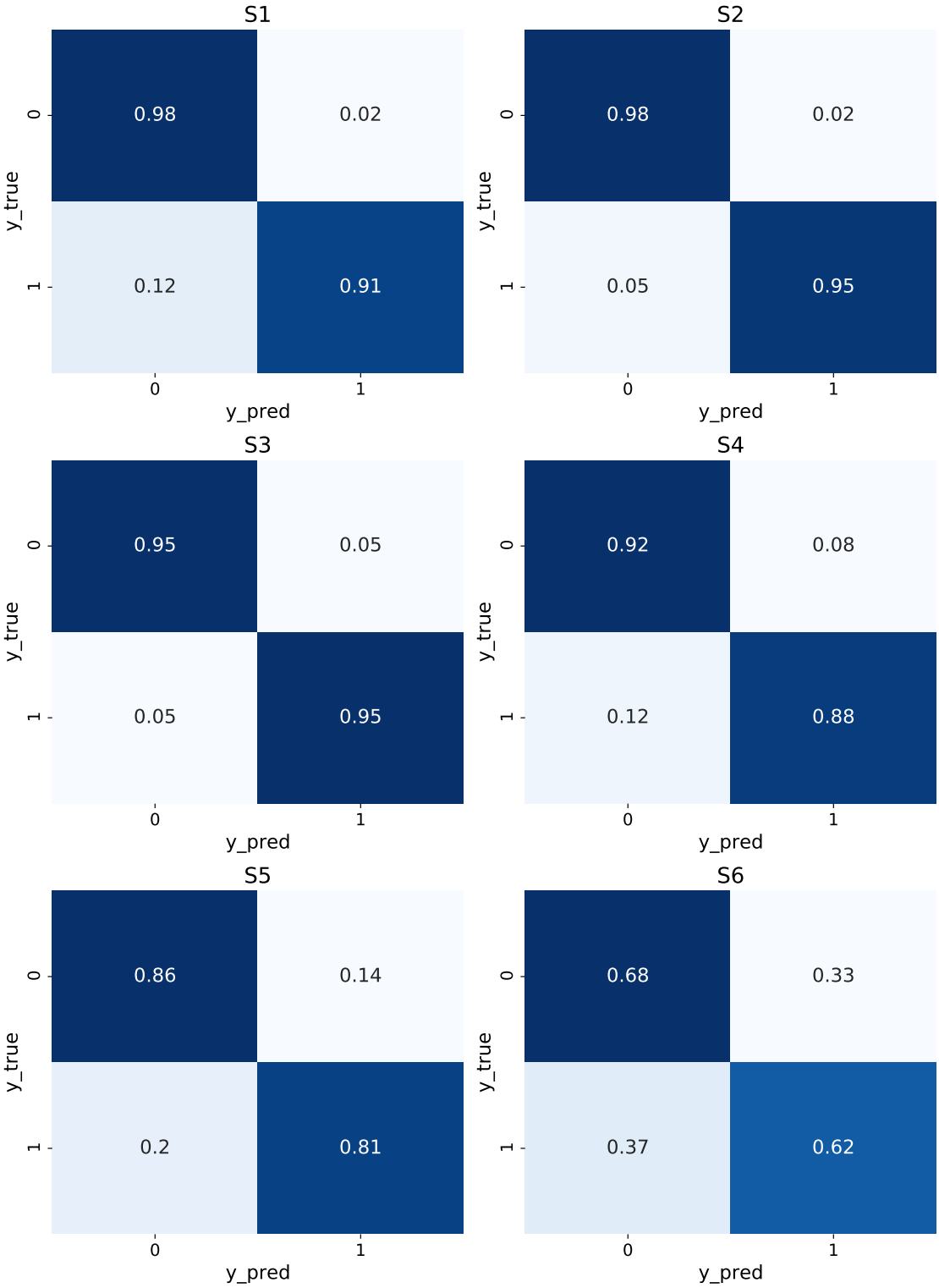


Figure 5.5: From the top panel on the left, to the bottom panel on the right, the confusion matrices obtained by testing the VGG-like Network on the test sets of the selections of data from S1 to S6.

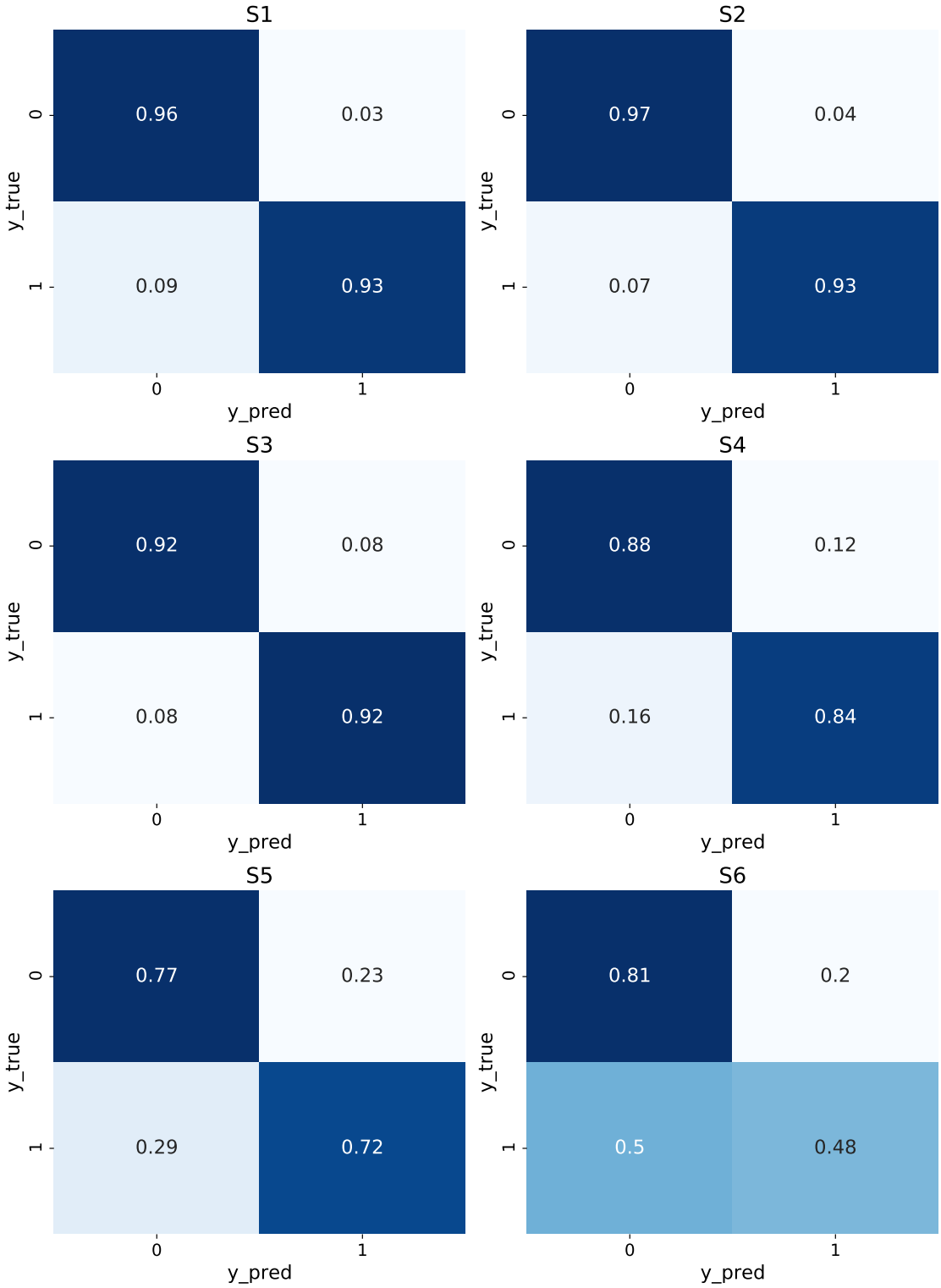


Figure 5.6: From the top panel on the left, to the bottom panel on the right, the confusion matrices obtained by testing the Inception Network on the test sets of the selections of data from S1 to S6.

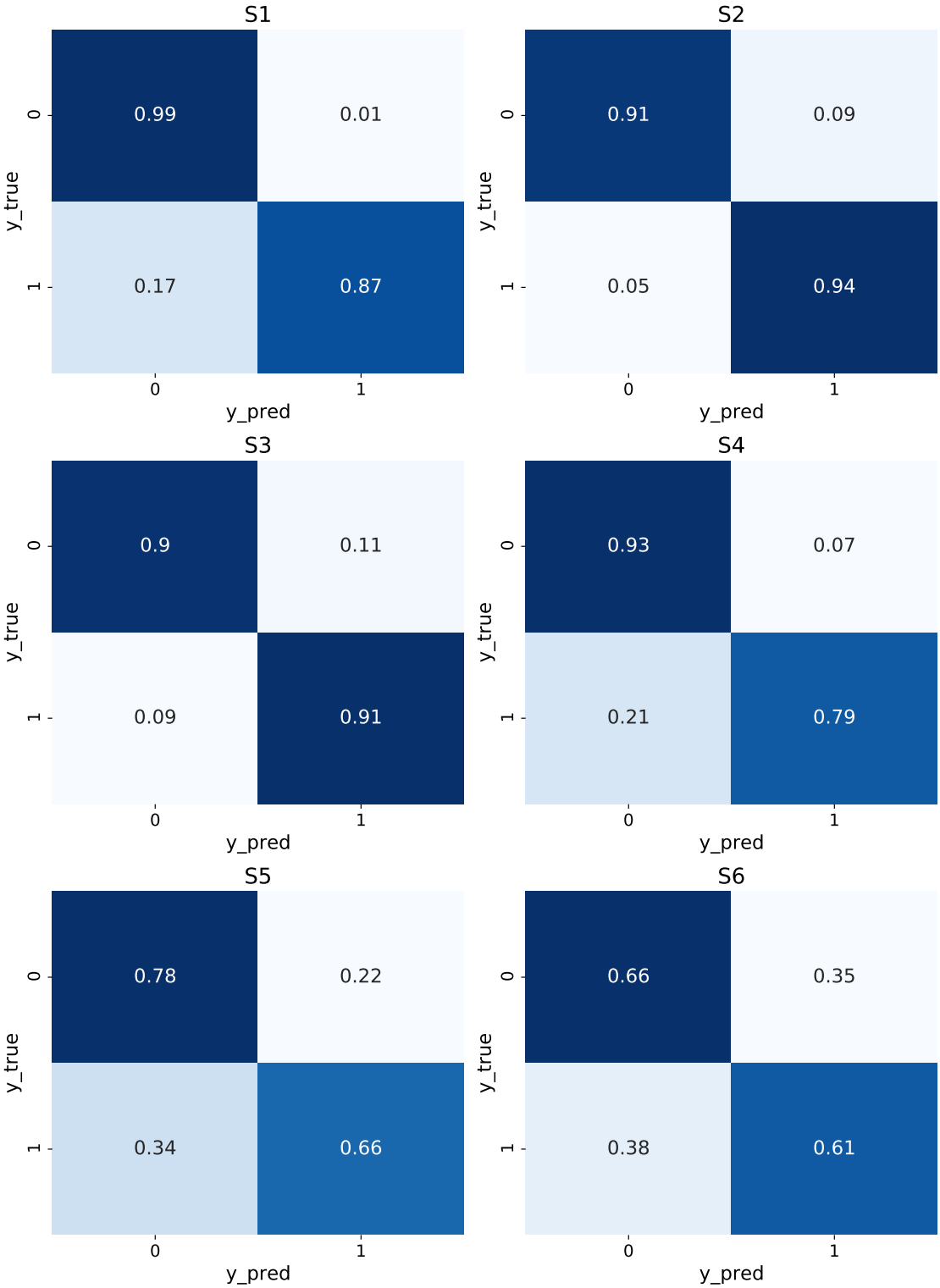


Figure 5.7: From the top panel on the left, to the bottom panel on the right, the confusion matrices obtained by testing the Residual Network on the test sets of the selections of data from S1 to S6.

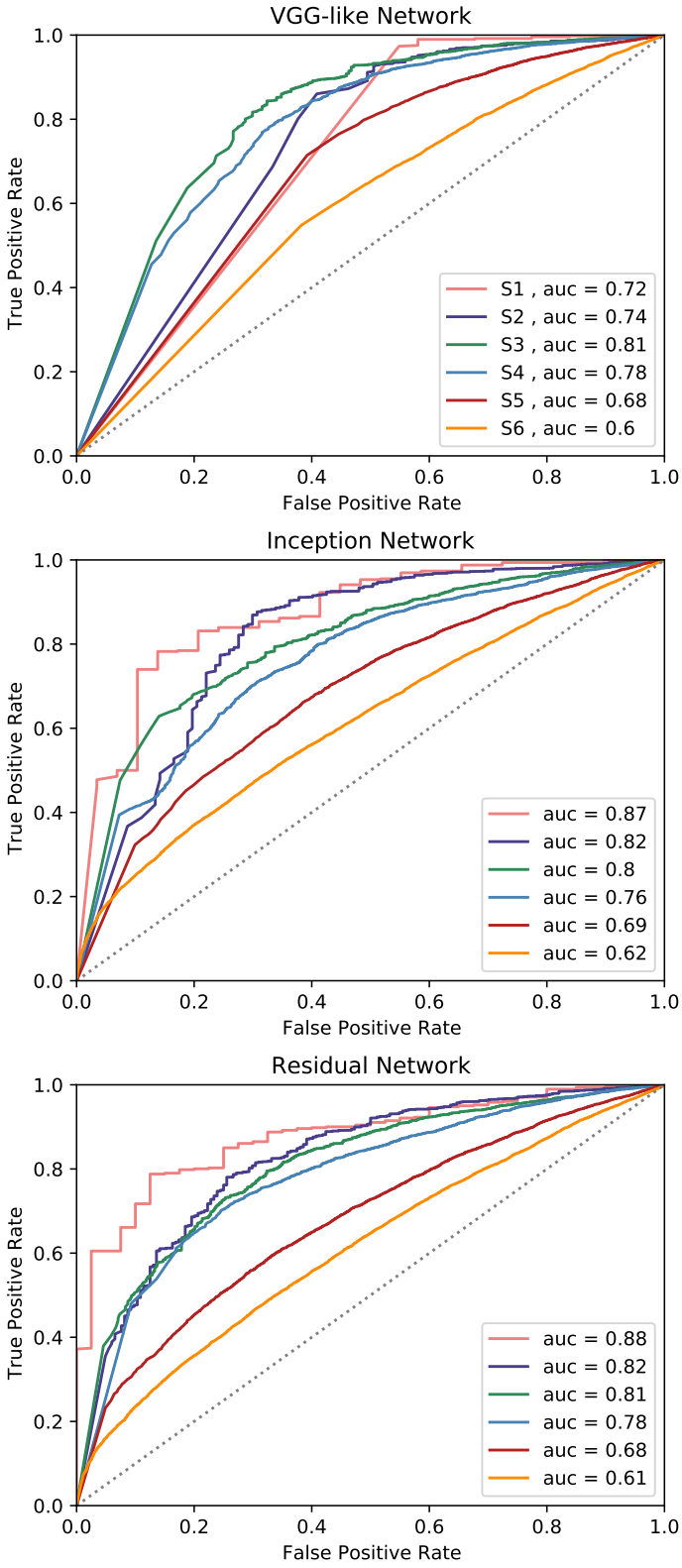


Figure 5.8: Each panel of this Figure displays the ROC curves obtained from the application of one architecture to the test sets of the different selections of data. From top to bottom, the VGG-like Network, the Inception Network and the Residual Network are considered.

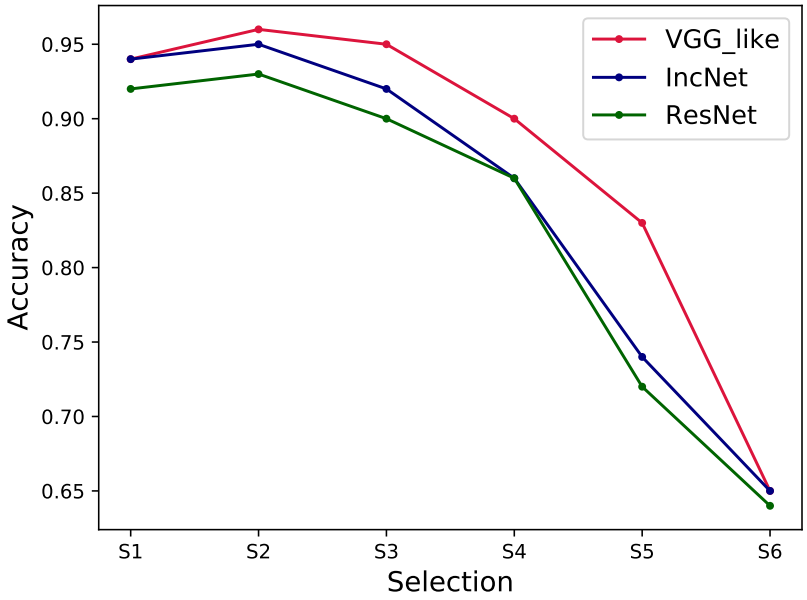


Figure 5.9: Trend of the classification accuracy of the three CNNs tested on the six selections of data.

the training set that do not actually display clear lensing features.

In order to further investigate the effect of the inclusion of the borderline objects in the dataset we test the models trained on S6 on the selection S3, that is characterized by more evident objects in both the subsets. The confusion matrices obtained from this test are in Figure 5.13, along with the ROC curves. In particular, we can analyze these results with respect to those obtained from training and testing the networks on the selection S3: this comparison is carried out in Table 5.4.
It is clear that the overall performance of the models trained on the selection S6 is worse than that of the models trained on S3, in the identification of the images that show lensing features as well as in the classification of the non lenses. In this context, it is important to emphasize that the objects that are part of the selection S3 will also inevitably be part of S6, since it corresponds to the complete dataset.
This observation indicates that the performance of the models trained on S6 is not only worse in a general sense, as we may expect since the borderline objects are intrinsically harder to classify and a large fraction of this selection is composed of non obvious lenses. In addition to this, the ability of the models to identify the clearest objects in the dataset, as the ones that are part of S3, also deteriorates.
This might depend on a combination of two complementary factors, regarding the training of the networks on the complete dataset. First of all, the percentage of clear objects in the training set of the selection S6 is smaller than in the other selections because of the relevant portion of borderline objects included. This reflects in the fact that the networks might not learn how to properly classify them. In fact, wide arcs and rings will be recognizable in only a moderate fraction of images, thus not being as relevant as they are in S3 when extracting the most relevant features in

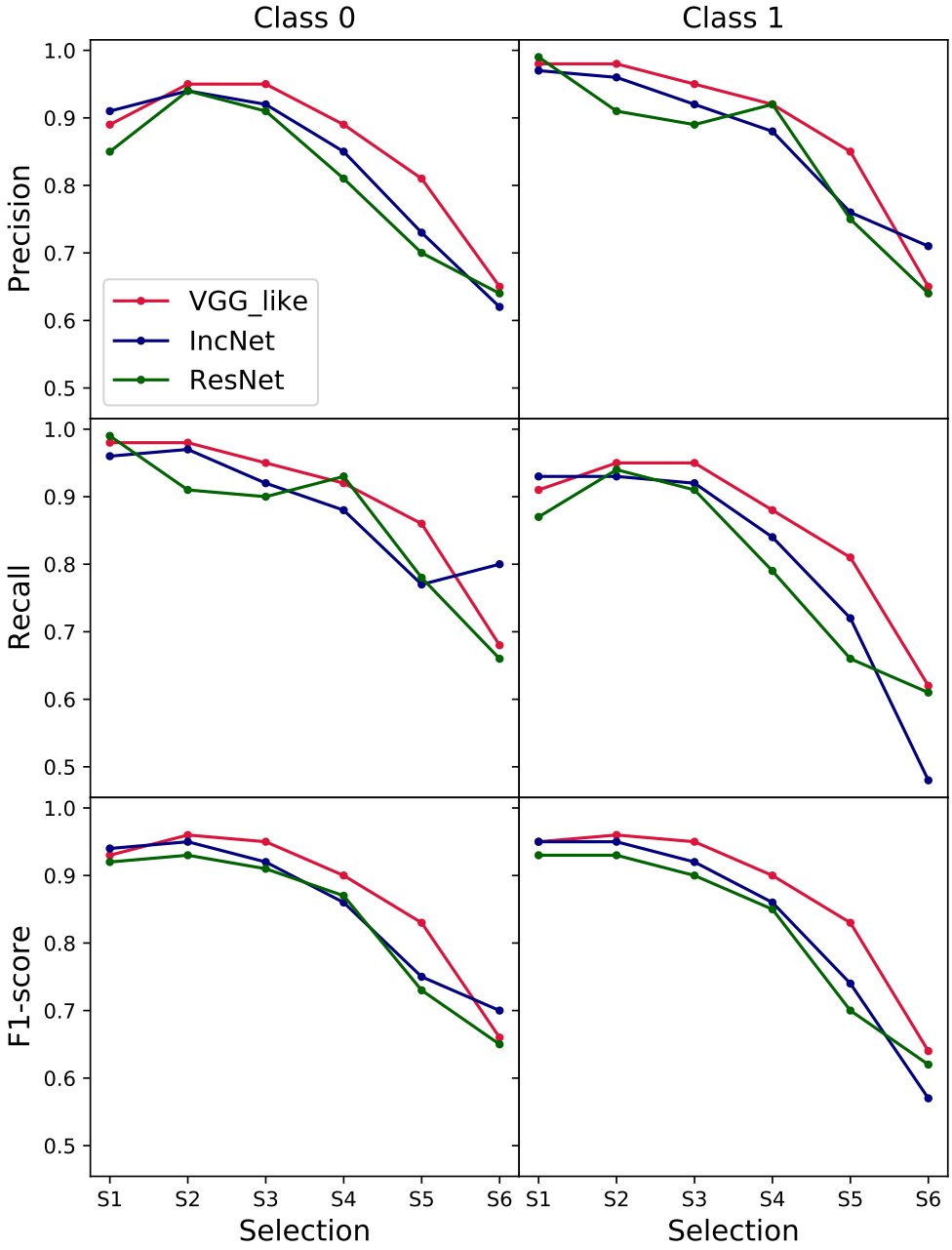


Figure 5.10: Trend of the precision (first row), recall (second row) and F1-score (third row) in the classification of the images that belong to the class of the non lenses (class 0, first column) and to the one of the lenses (class 1, second column) in the different selections.

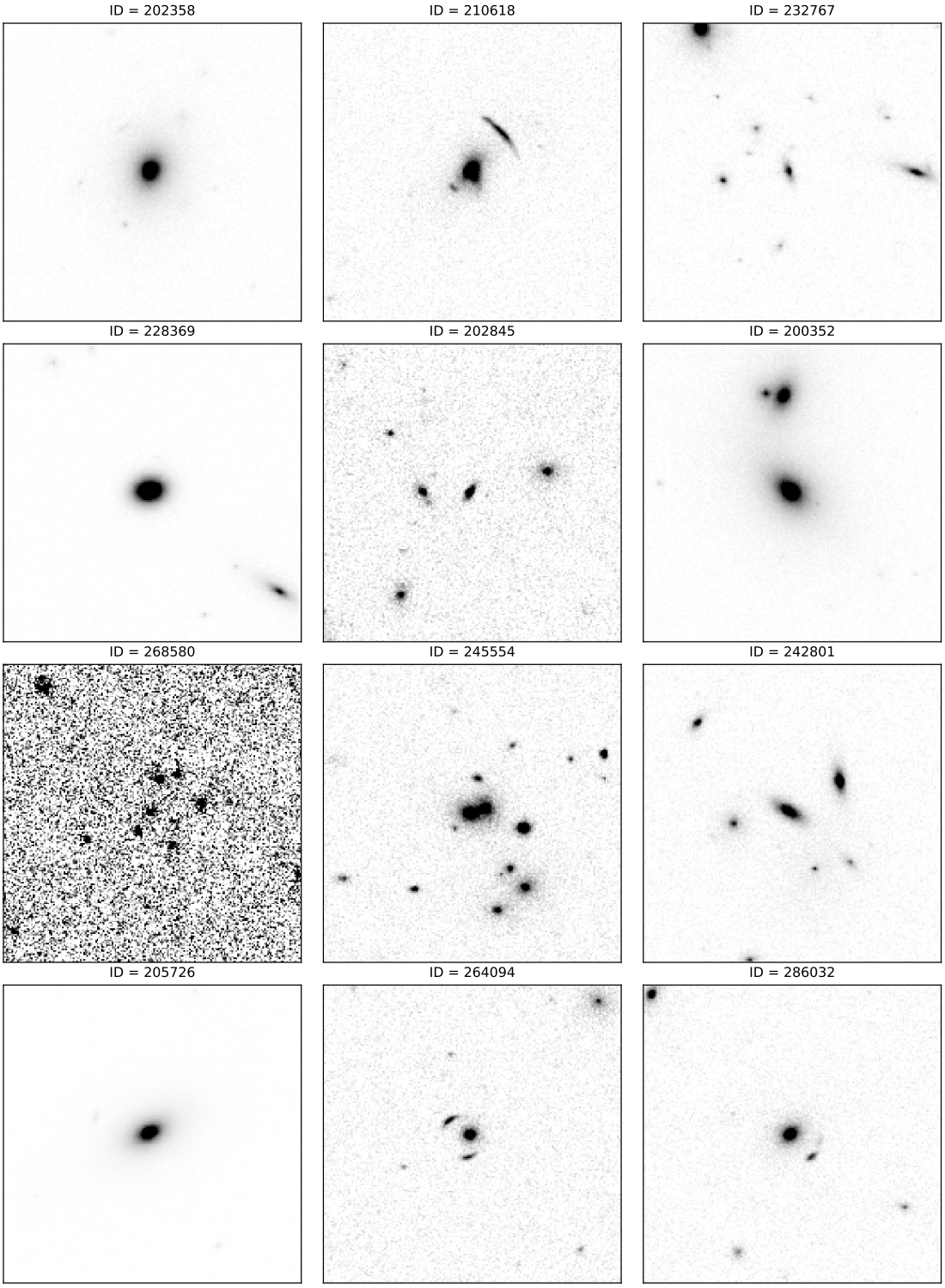


Figure 5.11: Images in the test set of the selection S6 that should be classified as lenses according to the criteria 4.1, but are classified as non lenses by the three models.

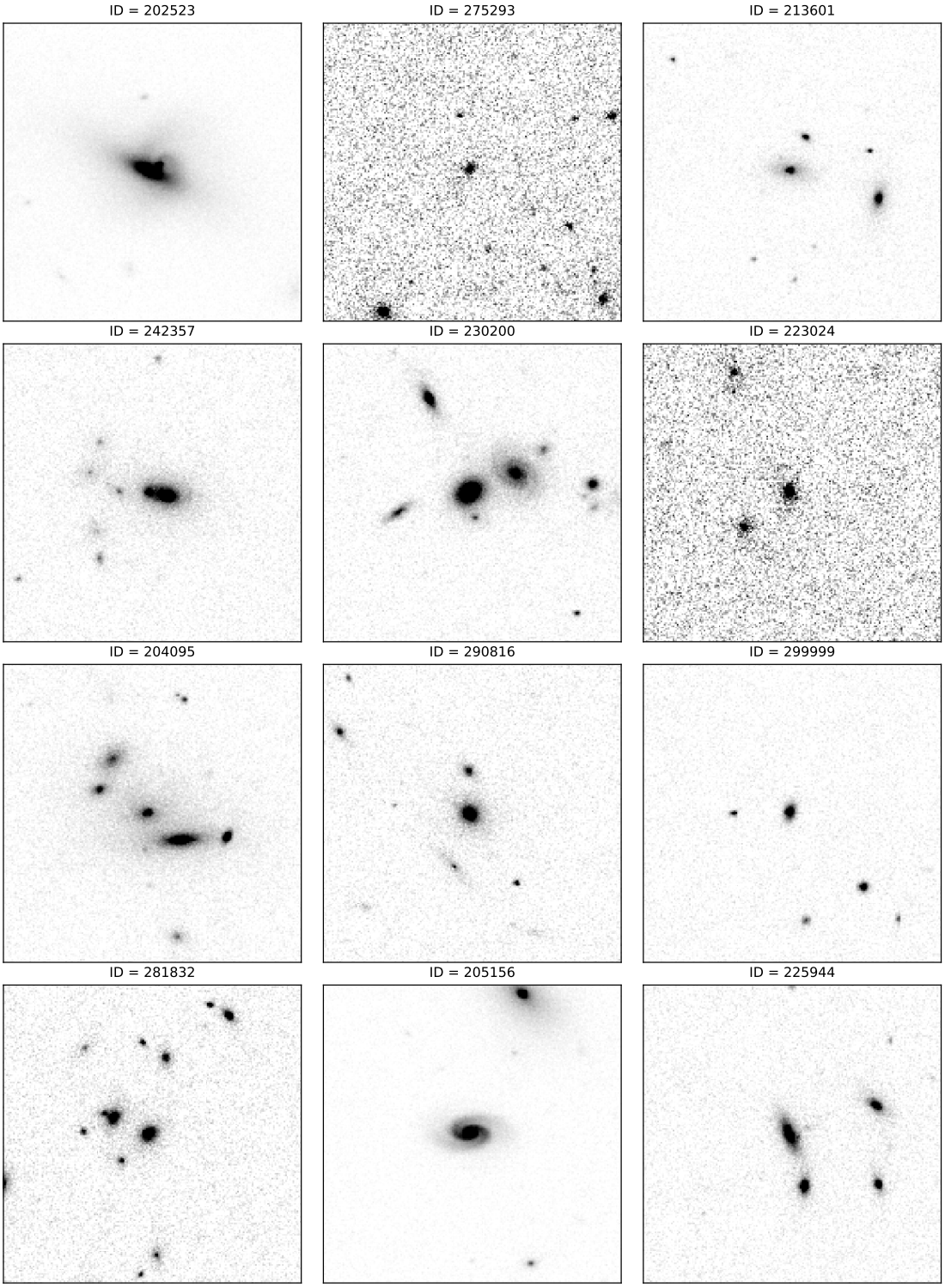


Figure 5.12: Images in the test set of the selection S6 that should be classified as non lenses according to the criteria 4.1, but are classified as lenses by the three models.

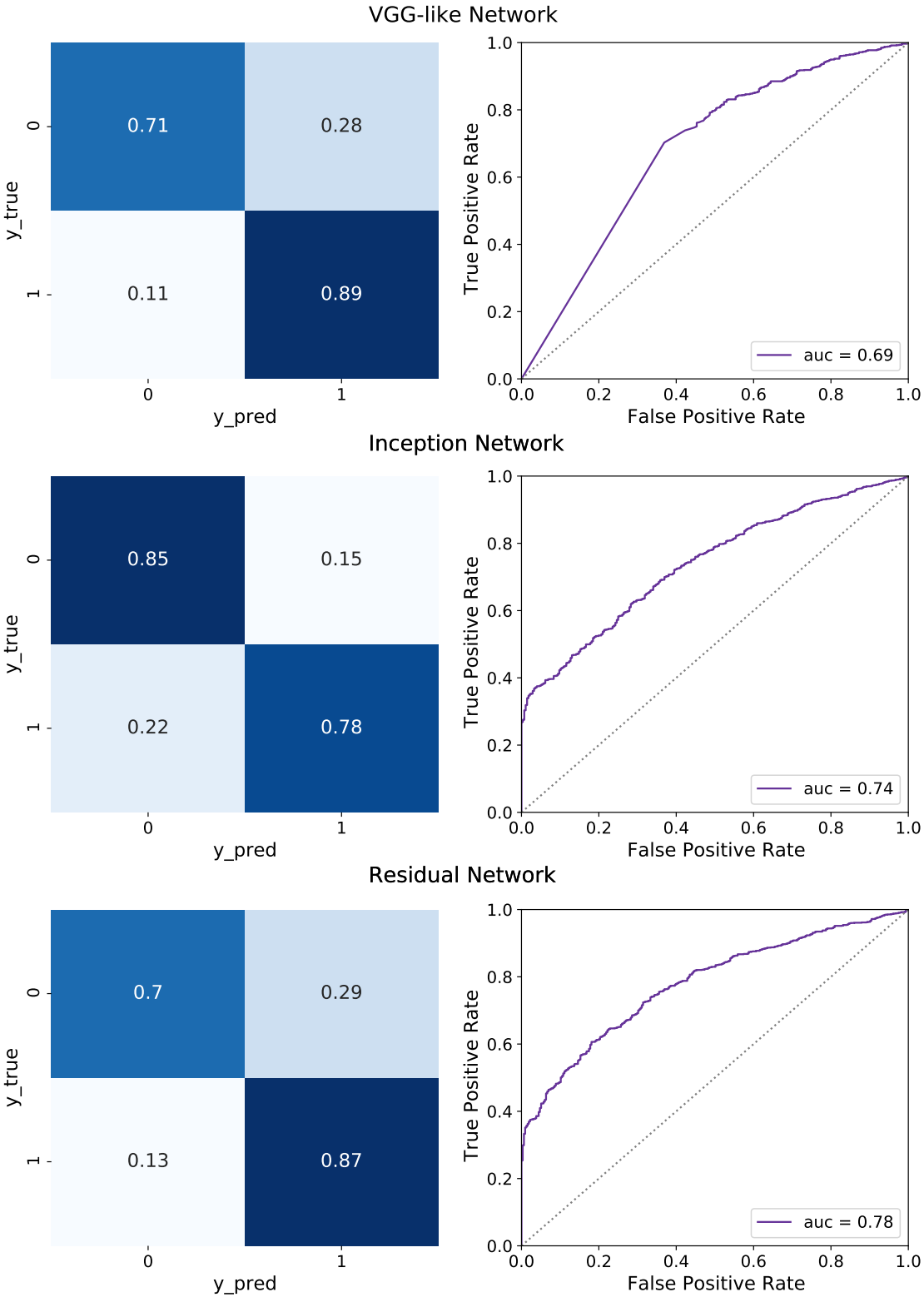


Figure 5.13: Confusion matrices and ROC curves obtained from the test of the VGG-like Network (top row), the Inception Network (central row) and Residual Network (bottom row) trained on the S6 selection on the S3 selection.

Table 5.4: Comparison between the metrics of tests on S3 with the models trained on S3 (top) and on S6 (bottom). Class 0 refers to the non lenses, while class 1 refers to the lenses.

S3/S3						
	VGG-like Network		**Inception Network**		**Residual Network**	
	0	1	0	1	0	1
Precision	0.95	0.95	0.92	0.92	0.91	0.90
Recall	0.95	0.95	0.92	0.92	0.90	0.91
F1-score	0.95	0.95	0.92	0.92	0.91	0.90
Accuracy	0.95		0.92		0.90	
S6/S3						
	VGG-like Network		**Inception Network**		**Residual Network**	
	0	1	0	1	0	1
Precision	0.86	0.76	0.79	0.84	0.84	0.75
Recall	0.71	0.89	0.85	0.79	0.70	0.87
F1-score	0.78	0.82	0.81	0.81	0.76	0.81
Accuracy	0.80		0.81		0.79	

the dataset. At the same time, the most recurrent features in the training set will be the ones that occur in the borderline images, thus concurring to explain the misinterpretation of some of the images that present evident lensing features.

All in all, we might conclude that the largest fraction of the objects that are not correctly identified as lenses by the networks trained on S6 are actually borderline, as the examples in Figure 5.11 point out. However, a certain fraction of evident lenses might also be missed if the training set is extended to include a significant fraction of borderline objects, as they might not be well represented.

In addition to this, we can highlight from the results of our tests that the architecture of the networks does not appear to be particularly influential in the outcome of the classification. In particular, when trained and tested on the same selections, the different architectures generally perform similar to each other.

The only possible exception to this remark is in the performance of the Inception Network on the entire dataset, as it can be seen in Figure 5.6. In fact, the application of this architecture to the selection S6 results into a significantly worse performance than the ones obtained from the other architectures, for what concerns the identification of the lenses in the dataset.

5.4 Additional tests

5.4.1 Trend of the classification with physical properties of lenses and sources

In this Section, we aim to investigate whether there is a correlation between the physical characteristics of the lenses and sources in the images and the ability of the networks to correctly classify them.
We do this by considering the performance of the models on the complete dataset. In particular, we focus on some of the properties of the lenses (the Einstein area, `ein_area` and the halo mass, `halo_mass`), of the sources (the redshift, `z_source` and the magnitude, `mag_source`) and of the simulated images (the amount of pixels where the source is visible, `n_pix_source` and the amount of images of the source, `n_source_im`).
We divide the range of values covered by each one of these parameters into several intervals and assign the elements in the test set of the selection S6 to the different subsets, depending on the value that characterizes each of the images with respect to the parameter considered. At this point, we evaluate the performance of the models in the classification of the objects that belong to the same interval, thus obtaining the trend of the classification with respect to the chosen characteristic.
Figures 5.14, 5.15, 5.16, 5.17, 5.18 and 5.19 show the results of these tests. Each of these Figures focuses on the metrics evaluated for the different models with respect to one of the parameters considered. The normalized histogram of the parameter for the class of the lenses and of the non lenses is shown in the background of the respective column. In particular, the distribution is referred to the training set, but the test set's distribution is very similar, as it was explained in Section 5.1.
In general, the clearest trends concern the precision of the classification, while in the majority of the cases the recall is virtually constant with respect to the parameters considered. Since the F1-score takes into consideration both of these metrics, it often shows the same trends as the precision, even though less evidently. Moreover, even though the overall performance of the networks on this selection is not ideal, as it was shown the previous Section, the models are still able to identify a great fraction of the objects characterized by certain values for the parameters considered in this analysis.
We might refer to the F1-score, that takes into consideration both the precision and the recall. Specifically:

1. The Einstein area of a lens is related to the type of lensing features that the system might display: in case of larger values of this parameter the lenses should be more easily recognizable because of the high distortion of the images of the sources. Figure 5.14 displays a weak trend in this sense.

2. The halo masses of the lenses in the dataset range from the values typical of galaxies to those that characterize small groups of galaxies ($M \sim 10^{14} M_{\odot}$), as it is shown in Figure 5.15. The ability of the classifiers to identify the strong lensing events increases in the range of massive galaxies and shows a weak decrement in the mass range that corresponds to groups of galaxies.

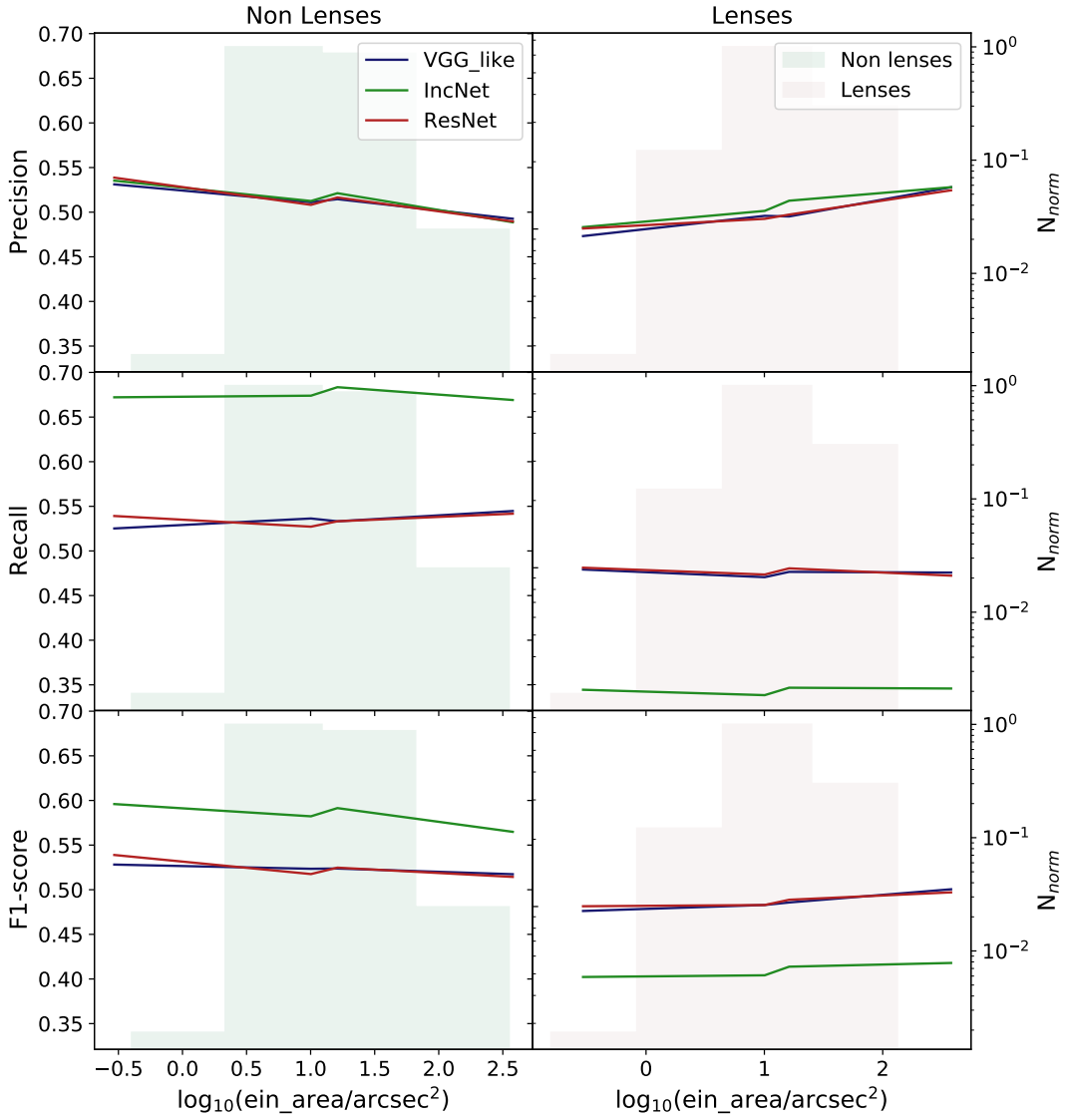


Figure 5.14: Trend of the performance of the VGG-like Network (blue), Inception Network (green) and Residual Network (red) with respect to the Einstein area of the lenses. The histograms in the background represent the distribution of the parameter `ein_area` for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

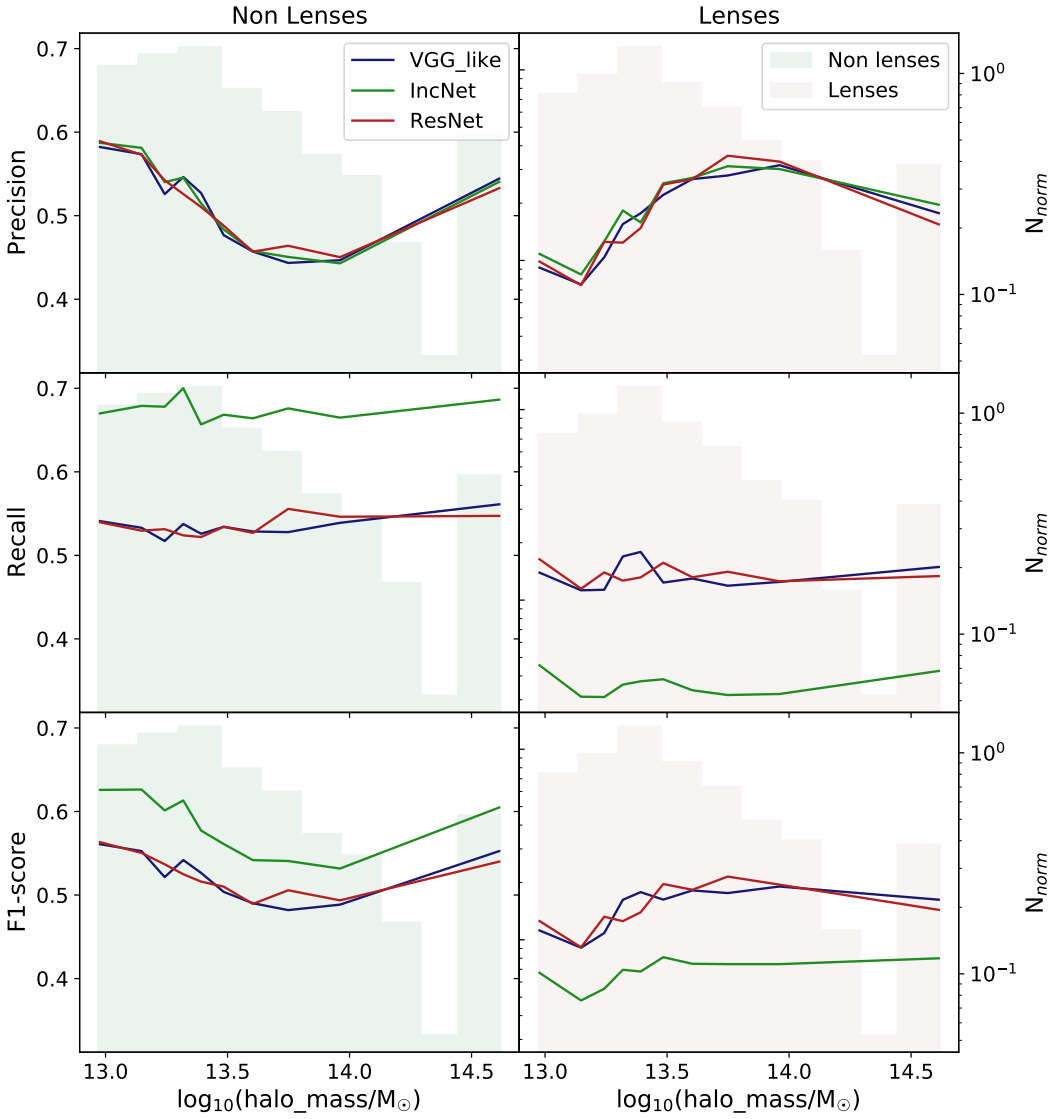


Figure 5.15: Trend of the performance of the VGG-like Network (blue), Inception Network (green) and Residual Network (red) with respect to the halo mass of the lenses. The histograms in the background represent the distribution of the parameter `halo_mass` for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

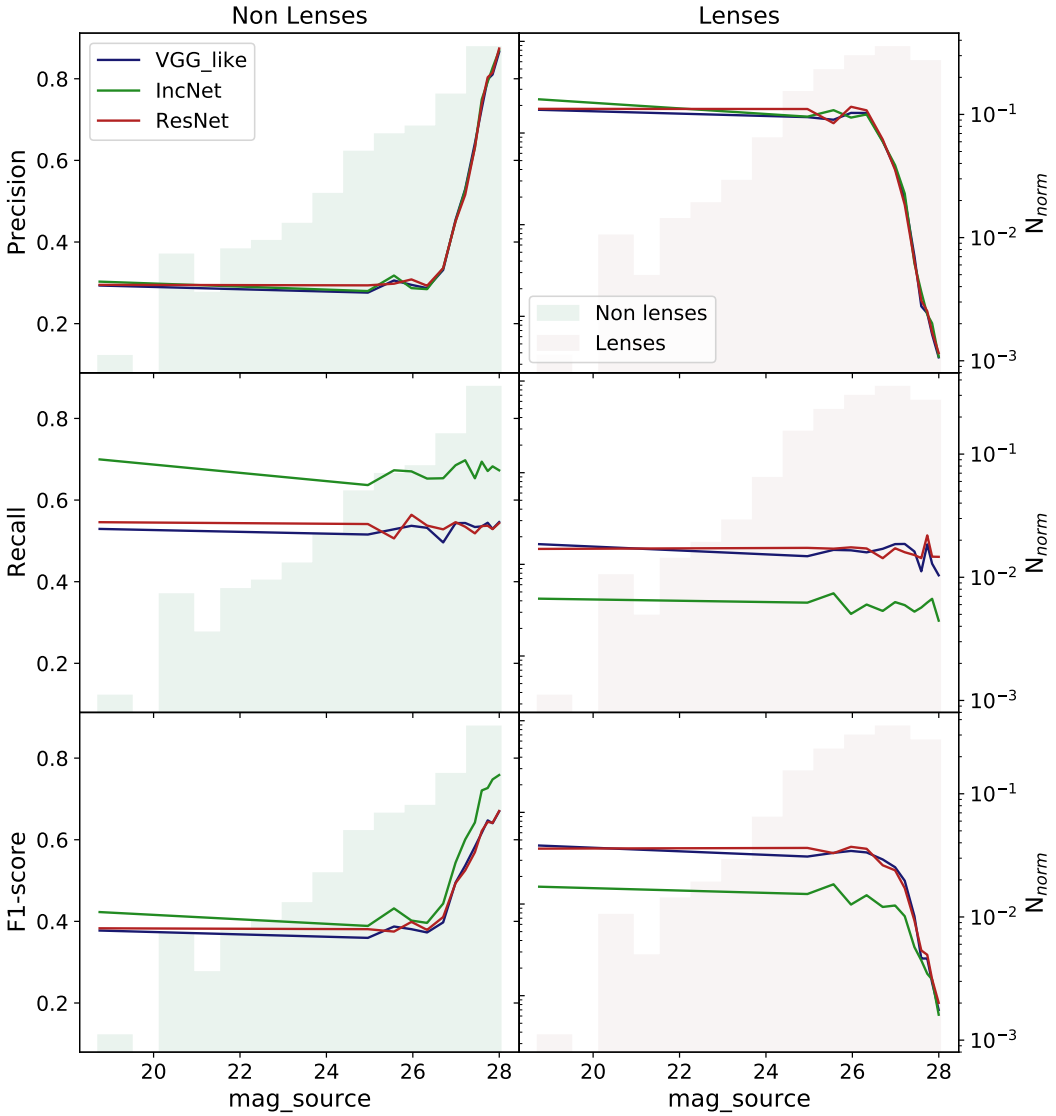


Figure 5.16: Trend of the performance of the VGG-like Network (blue), Inception Network (green) and Residual Network (red) with respect to the magnitude of the sources. The histograms in the background represent the distribution of the parameter `mag_source` for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

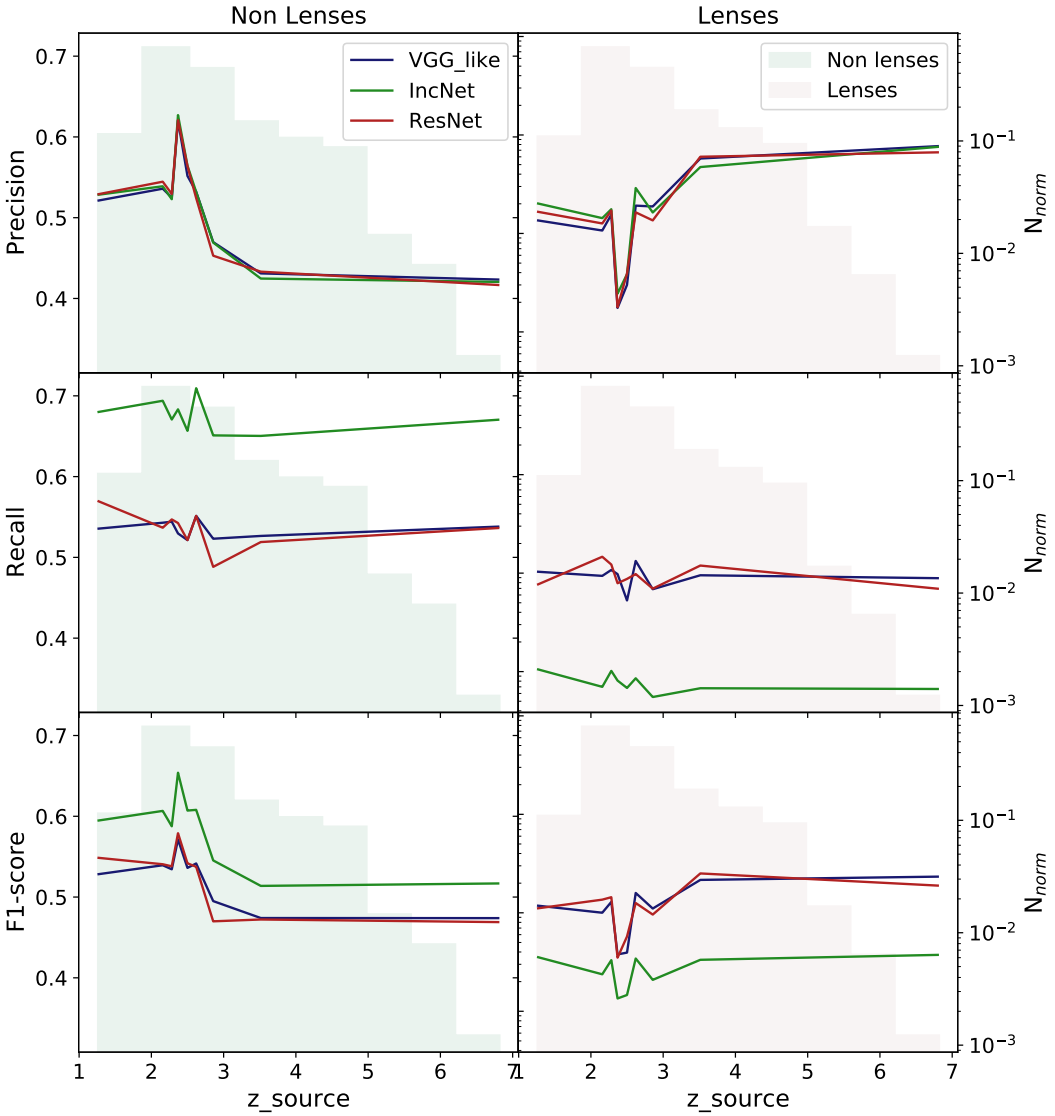


Figure 5.17: Trend of the performance of the VGG-like Network (blue), Inception Network (green) and Residual Network (red) with respect to the redshift of the sources. The histograms in the background represent the distribution of the parameter `z_source` for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

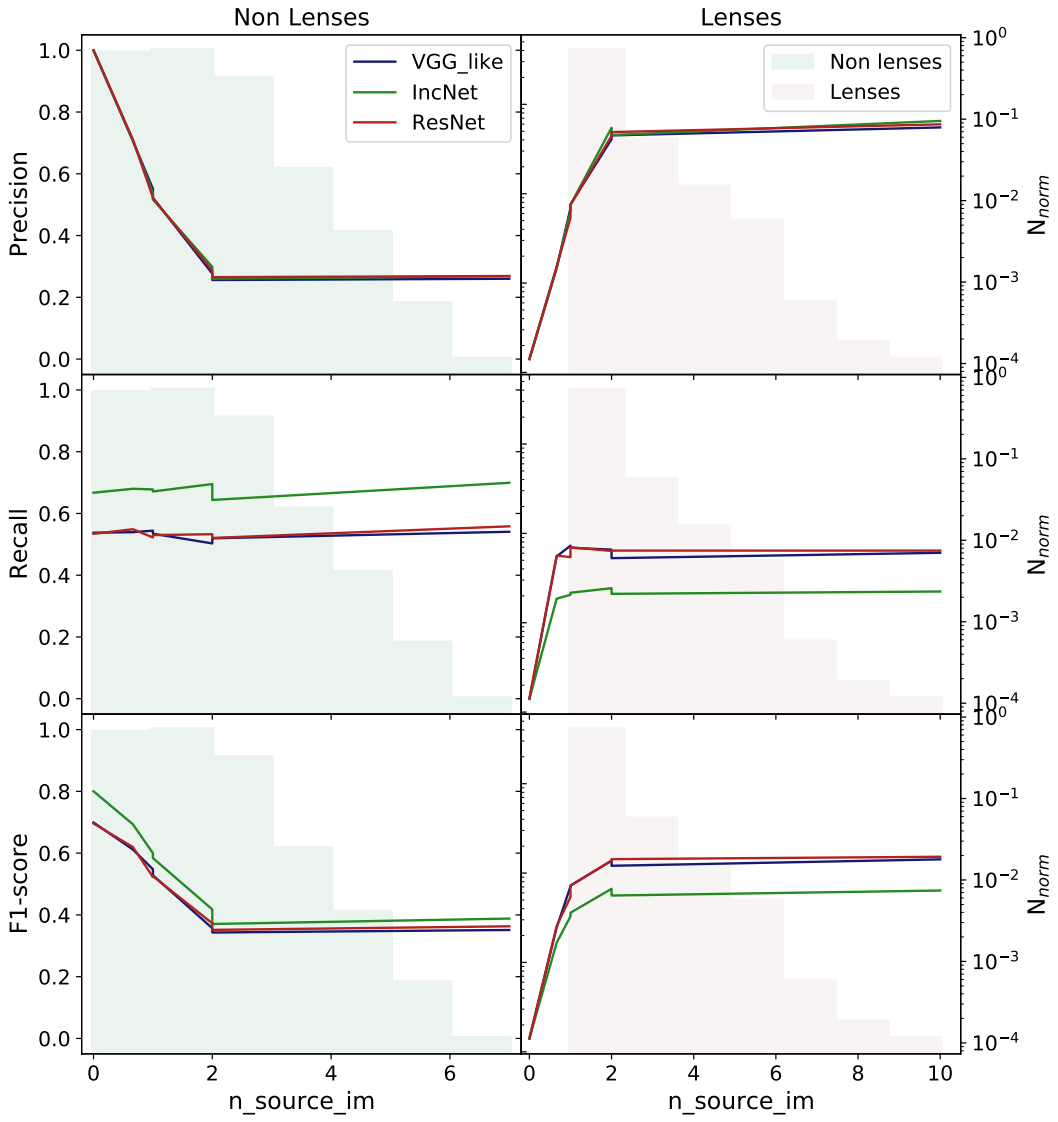


Figure 5.18: Trend of the performance of the VGG-like Network (blue), Inception Network (green) and Residual Network (red) with respect to the amount of multiple images of the same source. The histograms in the background represent the distribution of the parameter `n_source_im` for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

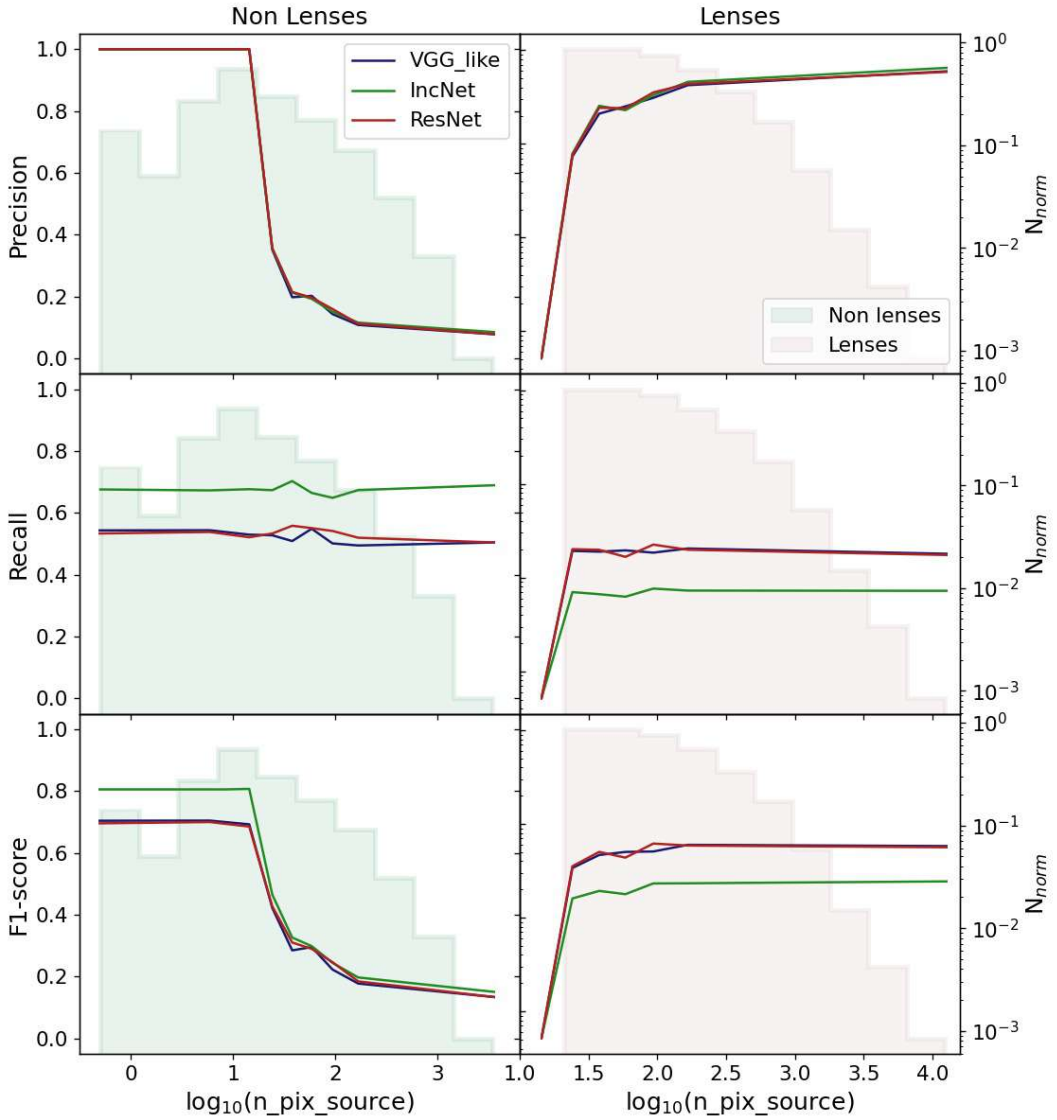


Figure 5.19: Trend of the performance of the VGG-like Network (blue), Inception Network (green) and Residual Network (red) with respect to the amount of pixels where the source is visible. The histograms in the background represent the distribution of the parameter `n_pix_source` for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

3. The magnitude of the background sources has a strong impact on the performance of the models, especially in the case of faint sources, for values of `mag_source` $\gtrsim 26$, as it is shown in Figure 5.16. The lenses are difficult to distinguish when the background source is intrinsically faint, while the identification of non lenses is easier if the background sources are not observable.

4. Sources at high redshifts produce larger Einstein rings, that are more likely to be identified than the ones those at low redshifts. However, these objects are also intrinsically more difficult to observe. In the combination of these two factors, the latter seems to be predominant, as it is shown in 5.17: the identification of the lenses does not improve at high redshifts.

5. As the number of multiple images of the source increases, the ability of the networks to identify the lenses improves, while the opposite happens in the case of the non lenses. This trend, shown in Figure 5.18 indicates that the presence of multiple images of the source facilitates the identification of the lenses, as expected.

   It should be noted that several images of the background source might be visible in the objects classified as non lenses if the criteria regarding the magnification and visibility of the source in Equations 4.1 are not simultaneously satisfied.

6. The amount of pixels where the source is visible, that was used as an indicator of the clarity of the lenses in the identification of the images to include in the different selections, is of course crucial for the correct classification of the elements in the dataset. The trend of the performance with respect to this parameter, shown in Figure 5.19, indicates that the images in which the source is not well observable are mistakenly classified as non lenses more commonly than those that display clear lensing features.

5.4.2 Identification of borderline objects with the models trained on the selection S3

We now aim to assess the flexibility of the models trained on one of the most limited selections, S3, in the identification of borderline objects that are not included in this training set.
For this task, we test the models on the selections S4, S5, S6, after removing the objects of these selections that are also present in the training set of S3. Table 5.5 summarizes the results of our tests. Moreover, Figures 5.20, 5.21 and 5.22 show the confusion matrices and ROC curves obtained by testing the VGG-like Network, Inception Network and Residual Network, respectively.
A first analysis of the results indicates that the general performance of the networks trained on the selection S3 deteriorates on the other selections considered, as it can be seen in Figure 5.24, that shows the trend of the accuracy: it ranges from ~ 0.85 on S4 to $\sin 0.64$ on S6. However, it should be noted that with respect to this particular metric, the performance of the models trained on S3 is not different from

Table 5.5: Summary of the performance of the VGG-like Network, the Inception Network and the Residual Network, trained on the selection S3, in the classification of the objects that are part of the selections S4, S5, S6. The precision, recall and F1-scoreare evaluated on the class of the non lenses (0) and of the lenses (1) separately.

VGG-like Network						
	S3/S4		S3/S5		S3/S6	
	0	1	0	1	0	1
Precision	0.79	0.93	0.68	0.89	0.59	0.79
Recall	0.95	0.75	0.93	0.55	0.91	0.35
F1-score	0.86	0.83	0.78	0.68	0.72	0.49
Accuracy	0.85		0.74		0.64	
AUC	0.8		0.72		0.64	
Inception Network						
	S3/S4		S3/S5		S3/S6	
	0	1	0	1	0	1
Precision	0.78	0.90	0.68	0.84	0.60	0.74
Recall	0.92	0.75	0.89	0.59	0.87	0.39
F1-score	0.85	0.82	0.77	0.69	0.71	0.51
Accuracy	0.83		0.74		0.63	
AUC	0.76		0.69		0.61	
Residual Network						
	S3/S4		S3/S5		S3/S6	
	0	1	0	1	0	1
Precision	0.75	0.87	0.66	0.81	0.59	0.71
Recall	0.89	0.71	0.87	0.55	0.85	0.38
F1-score	0.82	0.78	0.75	0.65	0.70	0.50
Accuracy	0.80		0.71		0.62	
AUC	0.75		0.66		0.59	

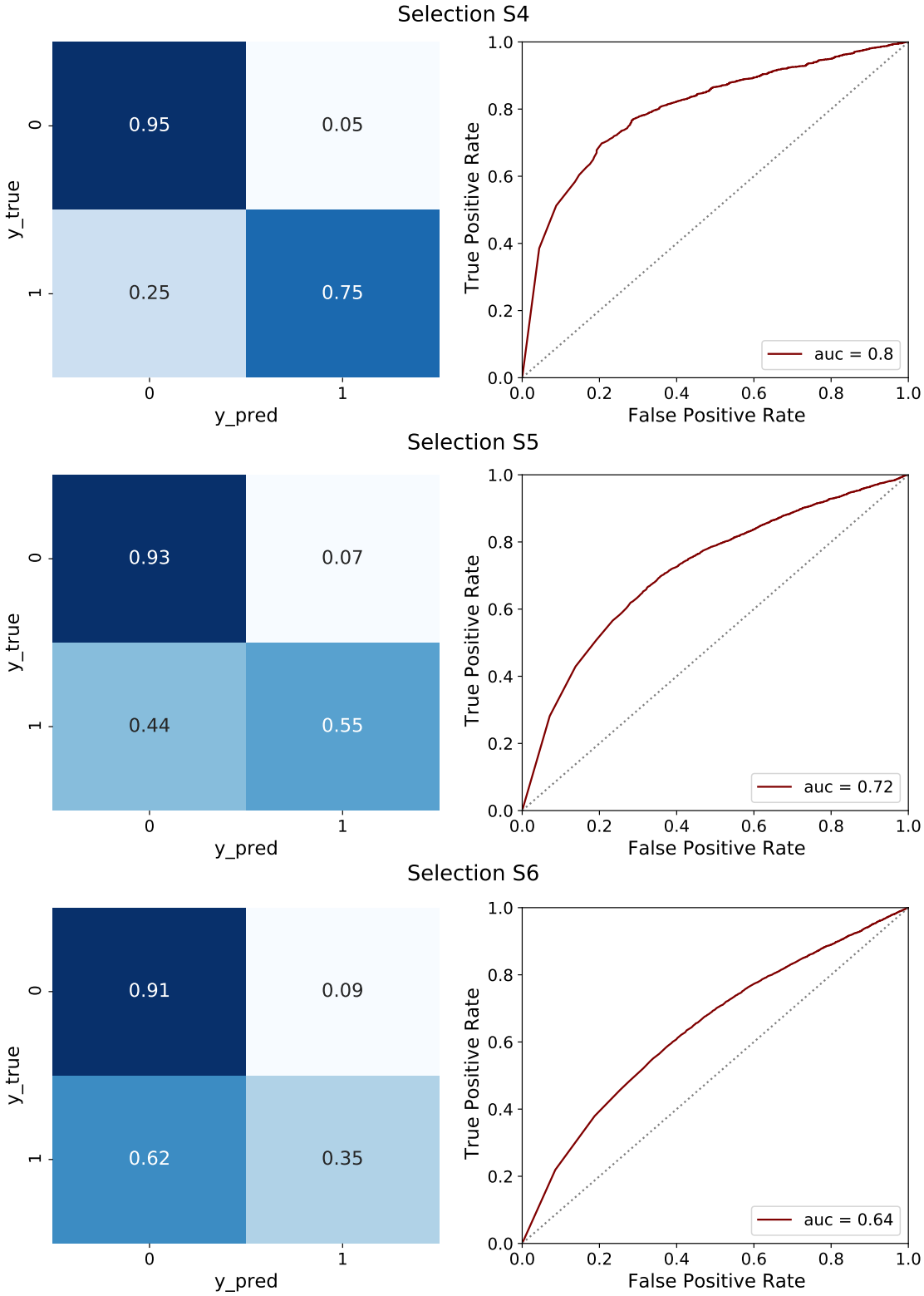


Figure 5.20: Confusion matrices and ROC curves obtained from the test of the VGG-like Network on the selections S4 (first row), S5 (second row) and S6 (third row), after being trained on the selection S3.

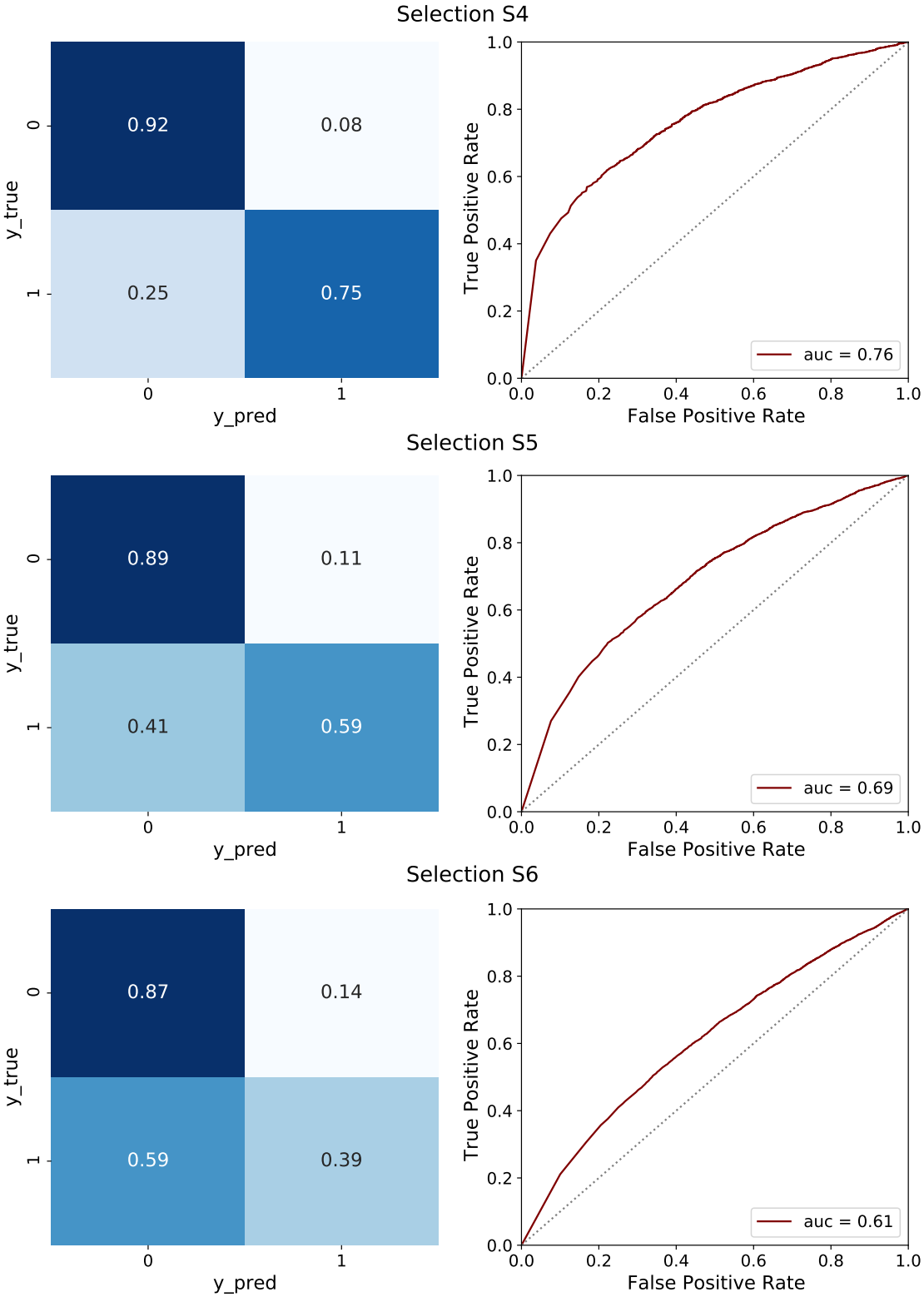


Figure 5.21: Confusion matrices and ROC curves obtained from the test of the Inception Network on the selections S4 (first row), S5 (second row) and S6 (third row), after being trained on the selection S3.

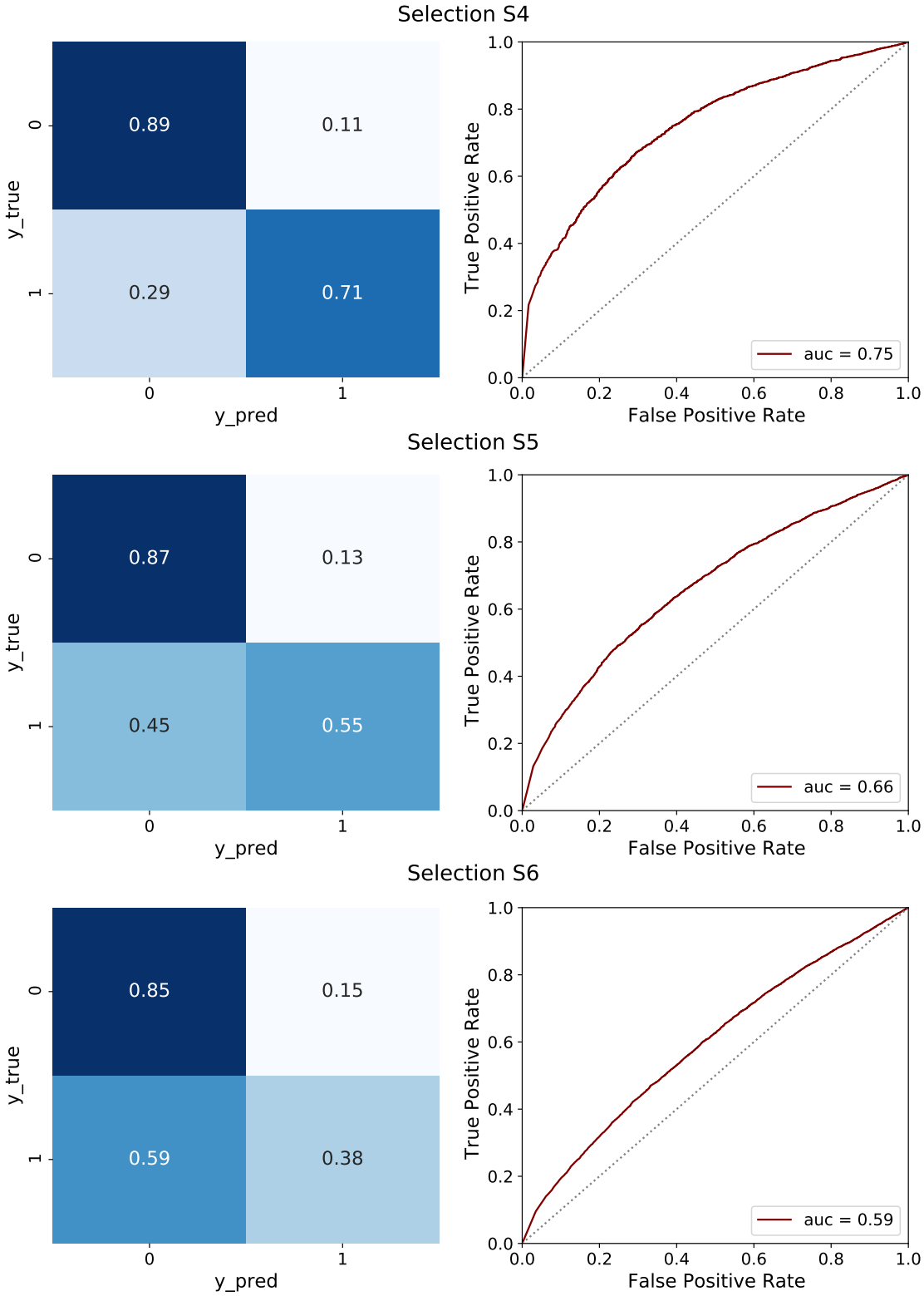


Figure 5.22: Confusion matrices and ROC curves obtained from the test of the Residual Network on the selections S4 (first row), S5 (second row) and S6 (third row), after being trained on the selection S3.

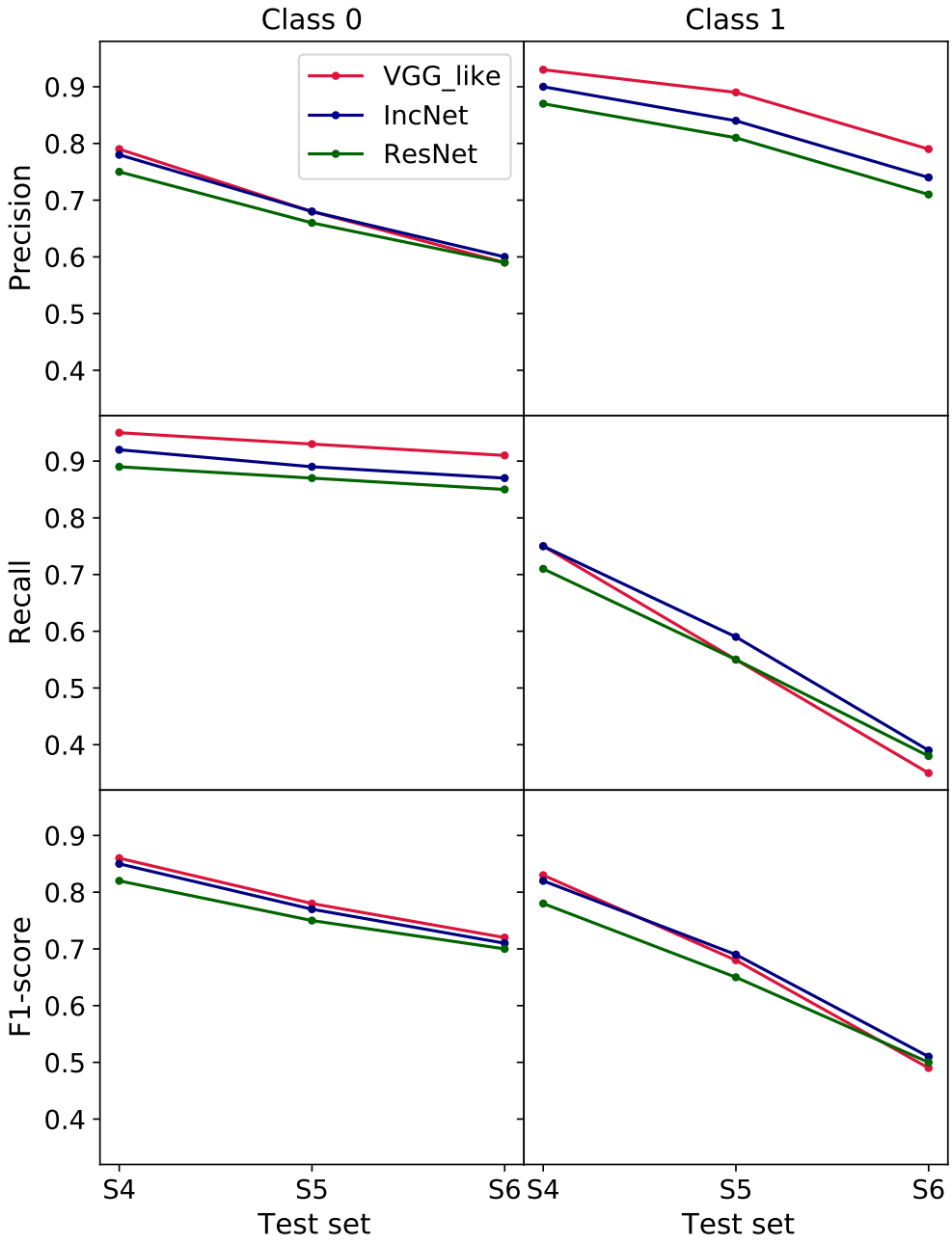


Figure 5.23: Trend of the precision (first row), recall (second row) and F1-score (third row) in the classification of the images that belong to the class of the non lenses (class 0, first column) and to the one of the lenses (class 1, second column) in the selections S4, S5, S6.

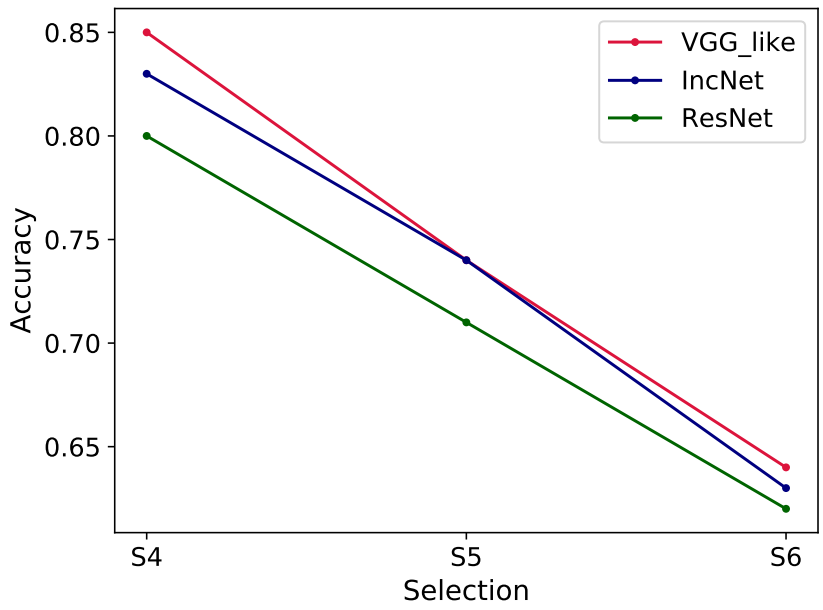


Figure 5.24: Trend of the classification accuracy of the models trained on S3 and tested on the selections S4, S5, S6.

that of the models trained on S6, as it can be seen by comparing this value to those in Table 5.3.
The deterioration of the performance on the wider selections is expected, since the networks primarily recognize the images that are similar to those in the dataset they have been trained on, and most of the images in the training set of S3 show clear lensing features, while the test sets progressively include a greater fraction of borderline objects. We can analyze in greater detail the results of the classification with respect to the two classes from Figure 5.23, that shows the trends of the precision, recall and F1-score: in these cases there are several differences with the metrics in Table 5.3.
In particular:

1. The precision of the classification in the class of the non lenses deteriorates as the selections with greater portions of data are considered in the test: it reaches the value of ~ 0.6 on S6. On the other hand, the recall is approximately constant at values of ~ 0.9 independently from the selection considered, meaning that the greatest part of the objects belonging to this class in the datasets tested is correctly recovered.

2. In the case of the lenses, the precision of the classification ranges from ~ 0.9 to ~ 0.8 depending on the selection being tested. On the other hand, the completeness of the recovered sample decreases drastically from ~ 0.8 to ~ 0.4 when the selections that contain a greater fraction of borderline objects are examined: this trend suggests that the models are not able to identify a great part of the images that belong to the class of the lenses in the complete dataset.

These trends can naturally be explained considering the impact of the inclusion of the borderline objects in the test sets, also inferable from the confusion matrices in Figures 5.20, 5.21, 5.22. In particular, the training set of the selection S3 mostly includes evident lenses, that are characterized by clear lensing features, and images of non lenses to whom no background sources have been added. When analysing the borderline objects, that represent an increasing fraction of the lenses subset of the test sets, the absence of clear arcs and rings, or more generally the faintness of the lensing features causes that a growing percentage of images is classified as being part of the non lenses class. These results highlight the inability of the networks to recover a considerable amount of lenses that are not similar to those in the selection S3. Moreover, the inability of correctly classifying the lenses, reflects in the performance of the networks on the non lenses class. Even though borderline objects are also included in this class, their features make them more similar to the objects of the non lenses class in the training set, therefore they are correctly classified by the models.
However, we can analyze in greater detail the type of objects that are correctly identified in the different tests. We do this by following a similar procedure as the one described in Section 5.4.1 for studying the trend of the classification with respect to some of the characteristics of the lenses and sources. In this case, since we are considering three different test sets, we also divide the trends that refer to the lenses from those that refer to the non lenses, to improve clarity.
In particular, we have evaluated the trends of the performance of the networks with respect to the following parameters: the Einstein area (Figures 5.25 and 5.26) and the halo mass (Figures 5.27 and 5.28) of the lenses; the magnitude (Figures 5.29 and 5.30) and the amount of multiple images (Figures 5.31 and 5.32) of the sources; the amount of pixels where the source is visible (Figures 5.33 and 5.34).
We can highlight some key points about these Figures:

1. The trends of the precision, recall and F1-score with respect to the parameters within each of the selections are similar to those identified and described in the previous Section. In fact, the distribution of the characteristics analyzed is similar in the different selections. The only exceptions are the parameters `n_pix_source` and `n_source_im`, since they are used to discriminate between the objects gradually included in the subsets considered.

2. Some common features in the trend of the recall of the classification with respect to the selections can be highlighted independently from the parameter considered. In particular, in the case of the non lenses the completeness tends to increase from S4 to S6, while the opposite happens in the case of the lenses. This type of trend is expected considering that the percentage of borderline lenses is higher in S6 than in S5 and S4, thus limiting the percentage of objects that we expect to be correctly identified in these selections.

3. The trend of the precision of the classification with respect to the selections varies according to the parameter considered. Both in the case of the lenses and of the non lenses, the parameters that characterize the sources and the simulation of the events, namely `mag_source`, `n_source_im`, `n_pix_source`,

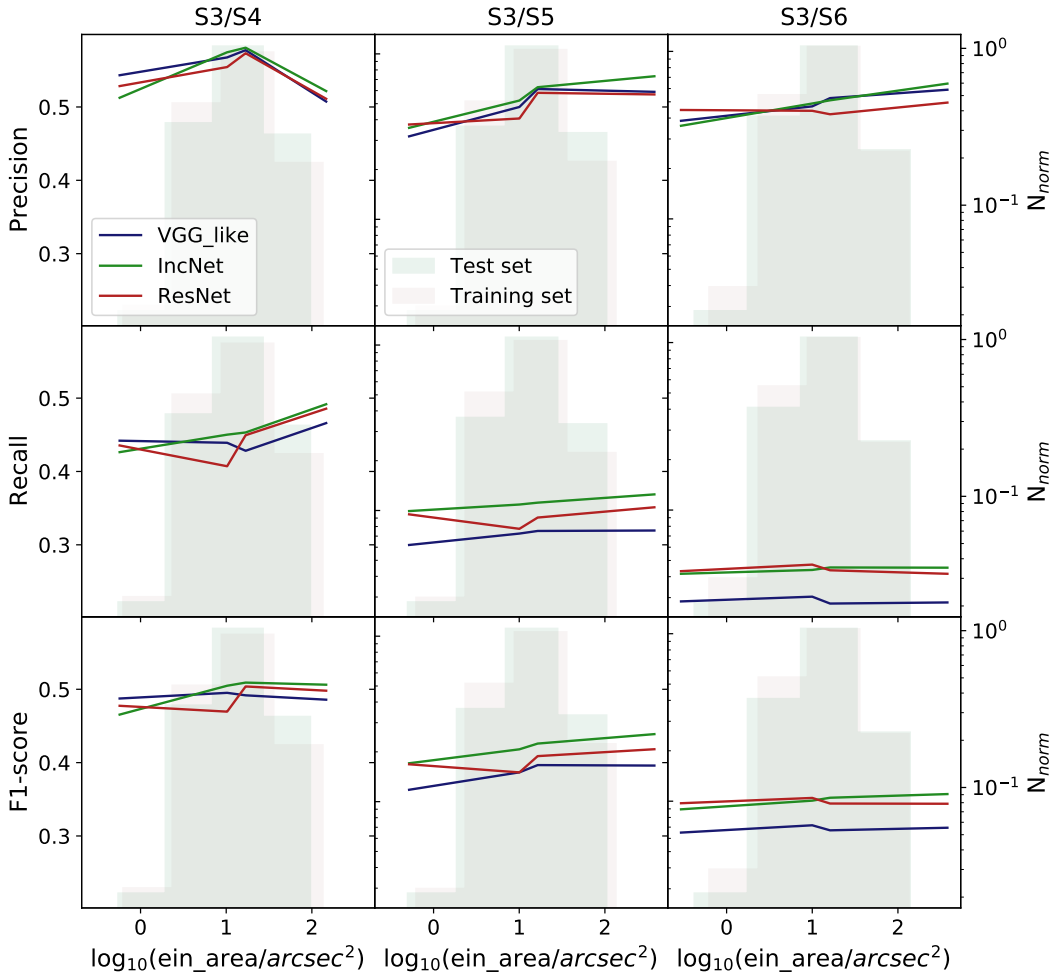


Figure 5.25: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the Einstein area of the lenses in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `ein_area` values in the lenses subsets of the training (purple) and test (green) sets are also displayed in each panel.

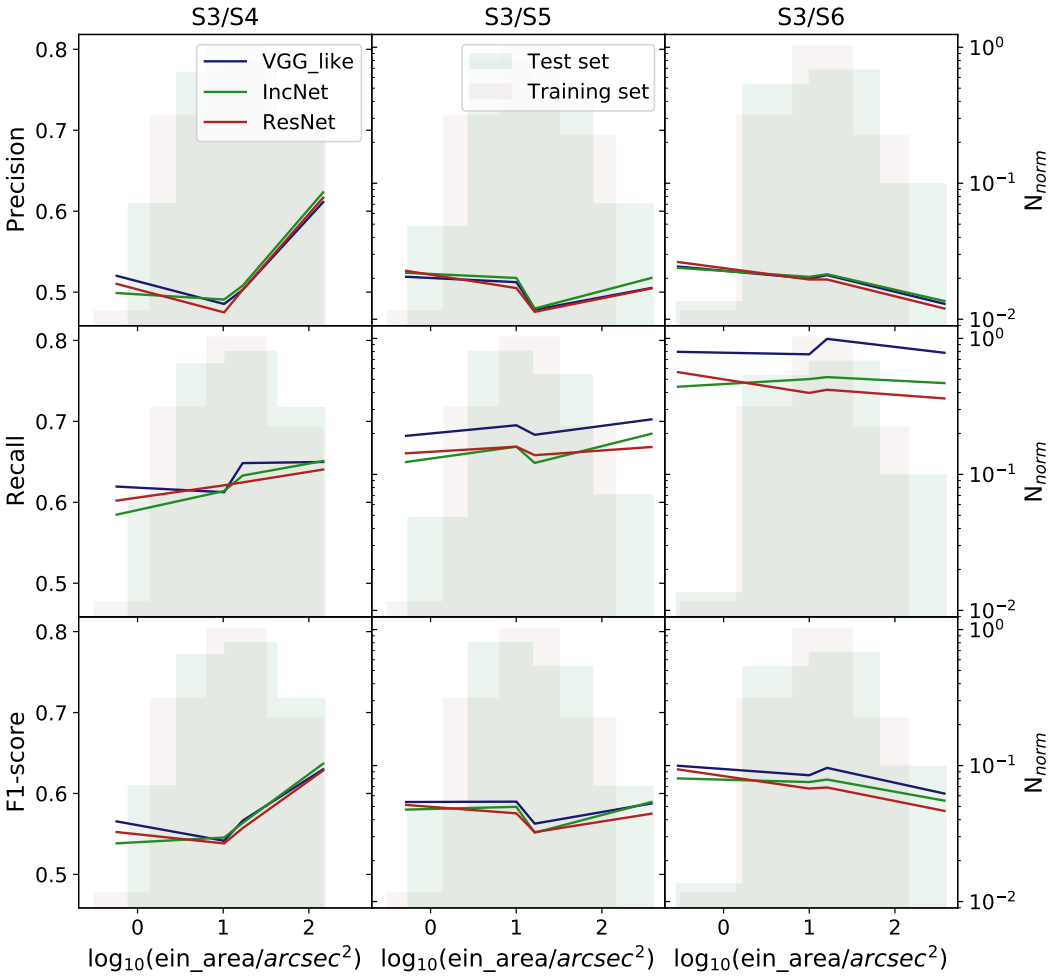


Figure 5.26: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the non lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the Einstein area of the lenses in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `ein_area` values in the non lenses subsets training (purple) and test (green) sets are also displayed in each panel.

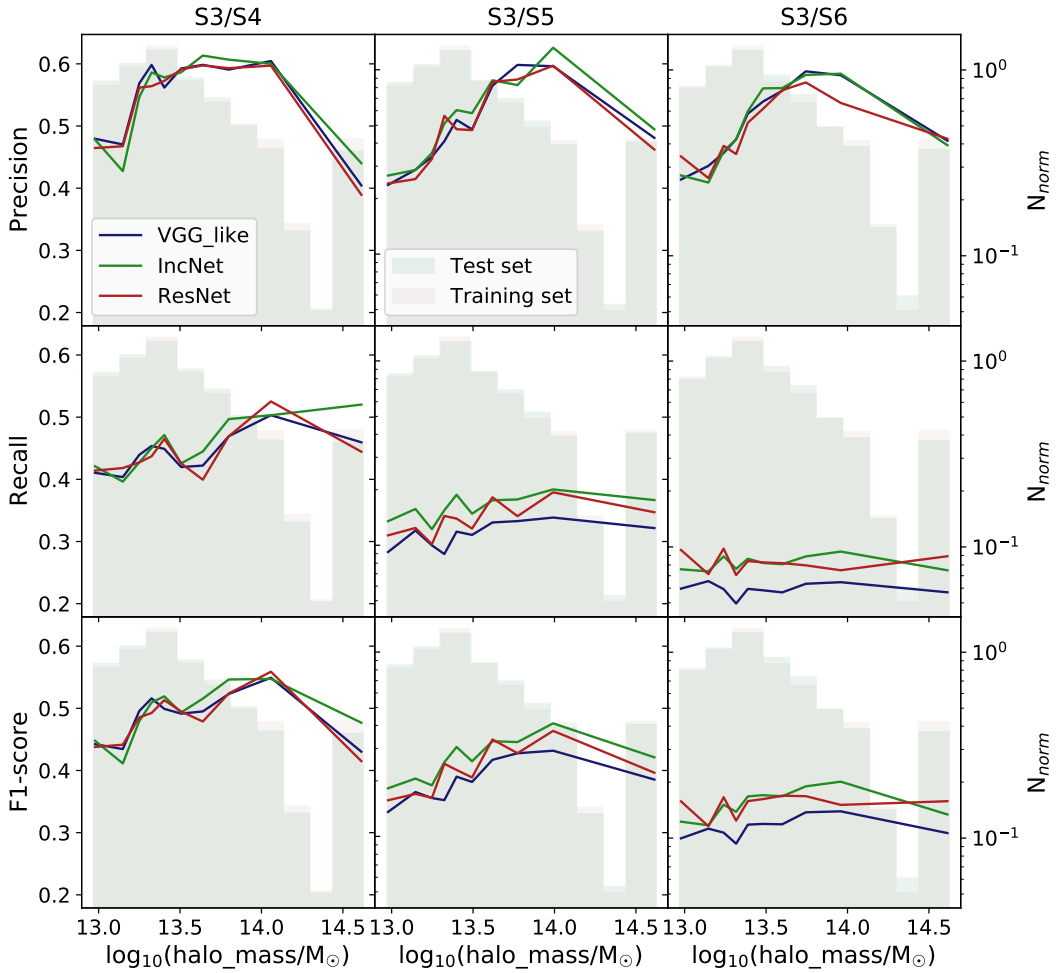


Figure 5.27: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the halo mass of the lenses in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `halo_mass` values in the lenses subsets in the training (purple) and test (green) sets are also displayed in each panel.

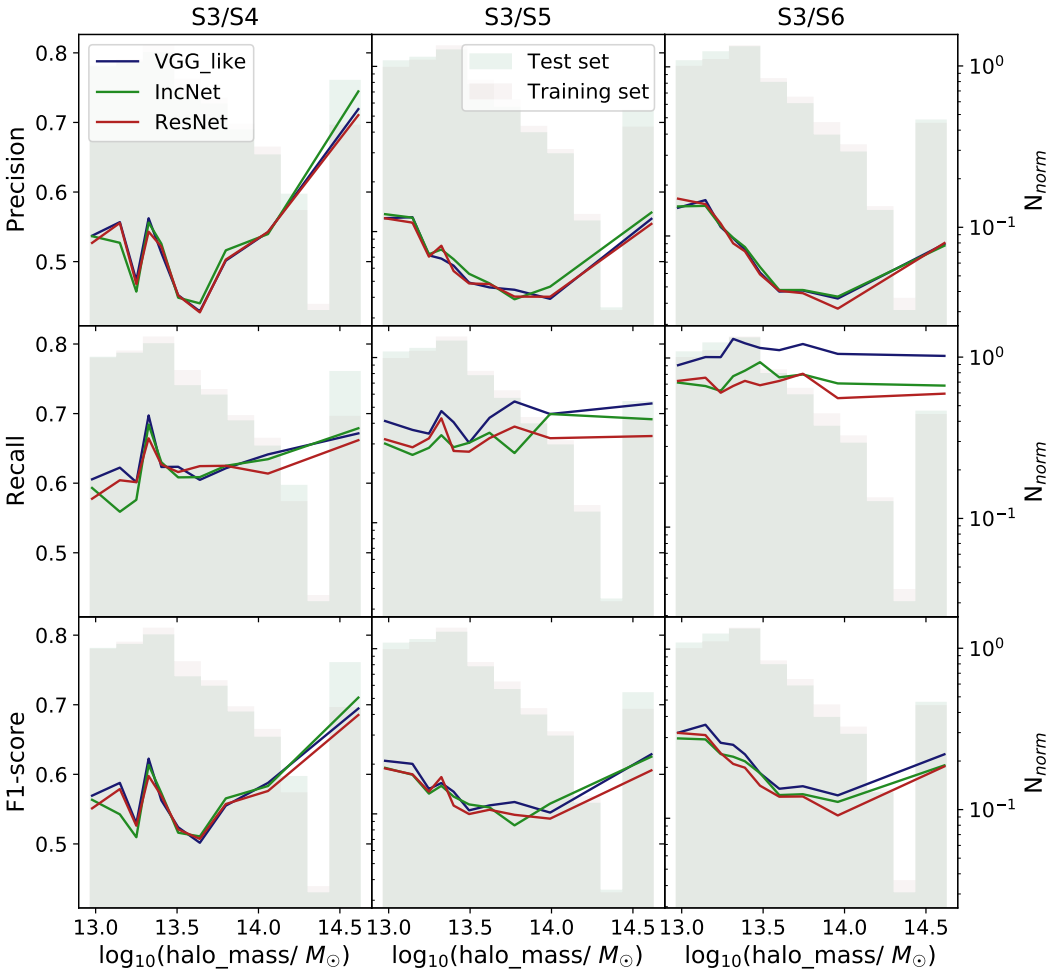


Figure 5.28: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the halo mass of the non lenses in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `halo_mass` values in the non lenses subsets in the training (purple) and test (green) sets are also displayed in each panel.

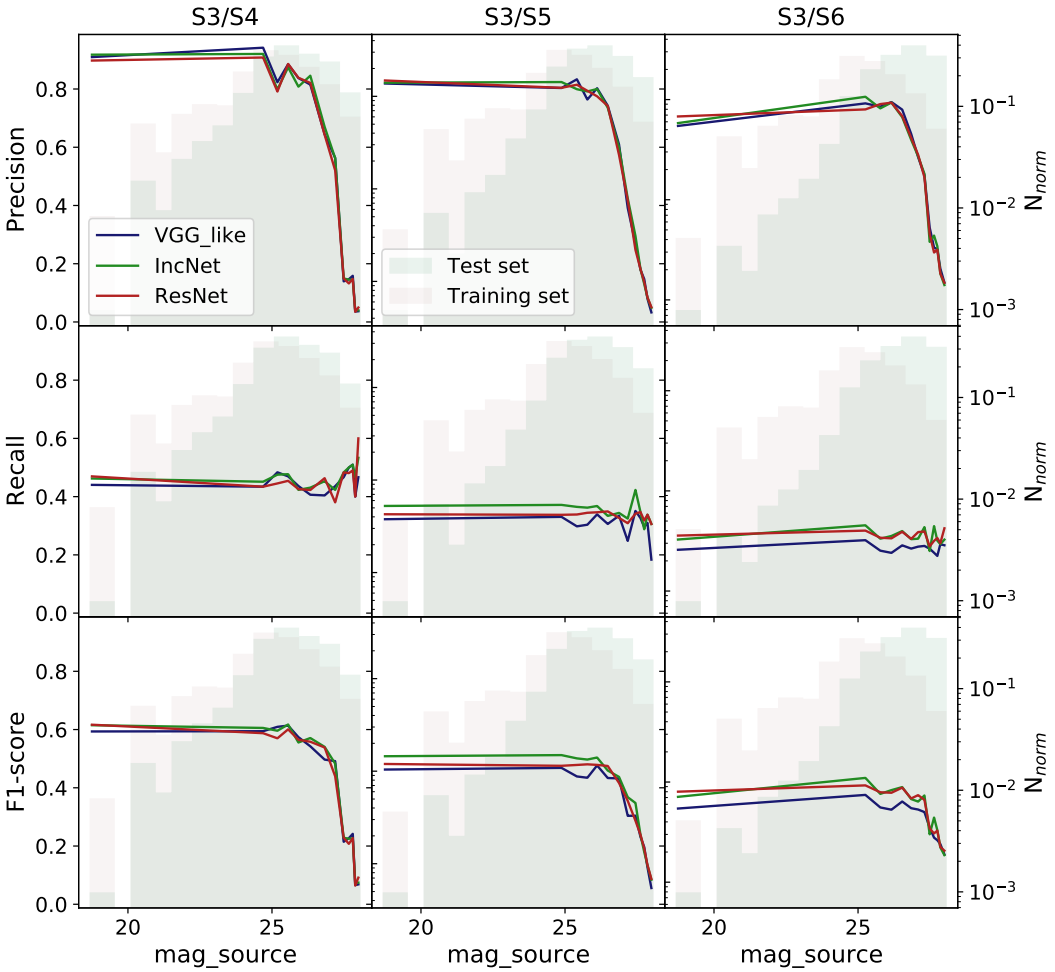


Figure 5.29: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the magnitude of the sources in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `mag_source` values in the subsets of the lenses in the training (purple) and test (green) sets are also displayed in each panel.

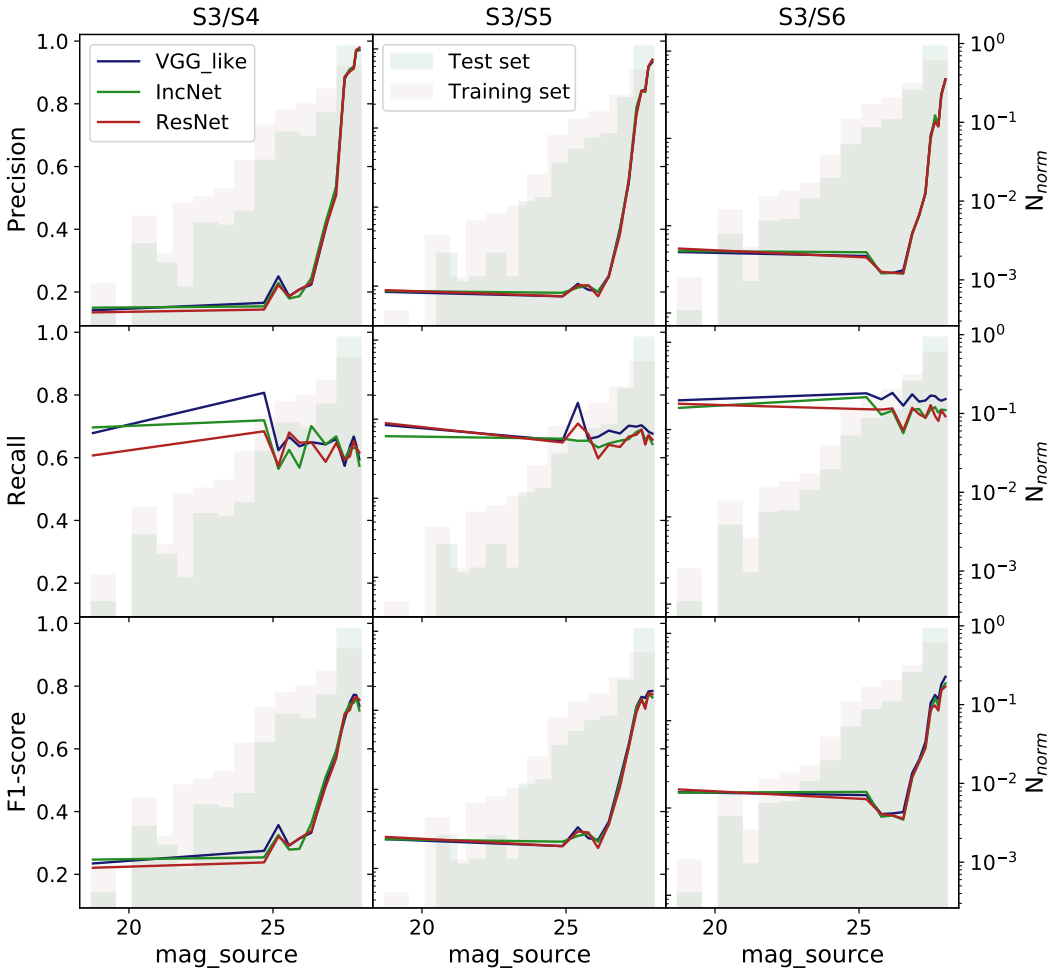


Figure 5.30: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the non lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the magnitude of the sources in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `mag_source` values in the subsets of the non lenses in the training (purple) and test (green) sets are also displayed in each panel.

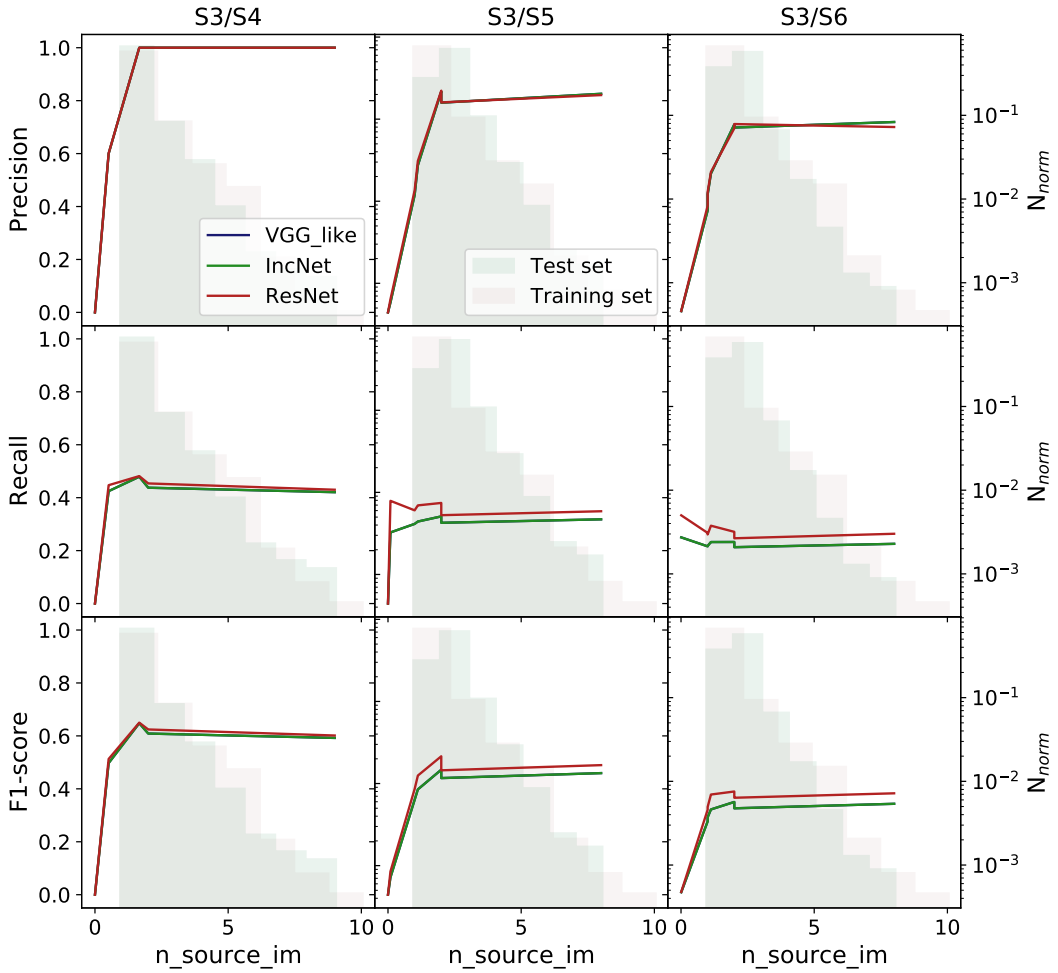


Figure 5.31: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the amount of images of the same source in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `n_source_im` values in the subsets of the lenses in the training (purple) and test (green) sets are also displayed in each panel.

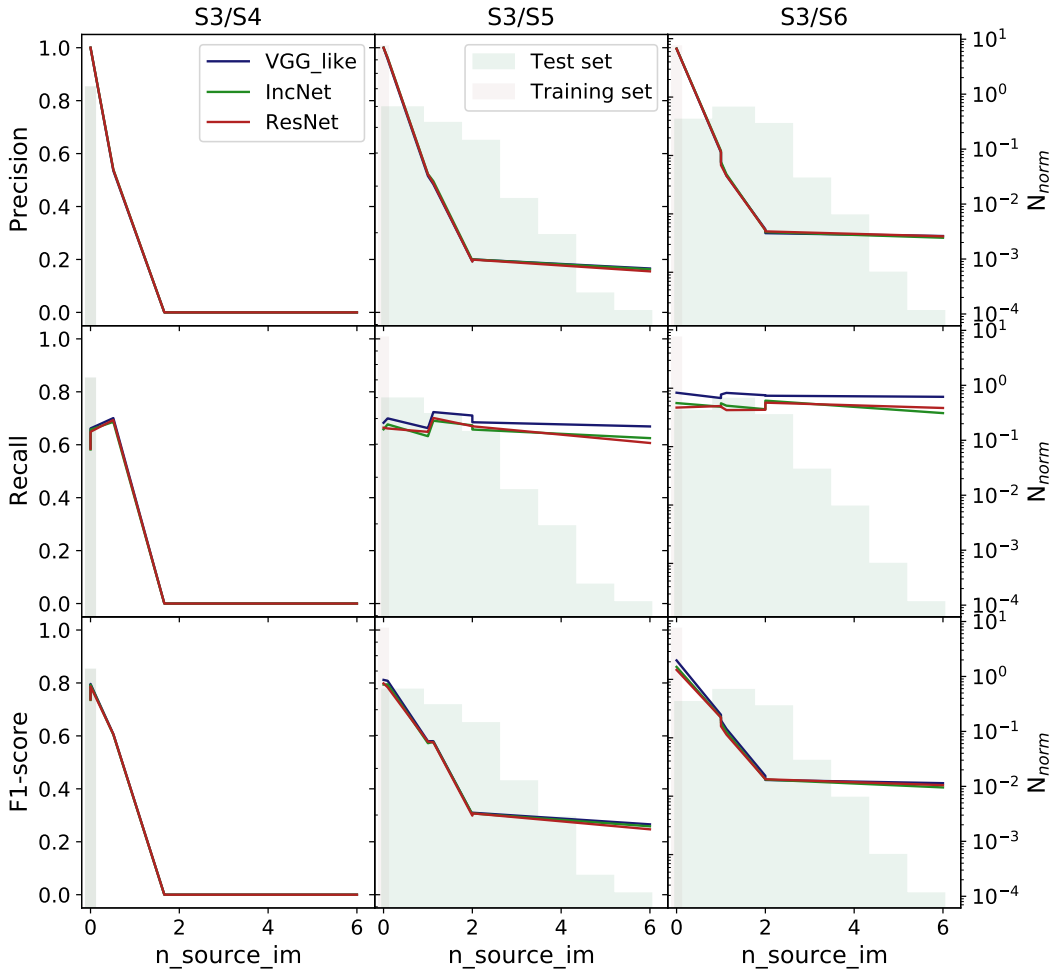


Figure 5.32: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the non lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the amount of images of the same source in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `n_source_im` values in the subsets of the non lenses in the training (purple) and test (green) sets are also displayed in each panel.

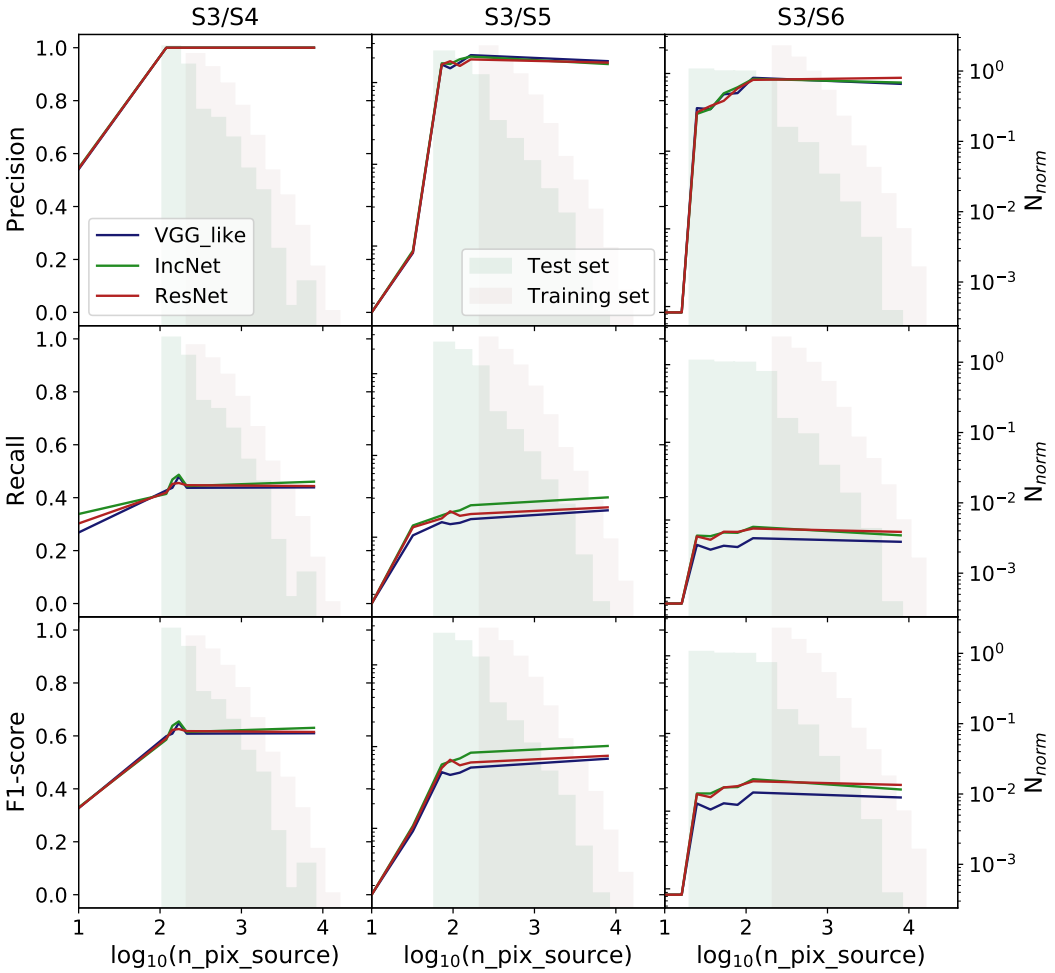


Figure 5.33: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the amount of pixels where the source is visible in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `n_pix_source` values in the subsets of the lenses in the training (purple) and test (green) sets are also displayed in each panel.

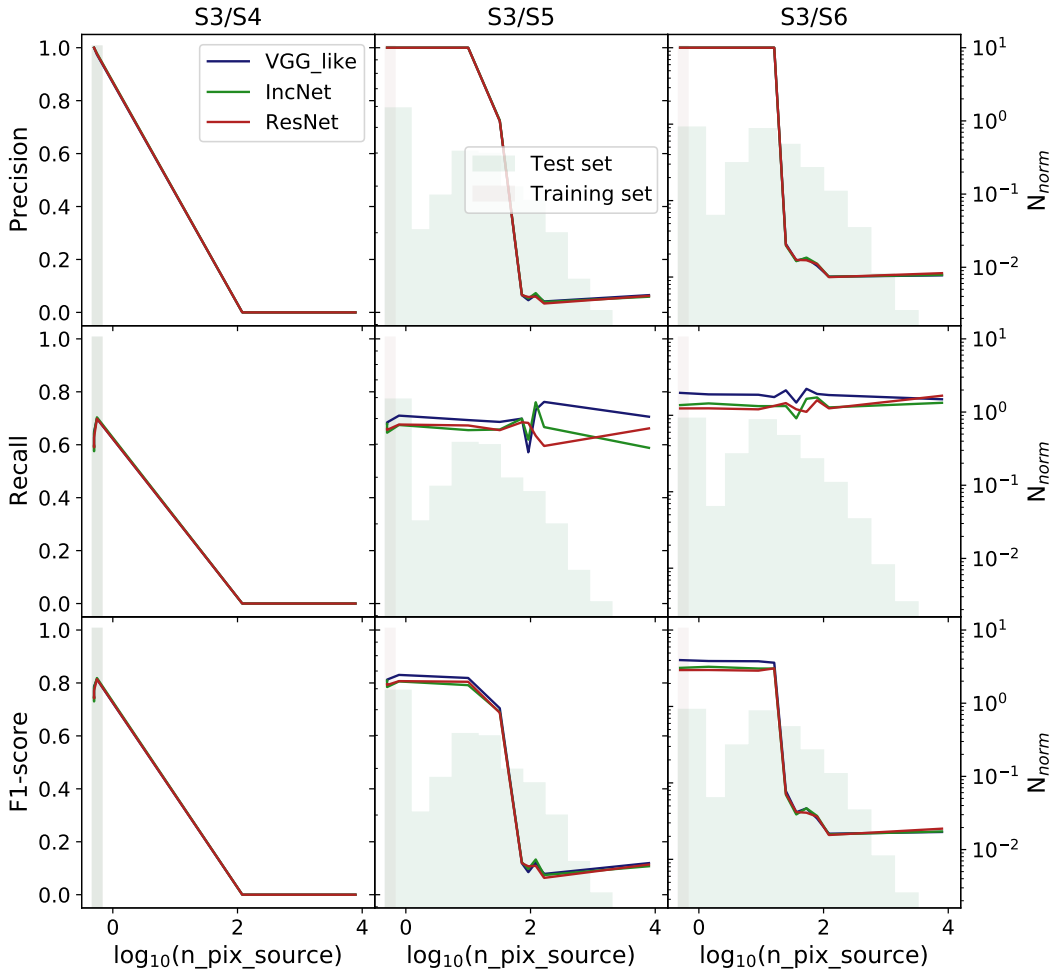


Figure 5.34: Performance of the VGG-like Network (blue line), Inception Network (green line) and Residual Network (red line) on the classification of the non lenses in the selections S4 (first column), S5 (second column) and S6 (third column) as a function of the amount of pixels where the source is visible in these datasets. The first row refers to the precision, the second to the recall and the third to the F1-score. Moreover, the distributions of `n_pix_source` values in the subsets of the lenses in the training (purple) and test (green) sets are also displayed in each panel.

`z_source` have a greater impact on the precision than the characteristics that describe the lenses, i.e. the Einstein area and the halo mass. In particular, the precision in the classification of the lenses tends to slightly increase from S4 to S6, while it tends to decrease in the classification of the non lenses. These trends can be explained as done earlier when the classification of borderline objects was addressed.

4. On average, while the precision obtained in the classification of the lenses is comparable to the one estimated in the tests described in the previous Section, the completeness tends to be lower. This result indicates that in order to retrieve a greater fraction of borderline objects it is better to train the networks on wider selections of data, that include a considerable amount of examples of this type. However, as it was explained in Section 5.3.2, this strategy might cause the wrong identification of some of the obvious lenses, which is a major drawback, since they are often the most useful for scientific purposes.

   All in all, these results indicate that training the networks on specific categories of lenses might be the best strategy to recover a catalog as precise and complete as possible.

5.5 Training on the Euclid VIS Dataset

In addition to training the networks on the Lens Finding Challenge dataset, we also train and test our models on the Euclid VIS dataset. The lenses and non lenses in this dataset are clearly distinguished by the presence or absence of a background source, hence we expect the identification of the lenses to be straightforward in the majority of the cases.
Figure 5.35 shows the confusion matrices and ROC curves obtained from the evaluation of our models on this dataset. In this case, as in the others considered, the performance of the different networks are comparable. In particular, Table 5.6 summarizes the performance of the networks on the two classes: the Residual Network achieves slightly worse results in the identification of the lenses, while the other two networks' performances are nearly indistinguishable.
These results stress the importance of the morphology of the events simulated in the dataset for the performance of the models. In particular, the presence of evident arcs and rings, as the ones that characterize the lenses in this dataset (see, for example, the images in Figure 4.5), is crucial for the networks to correctly identify Strong Lensing events.
We now aim to investigate whether there is a correlation between the performance of our models on this dataset and the main properties of the lens galaxies in the catalogue. We focus on the properties of the lens galaxies since the background sources are only added to the images that are classified as lenses.
We apply the same procedure described in Section 5.4.1 on the images in the test set. In particular, we consider the Einstein area, the halo mass, the magnitude and the redshift of the lenses. The trends of the precision, recall and F-score with respect to these parameters are respectively shown in Figure 5.36, 5.37, 5.38, 5.39.

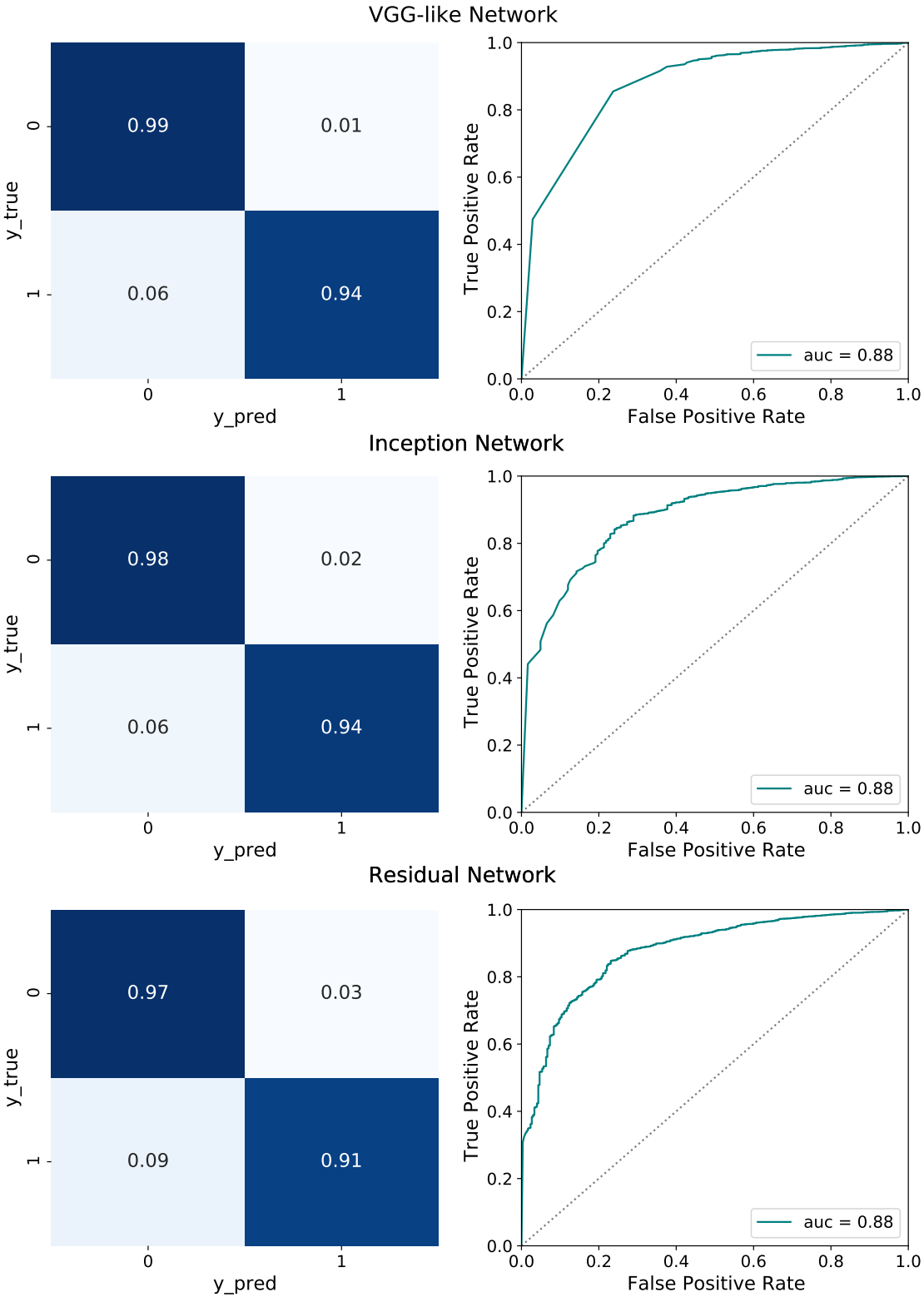


Figure 5.35: Confusion matrices and ROC curves obtained from the evaluation of the performance of the VGG-like Network (first row), Inception Network (second row) and Residual Network (third row) in the classification of the images in the test set of the Euclid VIS challenge.

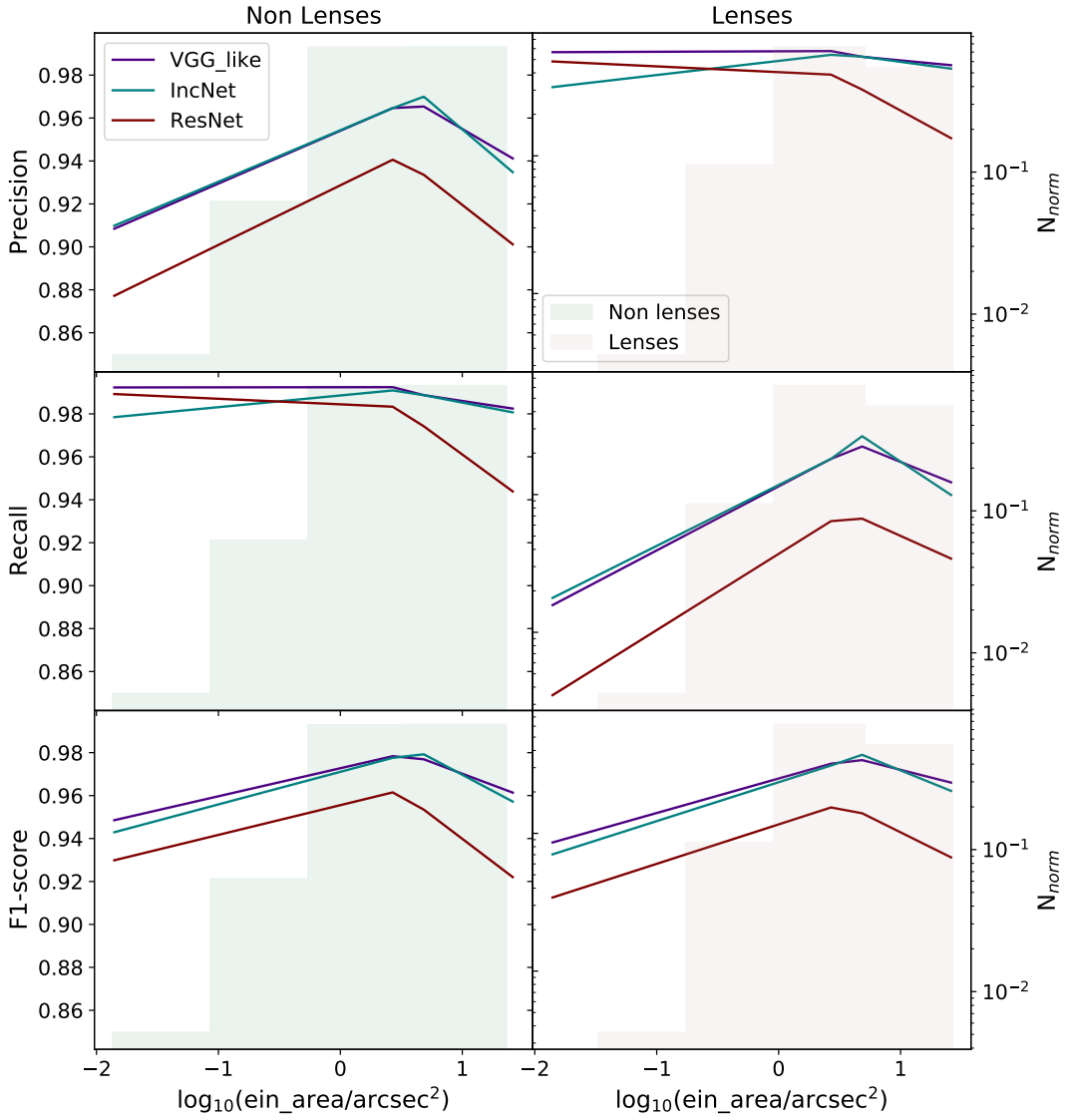


Figure 5.36: Trend of the performance of the VGG-like Network (purple), Inception Network (blue) and Residual Network (red) with respect to the Einstein area of the lenses. The histograms in the background represent the distribution of this parameter for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

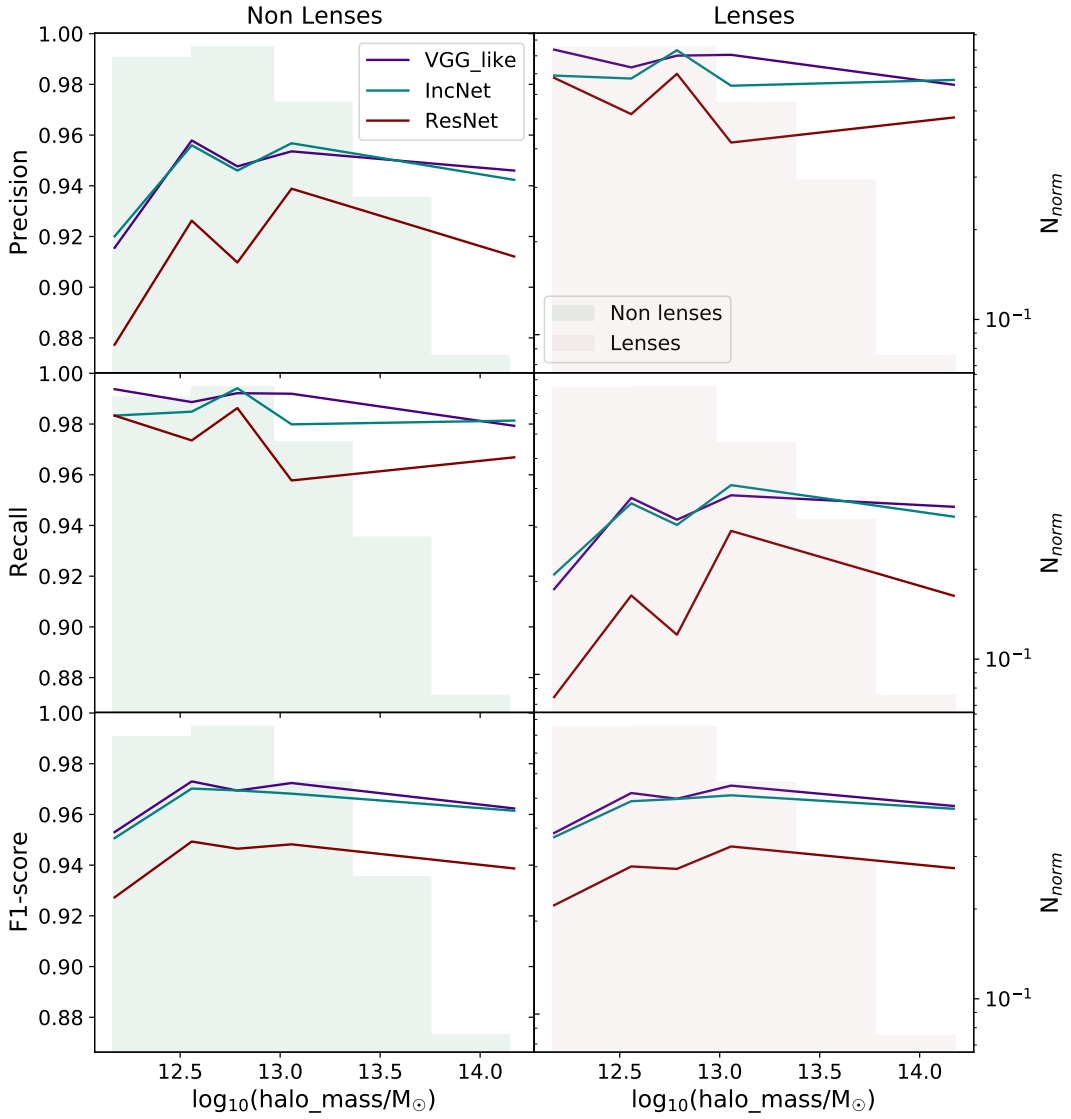


Figure 5.37: Trend of the performance of the VGG-like Network (purple), Inception Network (blue) and Residual Network (red) with respect to the halo mass of the lenses. The histograms in the background represent the distribution of this parameter for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

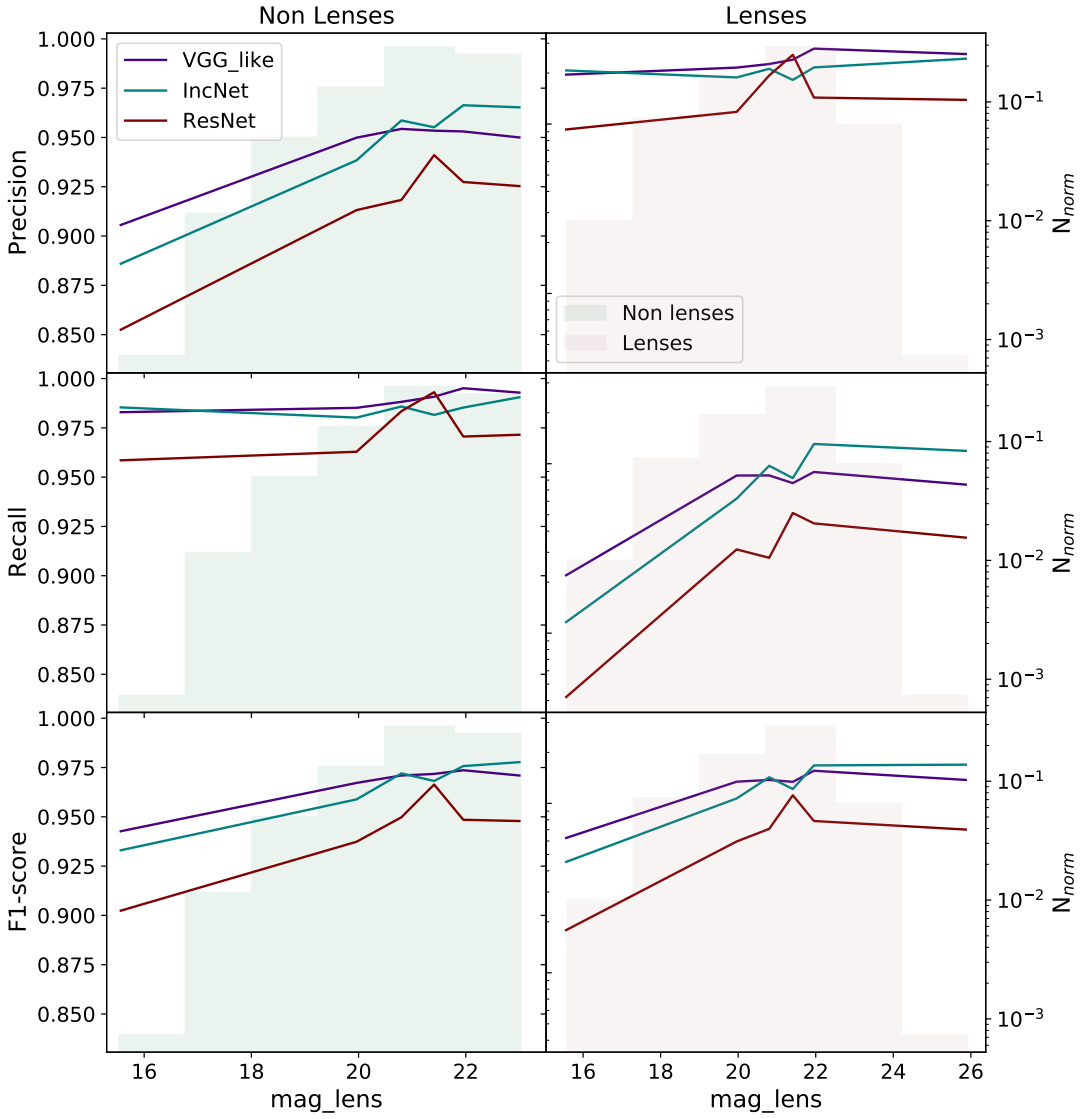


Figure 5.38: Trend of the performance of the VGG-like Network (purple), Inception Network (blue) and Residual Network (red) with respect to the magnitude of the lenses. The histograms in the background represent the distribution of this parameter for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

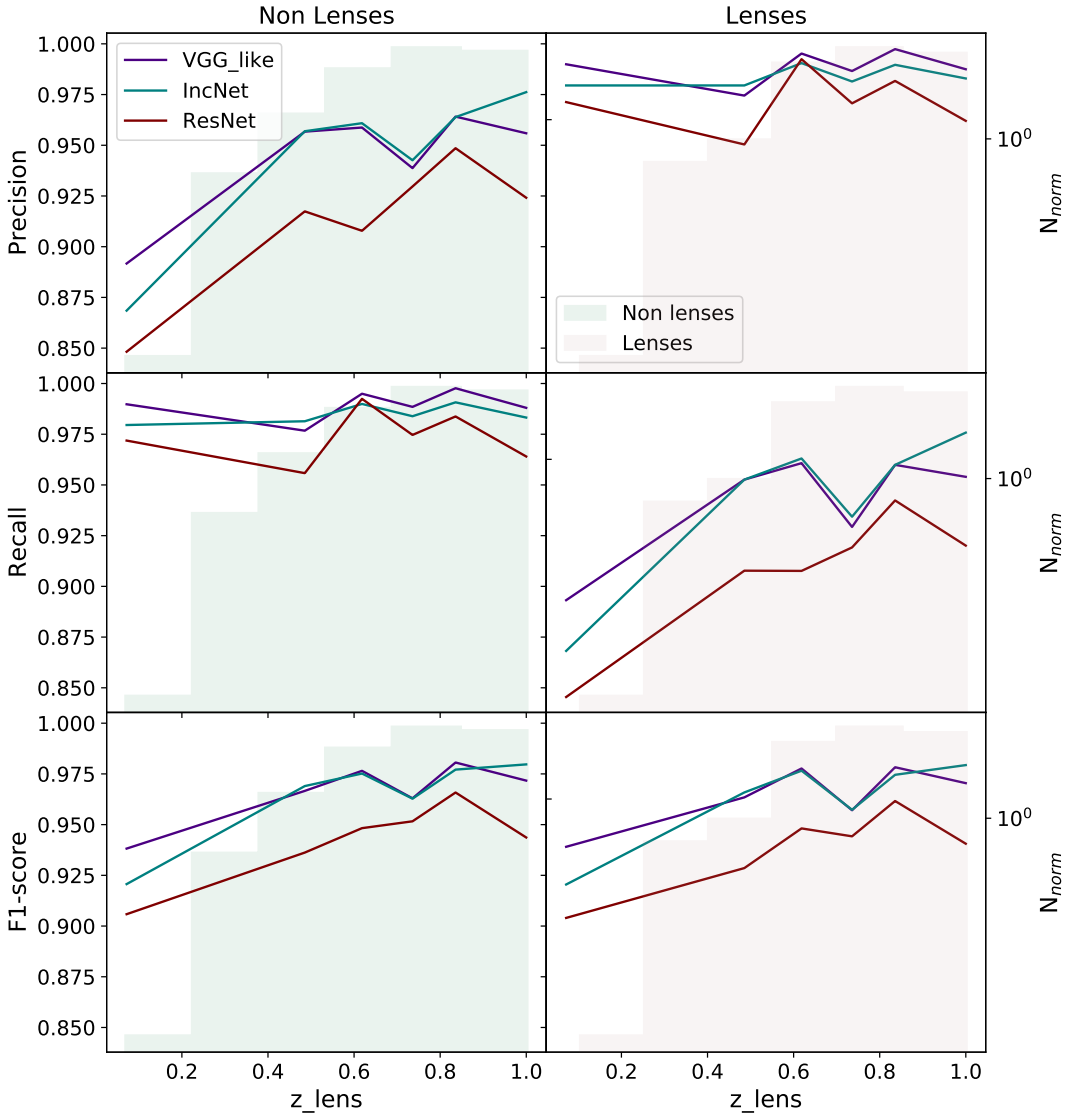


Figure 5.39: Trend of the performance of the VGG-like Network (purple), Inception Network (blue) and Residual Network (red) with respect to the redshift of the lenses. The histograms in the background represent the distribution of this parameter for the elements in the classes of the non lenses (in green) and of the lenses (in purple).

Table 5.6: Summary of the performance of the VGG-like Network, the Inception Network and the Residual Network in the classification of the objects in the Euclid VIS dataset. Class 0 refers to the non lenses, while class 1 refers to the lenses.

	VGG-like Network		**Inception Network**		**Residual Network**	
	0	1	0	1	0	1
Precision	0.94	0.99	0.94	0.98	0.91	0.97
Recall	0.99	0.94	0.98	0.94	0.97	0.91
F1-score	0.97	0.96	0.96	0.96	0.94	0.94
Accuracy	0.97		0.96		0.94	
AUC	0.88		0.88		0.88	

Let us discuss some of the most important results of these tests:

1. The results in terms of precision, recall and F1-score achieved by the three models are in the range of values between 0.85 and 1.0 independently from the parameter considered, as it might have been expected considering the overall performance in Table 5.6.

2. The trends of the performance with respect to the Einstein area (Fig. 5.36) and halo mass (Fig. 5.37) are not similar to those obtained with the tests on the Lens Finding Challenge dataset (see Figures 5.14 and 5.15).

   In the case of the halo mass, it should be considered that the lens masses in this dataset are on average smaller than the ones in the Lens Finding Challenge dataset. In fact, while in this dataset $M_{lens} \sim 10^{12} - 10^{14} M_{\odot}$ (although the largest fraction of the lenses is in the interval $M_{lens} \sim 10^{12} - 10^{13} M_{\odot}$), in the Lens Finding Challenge dataset $M_{lens} \sim 10^{13} - 10^{14} M_{\odot}$. This difference reflects into different types of events and might partially explain the dissimilarity between the trends.

   More generally, in this case and in the others considered, the trend of the F1-score reproduces the distributions of the parameters considered (shown in the background of the Figures under examination). This means that the networks tend to perform worse on the objects that are under-represented in the dataset.

Chapter 6

Conclusions

In this work we have evaluated the ability of Convolutional Neural Networks to identify Galaxy-Galaxy Strong Lensing events depending on their morphology.

We have done this by implementing three different architectures, the VGG-like Network, the Inception Network and the Residual Network, that are respectively inspired by the works of Simonyan & Zisserman (2015), Szegedy et al. (2016) and Xie et al. (2017). In particular, we have compared the performance of these models by training, validating and testing them on two datasets: the Lens Finding Challenge dataset and the Euclid VIS dataset. They have both been simulated by the Bologna Lens Factory and mimic the data quality expected by the Euclid space mission.

The presence in the Lens Finding Challenge dataset of a large amount of images that are characterized by faint lensing features, but do not display evident arcs or rings, makes this dataset ideal to investigate how the diverse morphology of potential candidates affects the classification ability of our models.

Specifically, we have trained and tested our models on six portions of this dataset, initially considering a selection of the most evident objects and gradually expanding the fraction of images that should be more challenging to classify. The images are included in or excluded from the selections according to a parameter used as an indicator of the observability of the event.

We have thus investigated the impact of training our networks on a dataset that includes different amounts of borderline objects by evaluating the recall, precision and accuracy of the classification. We have estimated how this inclusion affects the overall performance of our models as well as the identification of obvious lenses. On the other hand, we have also tested the ability of our models to detect the presence of faint features in the images albeit being trained on clear lenses.

We have found that the morphological characteristics of the lenses included in the training set influences in a crucial way the ability of the CNNs to identify the lenses in a separate test set, whether they show clear of faint lensing features. As it may be expected, the inclusion of a large fraction of borderline images deteriorates the performance of our models, since they are intrinsically more challenging to classify. Moreover, we have found that it impacts the ability of our models to identify the most evident lenses, since they are under-represented in the training set.

These results prove that the identification of lenses with different morphology might require specific trainings, focused on the type of lenses of interest for a certain

purpose. Alternatively, the classification of the lenses might be tackled as a multi-class classification problem, distinguishing the evident and probable lenses from the possible and clear non lenses. In this last case, however, the distinction between obvious and borderline objects should be further investigated and quantified.
Furthermore, we have examined the dependence of the performance of the networks in terms of precision and completeness on some of the physical properties of the simulated lenses and sources. The characterization of the lenses that have been correctly detected is of particular relevance to define the type of lenses that we might expect to detect and to miss in upcoming imaging surveys' data.
In this regard, we have found that the properties of the sources, such as the redshift and magnitude, influence the classification ability of our models more than those of the lenses, such as the halo mass and the Einstein area. Intuitive trends with the amount of multiple images of the background source and the observability of the lensing events were also found.
At this point, we have focused on the Euclid VIS dataset. The greatest majority of the lenses in this dataset is characterized by the presence of evident arc-shaped and ring-shaped features, that might be exploited by the networks for the correct identification of the events. In this case we have investigated the possible correlation between the performance of our models and the properties of the lenses, since the sources are only simulated in the images classified as lenses.
Our results show that the vast majority of the lenses is correctly identified by the three networks, although the Residual Network performs slightly worse than the other two. Moreover, our models generally have more difficulties in the detection of the objects that are under-represented in the dataset, rather than being influenced by the physical properties of the lenses.
Overall, we can also highlight that the architecture of our models does not have a great impact on the completeness and precision of the retrieved catalogues of lenses. In fact, the CNNs we have tested perform similarly on the same selections of images. Because of the faster training and the easier implementation, however, the VGG-like Network might be considered the best architecture among those tested for tackling the problem of identifying GGSL events in large datasets.

6.1 Future perspectives

In the future, several improvements and extensions of this work might be implemented.
First of all, it would be important to assess the flexibility of our models to identify lensing galaxies located in galaxy clusters. In fact, the simulated events in the datasets considered are mainly produced by field galaxies, while the majority of the actual observed events are due to cluster members. Because of the importance of the realism of the images in the training set with respect to those that the networks are actually expected to detect, this might be an issue.
In particular, the characteristics of the GGSL events observed in the field are not exactly the same as they would be if the lens galaxies were members of a cluster. Specifically, due to the large shear and background mass surface density in dense environments, the critical lines of the galaxy-scale lenses are expected to be larger

than those of field lenses with the same mass and light distribution (Meneghetti, M., 2018).
It would also be interesting to re-train the networks on a training set of cluster lenses to evaluate the improvement of the performance with respect to the one of our models. Appendix B presents a simple procedure, which is currently under implementation, for the simulation of GGSL events in galaxy clusters.
Secondly, it should be noted that we used single band images to evaluate the performance of our models: this means that the correct identification of the lenses in the dataset was solely based on their morphological features, namely on the presence of arc-shaped and ring-shaped images of the background sources. However, this is not the only characteristic of the images that might be exploited by the CNNs in the identification of potential candidates.
In particular, several works have highlighted the importance of colour information (see e.g. Metcalf et al. 2019), that might be of special relevance in the classification of the lenses whose morphology is not clear enough. Re-training the networks on the same selections of the Lens Finding Challenge dataset considered in this work, but including the images simulated in the IR bands would prove how significant such information is in the identification of borderline lenses.
Finally, the application of our models to real images would provide useful insights on the validity of the results obtained from the tests we have conducted on the simulated datasets. In particular, this type of test would prove whether or not our models are able to identify, other than the evident lenses, the most exotic ones. Even though the morphology of the lenses in the Lens Finding Challenge dataset is diverse, it might be that the simulations are not complicated enough to reproduce the wide variety of possible real configurations.
While the observations of the Euclid telescope are not yet available, a preliminary test might be carried out on the catalogues of GGSL events produced by the analysis of the images gathered by the HST. Of course, these images would have to be degraded to the image quality of the Euclid telescope, to grant homogeneity with images in the training set.

Appendix A

Implementation of the models

The models studied in this work were implemented, trained and tested using `Keras`[1] (Chollet, 2015) 2.4.3 with a `TensorFlow`[2] (Abadi et al., 2016) 2.2.0 backend on a NVIDIA Titan Xp Graphics Processing Unit (GPU).
`Keras` is an open-source library written in Python as part of the research work of the Open-ended Neuro-Electronic Intelligent Robot Operating System (ONEIROS) project. It essentially focuses on the development of intuitive methods and classes that enable fast experimentation with Neural Networks and, more generally, Machine Learning models.
`TensorFlow` is an open-source platform for Machine Learning developed and maintained by the Google Brain team of Google. It is used for several Machine Learning applications and is particularly advantageous because of its ability to run on multiple Central Processing Units (CPUs) and GPUs, speeding up the execution of a large variety of tasks, such as the training of deep models.
GPUs are not generally faster than CPUs, but they perform better in the execution of computations that can be done in parallel because they have a larger number of cores. Since many of the computations performed in the training procedure of Convolutional Neural Networks can be easily parallelized, GPUs are largely employed to reduce the execution time of this computationally expensive process.
Section A.1 illustrates the main characteristics and features of the architecture of our models, while Section A.2 contains a brief description of the training procedure's setup and specifics.

A.1 Architectures

We implement three CNN architectures, inspired by the work of Simonyan & Zisserman (2015), Szegedy et al. (2016) and Xie et al. (2017), respectively named VGG_like, IncNet and ResNet.
The specific network architectures applied to the images are the result of the trials of several possible configurations, in which the amount of layers and of kernels has been varied, in order to identify the arrangement that best performs on our classification problem.

[1] https://keras.io/
[2] https://www.tensorflow.org/

A.1.1 VGG-like Network

Our implementation of the VGG Network is composed of ten convolutional layers and five max pooling layers alternating. At the end of each convolutional-pooling block we perform the batch normalization[3] of the output of the block. Moreover, the output layer is preceded by two fully connected layers, that have been alternated with three dropout layers. The amount of parameters that constitute the architecture amounts to about 2 millions.
Figure A.1 shows a diagram of the network, that summarizes its most relevant properties. The different blocks are color-coded depending on their function: we distinguish the input layer (green), the convolutional layers (blue), the max pooling layers (red) and the fully connected layers (orange). The numbers indicated in the convolutional blocks represent the dimension (D) of the filters and the amount of filters (F) used in the layer, in the format D×D, F. The numbers highlighted in the max pooling layer, on the other hand, represent the pooling region (R) and the strides (S) of the pooling operation, in the format R×R, /S.
The numbers in the square brackets instead indicate the dimension and amount of the feature maps obtained as output of the layers in the format [D×D×F], if the input has dimension [200×200×1], as is the case of the VIS images of the Lens Finding Challenge Dataset. Moreover, it is always specified whether batch normalization or dropout were performed. Finally, the output of the output layer is the probability of the processed image to be part of each one of the two possible classes considered in our problem.

A.1.2 Inception Network

The building block of our implementation of the Inception Network is the module represented in Figure 3.9b. Before being fed to the inception modules, the images are initially processed through two convolutional and max pooling layers. The network is composed of seven modules, the fifth of whom is connected to an additional classifier. Dropout is performed before both the output layers, while batch normalization is performed on the output of each max pooling layer. The total amount of parameters that compose the model is circa 2 millions.
The chart in Figure A.2 shows the architecture of this network, following the same notation clarified in the previous Section.

A.1.3 Residual Network

The fundamental block of our Residual Network is the residual block displayed in Figure 3.11b, with cardinality equal to eight. In particular, the input is initially analyzed by two convolutional and pooling layers and then processed by four residual blocks alternated with two max pooling layers. Before being passed to the output layer, dropout is performed on the resulting feature maps. Moreover, batch normalization is performed after every max pooling layer.

[3]Batch normalization is a technique employed to accelerate and stabilize the training of deep networks, that consists in the re-normalization of the layer inputs (Ioffe & Szegedy, 2015).

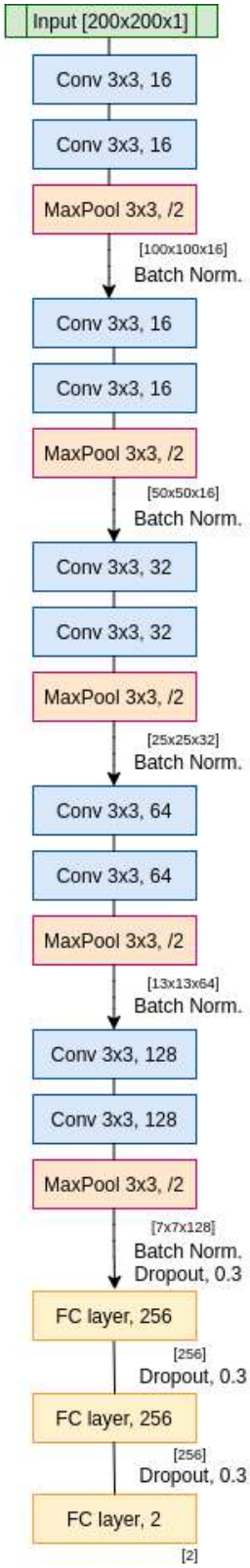


Figure A.1: Architecture of the VGG-like Network. This model is characterized by ten convolutional layers and five max pooling layers alternating, followed by three fully connected layers.

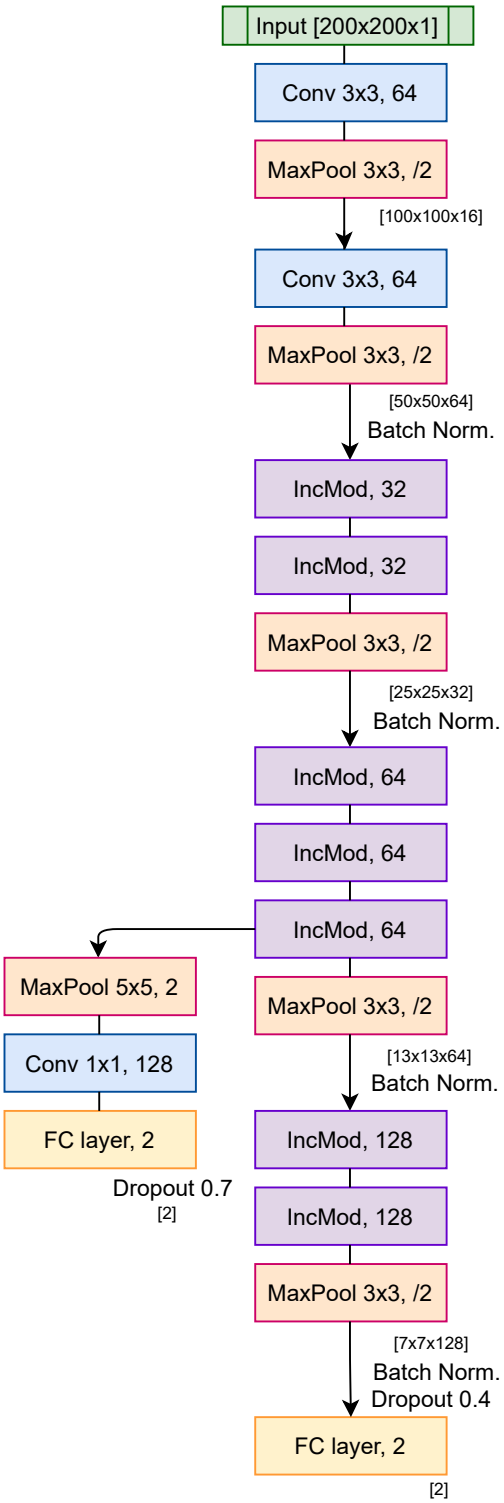


Figure A.2: Architecture of the Inception Network. The two initial convolutional-pooling blocks are followed by seven inception modules alternating with three max pooling layers. The output of the fifth inception module is both passed to the following inception module and to an additional classifier. The inception modules (purple boxes) are structured as the one in Figure 3.9b.

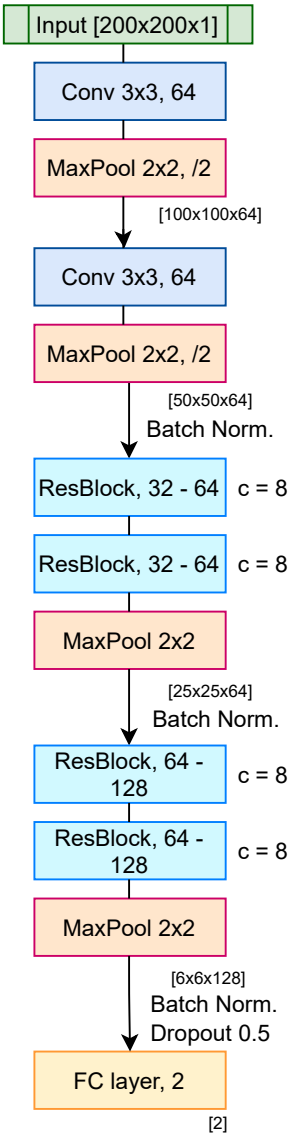


Figure A.3: Architecture of the Residual Network. This model is composed of two convolutional-pooling blocks and four residual blocks with cardinality equal to eight alternating with two max pooling layers. The residual block's (in sky blue) structure is the one reproduced in Figure 3.11b.

Table A.1: Summary of the main characteristics of the network architectures implemented.

	VGG-like Network	Inception Network	Residual Network
Parameters	1.9 M	2 M	1.1 M
Dropout	Yes	Yes	Yes
Additional classifier	No	Yes	No
Type of layers	Convolutional, Max Pooling, Fully Connected	Inception modules, Convolutional, Max Pooling, Fully Connected	Residual blocks, Convolutional, Max Pooling, Fully Connected

The parameters in the model amount to 1 million, therefore they are significantly fewer than in the other architectures. However, this configuration outperformed the other possible ones that were tested.
The diagram in Figure A.3 shows the structure of this network with the same notation described in Section A.1.1.

Table A.1 summarizes some of the characteristics of the architectures described.

A.2 Training

We implement a training algorithm that employs the `model.fit`[4] method of `Keras`. This method automatically carries out the procedure described in Section 3.3 to train the model on the training set and simultaneously validate it on the independent validation set.
We conduct eighteen trainings in total on the selections of data described in Section 5.3.1, since we train each architecture on each selection of data. We use the Adam optimizer with initial learning rate fixed to 10^{-4} and we choose to employ the binary cross-entropy to estimate the loss at the end of each epoch. We run the trainings for 100 epochs.
Moreover we conduct three additional trainings on the Euclid VIS dataset, one for each architecture, with the same characteristics.
We monitor the trainings by using two functions available in `Keras`. In particular, if the loss function estimated on the validation set does not improve for several consecutive epochs, the learning rate will be reduced because of the `ReduceLROnPlateau`[5] callback, or stopped before being completed by the `EarlyStopping`[6] callback. Both these preventative measures are taken to avoid overfitting.

[4] https://keras.io/api/models/model_training_apis/#fit-method
[5] https://keras.io/api/callbacks/reduce_lr_on_plateau/
[6] https://keras.io/api/callbacks/early_stopping/

Appendix B

Simulation of GGSL events in galaxy clusters

In this Section we describe a simple procedure, that is currently under implementation, to simulate GGSL events in galaxy clusters.
We simulate GGSL events by injecting background sources in real observations of the six galaxy clusters studied in the Frontier Fields[1] (FF) program. We position these background objects close to the secondary caustics produced by the cluster members. In this way, they act as strong lenses, distorting and magnifying the images of the sources.
Note that throughout the simulations we adopt a Λ-CDM cosmology with $\Omega_\Lambda = 0.7$, $\Omega_M = 0.3$, $H_0 = 70\, km\, s^{-1} Mpc^{-1}$.

B.1 The Frontier Fields clusters

The Frontier Fields is a program that combines the capabilities of the HST and the Spitzer Space Telescope[2] to obtain deep observations of six galaxy clusters and their lensed galaxies (Lotz et al., 2017). In particular, these massive objects were chosen because they act as high-magnifying lenses, allowing the observation of very faint and distant galaxies.
The campaign involves the observation of the six galaxy clusters listed in Table B.1 and six parallel fields. See Lotz et al. (2017) for a detailed summary of the main characteristics of the clusters and of the blank fields.
The observations gathered with this program have been made publicly available at https://archive.stsci.edu/pub/hlsp/frontier/.
In particular, we use the observations taken with the instrument Wide Field Camera 3[3] (WFC3) of the HST in the F814W filter of the UVIS channel and in the F105W, F140W and F160W filters of the IR channel. We choose to focus on these specific filters because they operate in a similar wavelength range as the Euclid Telescope's filters used to simulate the images of the Lens Finding Challenge (see Table B.2 and 4.1, respectively).

[1] https://outerspace.stsci.edu/display/HPR/HST+Frontier+Fields
[2] http://www.spitzer.caltech.edu/
[3] https://wfc3.gsfc.nasa.gov/

Table B.1: List of the galaxy clusters observed in the Frontier Fields, of their redshifts and of the coordinates of their centers. Data from https://outerspace.stsci.edu/display/HPR/HST+Frontier+Fields+Survey.

Cluster name	Nickname	Redshift	Cluster Coordinates (J2000)	
			RA	DEC
ABELL 370	A370	0.375	02:39:52.9	-01:34:36.5
ABELL S1063	A1063	0.348	22:48:44.4	-44:31:48.5
ABELL 2744	A2744	0.308	00:14:21.2	-30:23:50.1
MACSJ0416.1-2403	M0416	0.396	04:16:08.9	-24:04:28.7
MACSJ0717.5+3745	M0717	0.545	07:17:34.0	+37:44:49.0
MACSJ1149.5+2223	M1149	0.543	11:49:36.3	+22:23:58.1

Table B.2: List of the WFC3 filters, in which the observations of the FFs' clusters used in this work were taken.

Channel	Filter	λ (nm)
UVIS	F814W	835.3
IR	F105W	1045
	F140W	1400
	F160W	1545

The observation of several strong lensing events in these clusters has made a precise strong lensing modeling of them possible. In particular, we use the deflection angle fields obtained from this modeling (Caminha et al., 2017; Bergamini et al., 2019) to simulate the deflection of the light emitted by a background source located behind the cluster. These maps take into account the effects of the cluster mass and of the individual galaxies within it and allow us to reproduce the phenomenology of the multiple images and distortions typical of GGSL events in galaxy clusters.
Figure B.1 shows, as an example, the deflection angle maps of the galaxy cluster M0416.

B.2 Simulation procedure

The simulations are carried out by employing several of the methods and functions implemented in the Python library `pyLensLib`[4] .
Firstly, we build the *deflector*, namely the object that, given the deflection angle maps, the redshifts of the lens and of the source computes the deflection of the light of the source in a given cosmological context. We build the deflector by using the

[4]More information on this library is in the lecture notes at https://github.com/maxmen/LensingLectures

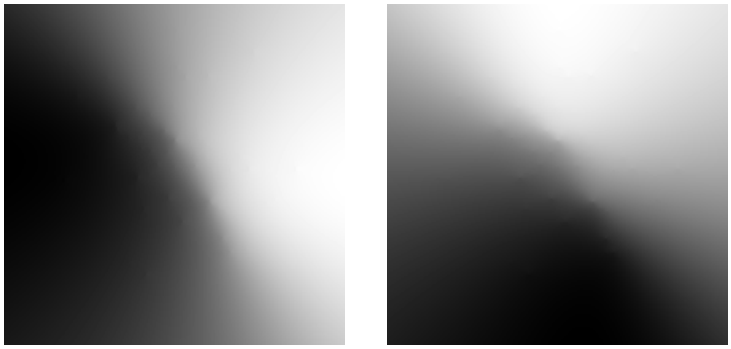


Figure B.1: Deflection angle maps of the galaxy cluster M0416 for a source plane at $z_s = 1$. The maps are used to compute the deflection of the light rays emitted by a background source, as in Equation 2.10.

method `deflector` of `pyLensLib`.
Using the deflector we can compute the tangential and radial critical lines of the cluster (primary critical line) and of the galaxies within it (secondary critical lines). The points of the critical lines can also be traced to the corresponding points of the caustics on the source plane.
We simulate the GGSL event by adding a background source in the vicinity of the caustic produced by one of the cluster members. In particular, we model the source by means of a surface brightness profile described by a Sérsic model (Sérsic, 1963), that we build using the method `sersic` of `pyLensLib`.
Given the deflector, the position of the source with respect to the cluster's center and the flux of the source, the method `sersic` produces the image of the lensed source. While we have already created the deflector, we need to define the source position and its flux.

We randomly choose the position of the source in a region around one of the secondary caustics of the cluster, calculated with the deflector. In particular, we use the Python package `shapely`[5] to select the area around the caustic that has the same radius as the effective radius of the galaxy we are simulating.
Figure B.2 displays, as an example, the position of a fictitious source extracted within the buffer around one of the radial caustics of the galaxy cluster M0416.
Since the radius of the buffer corresponds to the galaxy's effective radius, we are confident that the source will at least cross the caustic and the resulting image will be characterized by some type of strong lensing features. At the same time, following this procedure for the selection of the source's position allows the implementation of a wide range of possible distortions and multiple image configurations, that would not be reproducible by exclusively locating it within the caustic.

At this point, we evaluate the flux of the source in each of the observational bands of the simulation, by associating a SED to the source. We do this by choosing one of

[5] https://shapely.readthedocs.io/en/stable/manual.html

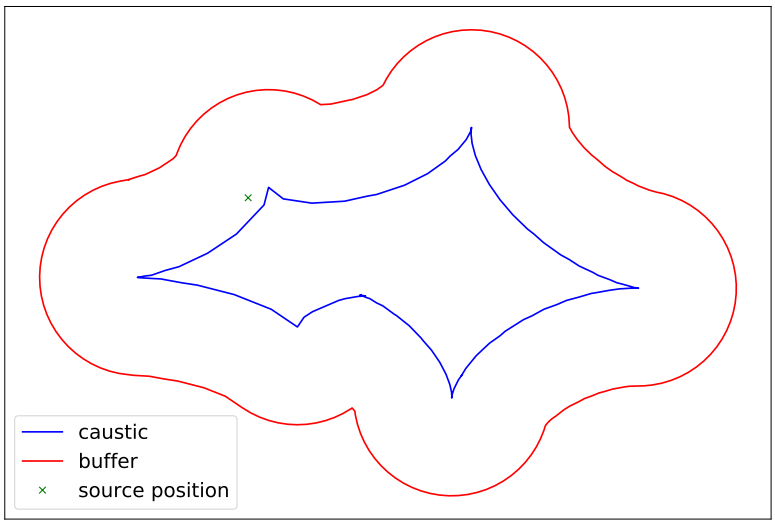


Figure B.2: This image displays an example of the choice of the source position. radial caustics (blue line), the buffer calculated around them (red line) and the position randomly chosen within the buffer.

the available templates of the software BPZ[6]. This software is used to estimate the photometric redshift of distant objects by cross-correlating the observed magnitudes of the sources at different wavelengths with the templates of various types of galaxies. The models at our disposal are suitable to describe Elliptical, Spiral and Starburst galaxies.
The SEDs in BPZ are rest-frame, so once we have selected one, we shift it at the redshift of our source. For this task, we employ the package `Barak`[7], that also allows us to extract the values of the flux of the source in arbitrary bands: in this case, we are interested in the bands listed in Section B.1. We proceed by defining the magnitude of the source in a certain band to normalize the SED. We can now integrate the SED in the desired bands to obtain the flux that corresponds to the given magnitude.
Given the flux, we can calculate the correspondent counts on the detector of the instrument used for the observation. For the conversion of the flux into counts, we use the zero-point of the instrument in a certain band, that can be found in the header of the image of the cluster.

The counts of the unlensed source in a certain filter and its position are finally used as input of the `sersic` function, along with the effective radius, the position angle, the Sérsic index and the axis ratio. Thanks to the deflector the light of the source is projected onto the lens plane and the GGSL event is observable. Two possible output configurations of the operations described are shown in Figure B.3.
We proceed by doing a cutout of the simulated image and of the HST image of $10'' \times 10''$ separately, centering it at the center of the critical line in both the images.

[6] https://www.stsci.edu/~dcoe/BPZ/
[7] http://nhmc.github.io/Barak/

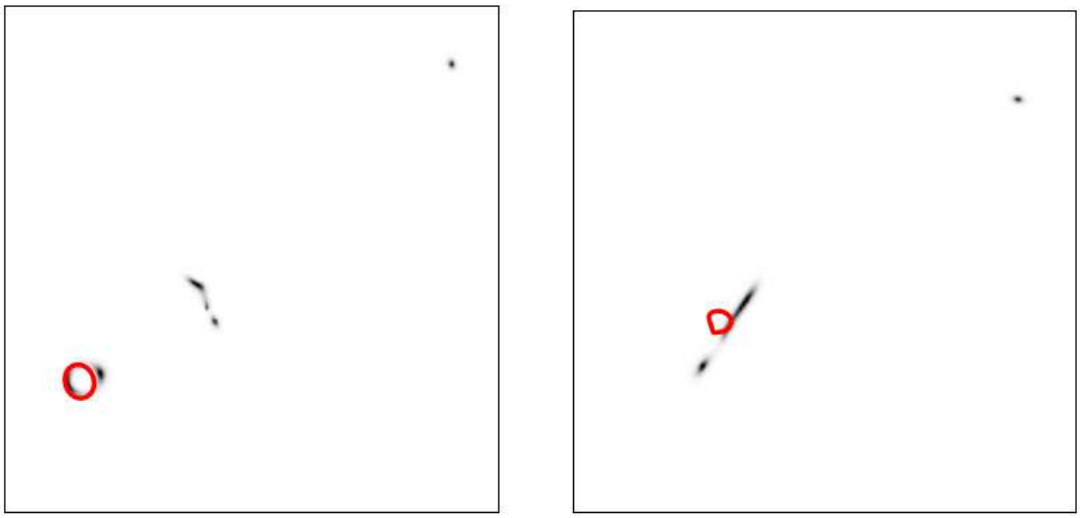


Figure B.3: Zoom-in of two GGSL events produced by the simulation procedure described. In the image on the left a ring is produced, while in the image on the right a tangential arc, as the critical line chosen is tangential. The other lensing features observed are due to the main caustic of the cluster. The red lines represents the critical lines of the galaxy lenses.

For this task, we use the function `Cutout2D` of `Astropy`[8]. At this point, we re-grid the simulated image so that its World Coordinate System (WCS) is the same as the one of the astronomical image. This operation involves the variation of pixel resolution, orientation and coordinate system and can be conducted using the package `reproject`[9], that takes the headers of the simulated image and of the cluster as input.
After this step, we perform the convolution of the simulated image with the PSF model characteristic of the HST filter in which the observation of the cluster was taken. We do this employing the function `psf_convolve` of the method `observation` in `pyLensLib`.
Finally, we sum the simulation of the lensing features and the real observation of the cluster and we obtain the image of the GGSL event at a particular wavelength. To simulate the event in other bands, we do not iterate the procedure from the beginning. In fact, we re-normalize the reprojected image to the counts of the source in another filter, as calculated from the SED, and we perform the convolution of the resulting image with the PSF model appropriate for the current filter. This method is based on the assumption the the Sérsic profile chosen to describe the source does not change as a function of λ.
In this way we obtain the image of the GGSL event we are simulating at different wavelengths. Figure B.4 shows a few examples of GGSL events simulated by following the simulation process discussed.

[8] https://www.astropy.org/
[9] https://pypi.org/project/reproject/

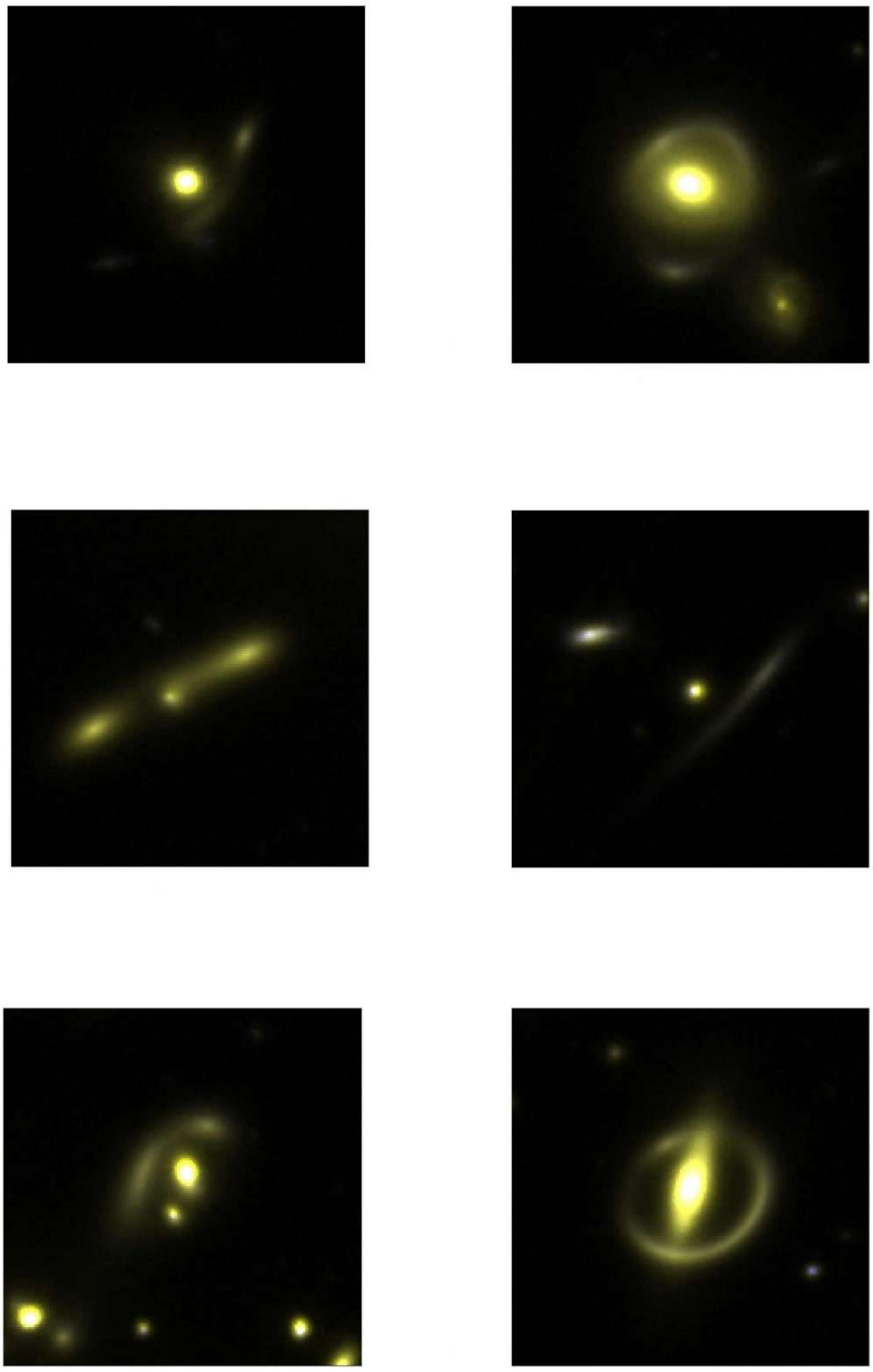


Figure B.4: This Figure shows six possible configuration of GGSL events obtained by following the procedure described in this Section. In these images, the F814W, F105W, and F140W are being used for the blue, green, and red channels, respectively.

Bibliography

Abadi M., et al., 2016, *TensorFlow: A System for Large-Scale Machine Learning*, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp 265–283

Angora G., et al., 2020, *The search for galaxy cluster members with deep learning of panchromatic HST imaging and extensive spectroscopy*, Astronomy and Astrophysics, 643, A177

Baron D., 2019, *Machine Learning in Astronomy: a practical overview*, arXiv e-prints, p. arXiv:1904.07248

Bartelmann M., Schneider P., 2001, *Weak gravitational lensing*, Physics Reports, 340, 291–472

Beckwith S. V. W., et al., 2006, *The Hubble Ultra Deep Field*, The Astronomical Journal, 132, 1729 –1755

Bengio Y., 2009, *Learning Deep Architectures for AI*, Foundation and Trends in Machine Learning, 2, 1–127

Bennett, C. L. et al. 1996, *4-Year COBE DMR Cosmic Microwave Background Observations: Maps and Basic Results*, The Astrophysical Journal, 464, L1 –L4.

Benson, B. A. et al. 2004, *Measurements of Sunyaev-Zel'dovich Effect Scaling Relations for Clusters of Galaxies*, The Astrophysical Journal, 617, 829 –846.

Bergamini P., et al., 2019, *Enhanced cluster lensing models with measured galaxy kinematics*, Astronomy & Astrophysics, 631, A130

Bergamini P., et al., 2020, *A new high-precision strong lensing model of the galaxy cluster MACS J0416.1-2403*, arXiv e-prints, p. arXiv:2010.00027

Bhattacharya, S. et al. 2013, *Dark matter Halo profiles of massive clusters: Theory versus observations*, The Astrophysical Journal, 766, 32.

Bishop C., 2006, *Pattern Recognition and Machine Learning*. Springer-Verlag New York

Boylan-Kolchin M., Springel V., White S. D. M., Jenkins A., Lemson G., 2009, *Resolving cosmic structure formation with the Millennium-II Simulation*, Monthly Notices of the Royal Astronomical Society, 398, 1150 –1164

Burges C. J., 1998, *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery, 2, 121 –167

Caminha G. B., et al., 2017, *Mass distribution in the core of MACS J1206. Robust modeling from an exceptionally large sample of central multiple images*, Astronomy & Astrophysics, 607, A93

Cavaliere, A. & Fusco-Femiano, R. 1976, *X-rays from hot plasma in clusters of galaxies.*, Astronomy and Astrophysics, 49, 137 –144.

Chollet F., 2015, *keras*, https://github.com/fchollet/keras

Coles P., Lucchin F., 2002, *Cosmology: The Origin and Evolution of Cosmic Structure*. John Wiley & Sons Inc.

Collett T. E., 2015, *The Population of Galaxy-Galaxy Strong Lenses in Forthcoming Optical Imaging Surveys*, The Astrophysical Journal, 811, 20

Collett T. E., Auger M. W., 2014, *Cosmological constraints from the double source plane lens SDSSJ0946+1006*, Monthly Notices of the Royal Astronomical Society, 443, 969–976

Davies A., Serjeant S., Bromley J. M., 2019, *Using convolutional neural networks to identify gravitational lenses in astronomical images*, Monthly Notices of the Royal Astronomical Society, 487, 5263 –5271

De Boni, C. et al. 2013, *Hydrodynamical simulations of galaxy clusters in dark energy cosmologies – II. c–M relation*, Monthly Notices of the Royal Astronomical Society, 428, 2921–2938.

Deng L., 2014, *A tutorial survey of architectures, algorithms, and applications for deep learning*, APSIPA Transactions on Signal and Information Processing, 3, e2

Desprez G., Richard J., Jauzac M., Martinez J., Siana B., Clément B., 2018, *Galaxy-galaxy lensing in the outskirts of CLASH clusters: constraints on local shear and testing mass-luminosity scaling relation*, Monthly Notices of the Royal Astronomical Society, 479, 2630–2648

Dressler, A. 1980, *Galaxy morphology in rich clusters: implications for the formation and evolution of galaxies.*, The Astrophysical Journal, 236, 351 –365.

Duchi J., Hazan E., Singer Y., 2011, *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*, Journal of Machine Learning Research, 12, 2121 –2159

Eckert, D. et al. 2019, *Non-thermal pressure support in X-COP galaxy clusters*, Astronomy & Astrophysics, 621, A40.

Fabian, A. C. 1994, *Cooling Flows in Clusters of Galaxies*, Annual Review of Astronomy and Astrophysics, 32, 277 –318.

Gavazzi R., Marshall P. J., Treu T., Sonnenfeld A., 2014, *RINGFINDER: Automated Detection of Galaxy-scale Gravitational Lenses in Ground-based Multi-filter Imaging Data*, The Astrophysical Journal, 785, 144

Gers A. F., Schraudolph N. N., Schmidhuber J., 2002, *Learning precise timing with lstm recurrent networks*, Journal of Machine Learning Research, pp 115–143

Giocoli C., Meneghetti M., Metcalf R. B., Ettori S., Moscardini L., 2014, *Mass and concentration estimates from weak and strong gravitational lensing: a systematic study*, Monthly Notices of the Royal Astronomical Society, 440, 1899–1915

Goodfellow I., Bengio Y., Courville A., 2016, *Deep Learning*. The MIT Press

Grillo C., 2012, *On the Average Density Profile of Dark-matter Halos in the Inner Regions of Massive Early-type Galaxies*, The Astrophysical Journal Letters, 747, L15

Guo Q., et al., 2011, *From dwarf spheroidals to cD galaxies: simulating the galaxy population in a ΛCDM cosmology*, Monthly Notices of the Royal Astronomical Society, 413, 101 –131

Hanley J. V. & McNeil B., 1982, *The meaning and use of the area under a receiver operating characteristic (ROC) curve*, Radiology, 143, 29–36

Hastie T., Tibshirani R., Friedman J., 2009, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag New York, doi:10.1007/978-0-387-84858-7

He K., Zhang X., Ren S., Sun J., 2016, *Deep Residual Learning for Image Recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778

Hebb D. O., 1949, *The organization of behavior: A neuropsychological theory*. Wiley, doi:10.1016/s0361-9230(99)00182-3

Hinton G., Srivastava N., Swersky K., 2012, *Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.*, http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf

Hogg, D. W. et al. 2004, *The Dependence on Environment of the Color-Magnitude Relation of Galaxies*, The Astrophysical Journal, 601, L29 –L32.

Huerta E. A., et al., 2019, *Enabling real-time multi-messenger astrophysics discoveries with deep learning*, Nature Reviews Physics, 1, 600–608

Ioffe S., Szegedy C., 2015, in Bach F., Blei D., eds, Proceedings of Machine Learning Research Vol. 37, Proceedings of the 32nd International Conference on Machine Learning. PMLR, Lille, France, pp 448–456, http://proceedings.mlr.press/v37/ioffe15.html

Jackson N., 2008, *Gravitational lenses and lens candidates identified from the COSMOS field*, Monthly Notices of the Royal Astronomical Society, 389, 1311–1318

Jacobs C., et al., 2019, *Finding high-redshift strong lenses in DES using convolutional neural networks*, Monthly Notices of the Royal Astronomical Society, 484, 5330 –5349

Jaffe, A. H. et al. 2001, *Cosmology from MAXIMA-1, BOOMERANG, and COBE DMR Cosmic Microwave Background Observations*, Physical Review Letters, 86, 3475–3479.

Jeans, J. H. 1902, *I. The stability of a spherical nebula*, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 199, 1–53.

Kaiser N., Squires G., Broadhurst T., 1995, *A Method for Weak Lensing Observations*, The Astrophysical Journal, 449, 460

Katayama, H. et al. 2003, *Properties of the Brightest Cluster Galaxy and Its Host Cluster*, The Astrophysical Journal, 585, 687.

Kingma D. P., Ba J., 2017, *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980)

Kodi Ramanah D., Wojtak R., Ansari Z., Gall C., Hjorth J., 2020, *Dynamical mass inference of galaxy clusters with neural flows*, Monthly Notices of the Royal Astronomical Society, 499, 1985–1997

Koopmans L. V. E., Treu T., Fassnacht C. D., Blandford R. D., Surpi G., 2003, *The Hubble Constant from the Gravitational Lens B1608+656*, The Astrophysical Journal, 599, 70–85

LSST Science Collaboration et al., 2009, *LSST Science Book, Version 2.0*, arXiv e-prints

Lacey, C. & Cole, S. 1993, *Merger rates in hierarchical models of galaxy formation*, Monthly Notices of the Royal Astronomical Society, 262, 627 –649.

Lagattuta D. J., et al., 2019, *Probing 3D structure with a large MUSE mosaic: extending the mass model of Frontier Field Abell 370*, Monthly Notices of the Royal Astronomical Society, 485, 3738–3760

Lanusse F., Ma Q., Li N., Collett T. E., Li C.-L., Ravanbakhsh S., Mandelbaum R., Póczos B., 2018, *CMU DeepLens: deep learning for automatic image-based galaxy-galaxy strong lens finding*, Monthly Notices of the Royal Astronomical Society, 473, 3895 –3906

LeCun Y., Bengio Y., 1998, *Convolutional Networks for Images, Speech, and Time Series*. MIT Press, Cambridge, MA, USA, p. 255–258

Lecun Y., Bottou L., Bengio Y., Haffner P., 1998, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86, 2278 –2324

Lin M., Chen Q., Yan S., 2013, *Network In Network*, http://arxiv.org/abs/1312.4400

Longair, S. M. 2011, *High Energy Astrophysics*. Cambridge University Press.

Lotz J. M., et al., 2017, *The Frontier Fields: Survey Design and Initial Results*, The Astrophysical Journal, 837, 97

Meneghetti, M. 2018, *Introduction to Gravitational Lensing - Lecture scripts*, https://www.researchgate.net/publication/310620466_Introduction_to_Gravitational_Lensing_-_Lecture_scripts

Meneghetti M., et al., 2017, *The Frontier Fields lens modelling comparison project*, Monthly Notices of the Royal Astronomical Society, 472, 3177–3216

Meneghetti M., et al., 2020, *An excess of small-scale gravitational lenses observed in galaxy clusters*, Science, 369, 1347–1351

Merritt, D. 1985, *Relaxation and tidal stripping in rich clusters of galaxies. III. Growth of a massive central galaxy.*, The Astrophysical Journal, 289, 18 –32.

Merten J., Giocoli C., Baldi M., Meneghetti M., Peel A., Lalande F., Starck J.-L., Pettorino V., 2019, *On the dissection of degenerate cosmologies with machine learning*, Monthly Notices of the Royal Astronomical Society, 487, 104–122

Metcalf R. B., Petkova M., 2014, *GLAMER - I. A code for gravitational lensing simulations with adaptive mesh refinement*, Monthly Notices of the Royal Astronomical Society, 445, 1942 –1953

Metcalf R. B., et al., 2019, *The strong gravitational lens finding challenge*, Astronomy & Astrophysics, 625, A119

Navarro, J. F., Frenk, C. S., White S. D. M. 1996, *The Structure of Cold Dark Matter Halos*, The Astrophysical Journal, 462, 263 –575.

Newman, A. B. et al. 2011, *The Dark Matter Distribution in A383: Evidence for a Shallow Density Cusp from Improved Lensing, Stellar Kinematic, and X-ray Data*, The Astrophysical Journal Letters, 728, L39.

Newton, I. 1704, *Opticks, or, a Treatise of the Reflections, Refractions, Inflections and Colours of Light*. Royal Society

Nielsen M., 2015, *Neural Networks and Deep Learning*. Determination Press, http://neuralnetworksanddeeplearning.com/index.html

Nowlan S. J., Hinton G. E., 1992, *Simplifying Neural Networks by Soft Weight-Sharing*, Neural Computation, 4, 473–493

Ntampaka M., Trac H., Sutherland D. J., Battaglia N., Póczos B., Schneider J., 2015, *A Machine Learning Approach for Dynamical Mass Measurements of Galaxy Clusters*, The Astrophysical Journal, 803, 50

Okabe N., Takada M., Umetsu K., Futamase T., Smith G. P., 2010, *LoCuSS: Subaru Weak Lensing Study of 30 Galaxy Clusters*, Publications of the Astronomical Society of Japan, 62, 811

Ostriker, J. P. & Hausman, M. A. 1977, *Cannibalism among the galaxies: dynamically produced evolution of cluster luminosity functions.*, Astrophysical Journal, Part 2 - Letters to the Editor, 217, L125 –L129.

Overzier R., Lemson G., Angulo R. E., Bertin E., Blaizot J., Henriques B. M. B., Marleau G. D., White S. D. M., 2013, *The Millennium Run Observatory: first light*, Monthly Notices of the Royal Astronomical Society, 428, 778 –803

Padmanabhan, T. 2003, *Cosmological Constant - the Weight of the Vacuum*, Physics Reports, 380, 235 –320.

Pawase R. S., Courbin F., Faure C., Kokotanekova R., Meylan G., 2014, *A 7 deg^2 survey for galaxy-scale gravitational lenses with the HST imaging archive*, Monthly Notices of the Royal Astronomical Society, 439, 3392 –3404

Penzias, A. A. & Wilson, R. W. 1965, *A Measurement of Excess Antenna Temperature at 4080 Mc/s.*, The Astrophysical Journal, 142, 419 –421.

Perlmutter, S. et al. 1999, *Measurements of Omega and Lambda from 42 High-Redshift Supernovae*, The Astrophysical Journal, 517, 565–586.

Petkova M., Metcalf R. B., Giocoli C., 2014, *GLAMER - II. Multiple-plane gravitational lensing*, Monthly Notices of the Royal Astronomical Society, 445, 1954 –1966

Petrillo C. E., et al., 2019, *Testing convolutional neural networks for finding strong gravitational lenses in KiDS*, Monthly Notices of the Royal Astronomical Society, 482, 807–820

Planck collaboration et al. 2020, *Planck 2018 results - VI. Cosmological parameters*, Astronomy & Astrophysics, 641, A6, 67.

Reddi S. J., Kale S., Kumar S., 2019, *On the Convergence of Adam and Beyond* (arXiv:1904.09237)

Rephaeli, Y. 1995, *Comptonization of the Cosmic Microwave Background: The Sunyaev-Zeldovich Effect*, Annual Review of Astronomy and Astrophysics, 33, 541 –579.

Riess, A. G. et al. 1998, *Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant*, The Astronomical Journal, 86, 1009–1038.

Rojas R., 2009, *Neural Networks. A systematic Introduction*. Springer-Verlag Berlin Heidelberg, doi:10.1007/978-3-642-61068-4

Rosati, P. et al. 2002, *The Evolution of X-ray Clusters of Galaxies*, Annual Review of Astronomy and Astrophysics, 40, 539 –577.

Sadeh I., Abdalla F. B., Lahav O., 2016, *ANNz2: Photometric Redshift and Probability Distribution Function Estimation using Machine Learning*, Publications of the Astronomical Society of the Pacific, 128, 104502

Sanders, J. S. et al. 2020, *Measuring bulk flows of the intracluster medium in the Perseus and Coma galaxy clusters using XMM-Newton*, Astronomy & Astrophysics, 633, A42.

Schaefer C., Geiger M., Kuntzer T., Kneib J. P., 2018, *Deep convolutional neural networks as strong gravitational lens detectors*, Astronomy & Astrophysics, 611, A2

Schneider, P., Ehlers, J., Falco, E. E. 1992, *Gravitational Lenses*. Springer Verlag, doi:10.1007/978-3-662-03758-4

Schutz B., 2009, *A first course in general relativity*. Cambridge University Press.

Seidel G., Bartelmann M., 2007, *Arcfinder: an algorithm for the automatic detection of gravitational arcs*, Astronomy and Astrophysics, 472, 341 –352

Sérsic J. L., 1963, *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, 41–43

Simonyan K., Zisserman A., 2015, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings

Springel, V. et al. 2005, *Simulating the joint evolution of quasars, galaxies and their large-scale distribution*, Nature, 435, 620 –636.

Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, Journal of Machine Learning Research, 15, 1929–1958

Stark D. P., Ellis R. S., Richard J., Kneib J.-P., Smith G. P., Santos M. R., 2007, *A Keck Survey for Gravitationally Lensed Lyα Emitters in the Redshift Range 8.5<z<10.4: New Constraints on the Contribution of Low-Luminosity Sources to Cosmic Reionization*, The Astrophysical Journal, 663, 10–28

Stehman S. V., 1997, *Selecting and interpreting measures of thematic classification accuracy*, Remote Sensing of Environment, 62, 77–89

Szandała T., 2021, *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks*, Bio-inspired Neurocomputing. Studies in Computational Intelligence, 903

Szegedy C., et al., 2015, *Going deeper with convolutions*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 1–9

Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z., 2016, *Rethinking the Inception Architecture for Computer Vision*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 2818–2826

The Dark Energy Survey Collaboration 2005, *The Dark Energy Survey*, arXiv e-prints, pp astro–ph/0510346

Umetsu, K. 2010, *Cluster Weak Gravitational Lensing*, doi:10.3254/978-1-60750-819-9-269

Vanzella E., et al., 2020a, *Ionizing the intergalactic medium by star clusters: the first empirical evidence*, Monthly Notices of the Royal Astronomical Society, 491, 1093–1103

Vanzella E., et al., 2020b, *Candidate Population III stellar complex at $z = 6.629$ in the MUSE Deep Lensed Field*, Monthly Notices of the Royal Astronomical Society, 494, L81–L85

Wong K. C., et al., 2020, *H0LiCOW - XIII. A 2.4 per cent measurement of H_0 from lensed quasars: 5.3σ tension between early- and late-Universe p*, Monthly Notices of the Royal Astronomical Society, 498, 1420–1439

Xie S., Girshick R., Dollár P., Tu Z., He K., 2017, *Aggregated Residual Transformations for Deep Neural Networks*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 5987–5995

Zeiler M. D., 2012, *ADADELTA: An Adaptive Learning Rate Method* (arXiv:1212.5701)

Zheng, Z. et al. 2005, *Theoretical Models of the Halo Occupation Distribution: Separating Central and Satellite Galaxies*, The Astrophysical Journal, 633, 791 –809.

Zhou Y., Chellappa R., 1988, *Computation of optical flow using a neural network*, IEEE 1988 International Conference on Neural Networks, p. 71–78

de Jong J. T. A., et al., 2015, *The first and second data releases of the Kilo-Degree Survey*, Astronomy & Astrophysics, 582, A62