Astronomy & Astrophysics manuscript no. output
March 1, 2023

Euclid preparation. TBD. Characterization of convolutional neural networks for the identification of galaxy-galaxy strong lensing events

Euclid Collaboration: L. Leuzzi[1,2,⋆], M. Meneghetti[2], G. Angora[3], R. B. Metcalf[1,2], L. Moscardini[1,2,4], P. Rosati[3,5], P. Bergamini[2,6], F. Calura[2], B. Clément[7], R. Gavazzi[8,9], F. Gentile[1,2], C. Grillo[10,11], M. Lochner[12,13], and G. Vernardos[7]
...

[1] Department of Physics and Astronomy of the University of Bologna, Via Gobetti 93/2, 40129 Bologna, Italy
[2] INAF OAS - Astronomical Observatory of Bologna, via Gobetti 93/3, 40129 Bologna, Italy
[3] Department of Physics and Earth Science of the University of Ferrara, Via Saragat 1, 44122 Ferrara, Italy
[4] INFN, Sezione di Bologna, Viale Berti Pichat 6/2, 40127 Bologna, Italy
[5] INFN, Sezione di Ferrara, Via Saragat 1, 44122 Ferrara, Italy
[6] Department of Physics, Universita degli Studi di Milano, via Celoria 16, I-20133 Milano, Italy
[7] Institute of Physics, Laboratory of Astrophysics, Ecole Polytechnique Fédérale de Lausanne (EPFL), Observatoire de Sauverny, 1290 Versoix, Switzerland
[8] Sorbonne Université, UPMC Université Paris 6 and CNRS, UMR 7095, Institut d'Astrophysique de Paris, 98bis boulevard Arago, F-75014 Paris, France
[9] Aix-Marseille Université, CNRS and CNES, Laboratoire d'Astrophysique de Marseille, 38, Rue Frédéric Joliot-Curie, 13388 Marseille, France
[10] Dipartimento di Fisica, Università degli Studi di Milano, via Celoria 16, I-20133 Milano, Italy
[11] INAF - IASF Milano, via A. Corti 12, I-20133 Milano, Italy
[12] Department of Physics and Astronomy, University of the Western Cape, Bellville, Cape Town, 7535, South Africa
[13] South African Radio Astronomy Observatory, 2 Fir Street, Black River Park, Observatory, 7925, South Africa

Received September 15, 1996; accepted March 16, 1997

ABSTRACT

Context. Forthcoming imaging surveys can potentially increase the number of known galaxy-scale strong lenses by several orders of magnitude. For this to happen, images of tens of millions of galaxies will have to be inspected to identify potential candidates. In this context, Deep learning techniques are particularly suitable for the analysis of large data sets, and convolutional neural networks (CNNs) in particular can efficiently process big volumes of images.
Aims. We assess and compare the performance of three network architectures in the classification of strong lensing systems on the basis of their morphological characteristics.
Methods. We train and test our models on different portions of a data set of forty thousand mock images, having characteristics similar to those expected in the survey planned with the ESA mission *Euclid*, gradually including larger fractions of faint lenses. We also evaluate the importance of adding information about the color difference between the lens and source galaxies by repeating the same training on single-band and multi-band images.
Results. Our analysis confirms the potential of the application of CNNs to tackle this problem, since our models find samples of clear lenses with $\gtrsim$ 90% precision and completeness, without significant differences in the performance of the three architectures. Nevertheless, when including lenses with fainter arcs in the training set, the three models' performance deteriorates with accuracy values of $\sim$ 0.87 to $\sim$ 0.75 depending on the model.
Conclusions. We suggest that specific training with separate classes of lenses might be needed for detecting the faint lenses since not even the addition of the color information yields a relevant improvement in this sense.

Key words. Gravitational lensing: strong – Methods: statistical

1. Introduction

Galaxy-galaxy strong lensing (GGSL) events occur when the light emitted by a background galaxy is substantially deflected by a foreground galaxy's gravitational potential. When the observer, the lens, and the source are nearly aligned, and their mutual distances are favorable, the background galaxy appears as a set of multiple images surrounding the lens. These images often have the form of extended arcs or rings.

These events have multiple astrophysical and cosmological applications. For example, GGSL enables probing of the total mass of the lens galaxies within the so-called Einstein radius (e.g., Treu & Koopmans 2004; Gavazzi et al. 2012; Nightingale et al. 2019). By independently measuring the stellar mass and combining lensing with other probes of the galaxy's gravitational potential (e.g., stellar kinematics), one can disentangle the

⋆ e-mail: laura.leuzzi3@unibo.it

dark and baryonic mass distributions, thus studying the interplay between these two mass components (e.g., Barnabè et al. 2011; Suyu et al. 2012; Schuldt et al. 2019). Accurately measuring the dark matter mass profiles and the substructure content of galaxies also enables one to test the predictions of the standard cold dark matter (CDM) model of structure formation and to shed light on the nature of dark matter (e.g., Grillo 2012; Oguri et al. 2014; Vegetti et al. 2018; Minor et al. 2021). In this context, the frequency of GGSL events is a powerful tool to stress-test the CDM paradigm also when it involves cluster members as lenses (Meneghetti et al. 2020, 2022), in addition to help constraining the small scale component of galaxy clusters (e.g., Tu et al. 2008; Desprez et al. 2018; Jauzac et al. 2021; Bergamini et al. 2021). Finally, the lensing magnification makes it possible to study details of the source galaxy's brightness distribution, which would remain unresolved in the absence of the lensing effects (e.g., Impellizzeri et al. 2008; Allison et al. 2017; Stacey et al. 2018).

Less than one thousand galaxy-scale lenses have been discovered so far by employing a variety of methods, including searches for unexpected emission lines in the spectra of elliptical galaxies (Bolton et al. 2006), sources with anomalously high fluxes at submm wavelengths (Negrello et al. 2010, 2017), and sources with unusual shapes (Myers et al. 2003). Some arc and ring finders, which typically looking for blue features around red galaxies, have been developed to search optical images (e.g., Cabanac et al. 2007; Seidel & Bartelmann 2007; Gavazzi et al. 2014; Maturi et al. 2014; Sonnenfeld et al. 2018). Forming extensive catalogs of GGSL systems is arduous due to their rarity, but it is expected that this will change in the next decade thanks to upcoming imaging surveys. In fact, it was estimated that the ESA *Euclid* space telescope (Laureijs et al. 2011) and the Legacy Survey of Space and Time (LSST; LSST Science Collaboration et al. 2009) performed with the Vera Rubin Observatory will observe up to one hundred thousand strong lenses (Collett 2015), thus significantly increasing the number of known systems.

The identification of potential candidates will require the examination of tens of millions of galaxies, thus making the development of reliable and automated methods for the analysis of large volumes of images of fundamental importance. Over the past few years, machine learning (ML) and deep learning (DL) techniques are proving extremely promising in this context.

In particular, convolutional neural networks (CNNs, e.g., LeCun et al. 1989) are a DL algorithm that has been successfully applied to several astrophysical problems and is expected to play a key role in the future of astronomical data analysis. Among the many different applications, they have been employed for estimating the photometric redshifts of luminous sources (Pasquet et al. 2019; Shuntov et al. 2020), for performing the morphological classification of galaxies (Zhu et al. 2019; Ghosh et al. 2020), for constraining the cosmological parameters (Merten et al. 2019; Fluri et al. 2019; Pan et al. 2020), for identifying cluster members (Angora et al. 2020), for quantifying galaxy metallicities (Wu & Boada 2019; Liew-Cain et al. 2021), and for estimating galaxy cluster dynamical masses (Ho et al. 2019; Gupta & Reichardt 2020).

Several CNN architectures were recently used also to identify strong lenses on ground-based wide field surveys such as the Kilo Degree Survey (KiDS; de Jong et al. 2015; Petrillo et al. 2019a,b; He et al. 2020; Li et al. 2020; Napolitano et al. 2020), the Canada-France-Hawaii Telescope Legacy Survey (CFHTLS; Gwyn 2012; Jacobs et al. 2017), the Canada France Imaging Survey (CFIS; Savary et al. 2022) and the Dark Energy Survey (DES; The Dark Energy Survey Collaboration 2005; Jacobs et al. 2019b,a; Rojas et al. 2022). Most of them were also recently employed in two challenges aimed at comparing and quantifying the performance of several methods to find lenses, both based on artificial intelligence and not. The first challenge's results, presented in Metcalf et al. (2019), show that DL methods are particularly promising with respect to other traditional techniques like, visual inspection and classical arcfinders.

In this work, we investigate the ability of three different network architectures in the identification of GGSL systems. We test them on different portions of a data set of *Euclid*-like mock observations. In particular, we evaluate how the inclusion of faint lenses in the training set affects the classification.

The paper is organized as follows: in Sect. 2, we explain how CNNs are implemented and trained to be applied to image recognition problems; in Sect. 3, we introduce the data set of simulated images used for training and testing our networks; in Sect. 4, we describe our experiments and we present and discuss our results; in Sect. 5, we summarize our conclusions.

2. Convolutional neural networks

Artificial neural networks (ANNs; e.g., McCulloch & Pitts 1943; Goodfellow et al. 2016) are a ML algorithm inspired by the biological functioning of the human brain. They consist of artificial neurons, or nodes, that are organized in consecutive layers and linked together through weighted connections. The weights define the sensitivity among individual nodes (Hebb 1949) and are adapted in order to make the network able to carry out a specific task.

The output of the k-th layer $\boldsymbol{h}^k$ depends on the output of the previous layer $\boldsymbol{h}^{k-1}$ as (Bengio 2009):

$$\boldsymbol{h}^k = f(\boldsymbol{b}^k + \mathsf{W}^k \boldsymbol{h}^{k-1}). \quad (1)$$

Here $\boldsymbol{b}$ is the vector of offsets (biases) and W^k is the weight matrix associated with the layer: their dimension is defined by the number of nodes in the layer; f is the activation function, that introduces non-linearity in the network that would otherwise only be characterized by linear operations.

CNNs are a special class of ANNs that make use of the convolution operation. Thanks to this property, they perform particularly well on image recognition and classification problems. The basic structure of a CNN can be described as a sequence of convolutional and pooling layers. Convolutional layers consist of a series of filters, also called kernels, that are matrices of weights of typical dimension 3×3 to 7×7 and act as the weights of a generic ANN. They are convolved with the input of the layer to produce the feature maps, that are passed as input to the following layer. The organization of the filters in multiple layers ensures that the CNN can learn complex mappings between the inputs and outputs by dividing them into a series of simpler functions, each of which extracts relevant features from the images. The pooling operation downsamples each feature map by dividing it into quadrants of typical dimension 2×2 or 3×3 and substituting them with a summary statistic, such as the maximum (Zhou & Chellappa 1988). This operation has the twofold purpose of reducing the size of the feature maps, and therefore the number of parameters of the model, and making the architecture invariant to little modifications of the input (Goodfellow et al. 2016). After these layers, the feature maps are flattened into a 1-D vector that is processed by fully-connected layers, and is then passed to the output layer, that predicts the output. In classification problems, the activation function used for the output layer is often the softmax, providing an output in the range $[0, 1]$ that can be interpreted (Bengio 2009) as an indicator of $P(Y = i|\boldsymbol{x})$,

where Y is the class associated with the input $\boldsymbol{x}$, among all the possible classes i.

CNNs master the execution of a given task as a result of a supervised learning process, called training, in which they analyze thousands of known input-output couples. The weights of the network, that are randomly initialized, are readjusted so that the network's predictions of the output are correct for the largest amount of possible examples. This step is of crucial importance, since the weights are not modified afterwards, when the final model is applied to other data. The aim of the training is to minimize a loss (or cost) function that gives an estimation of the difference between the outputs predicted by the network and the true labels. In order to do this, the images are passed to the network several times, and at the end of each pass, called an epoch, the gradient of the cost function is computed with respect to the weights and backpropagated (Rumelhart et al. 1986) from the output to the input layer, so that the kernels can be adapted accordingly. The magnitude of the variation of the weights is regulated through the learning rate, a hyperparameter to be defined at the beginning of the training.

In addition to showing a good performance on the training set, it is essential that the network generalizes to other images. Preventing the model from overfitting (i.e. memorizing peculiar characteristics of the images in the training set that can not be used to make correct predictions on other data sets) is possible by monitoring the training with a validation step. At the end of each epoch, the performance of the network is assessed on the validation set, that is a small part of the data set (usually 5 – 10%) excluded from the training set. If the loss function evaluated on these images does not improve for several consecutive epochs, the training will be interrupted or the learning rate reduced. Once the training is completed, the performance of the final model is evaluated on the test set, a part of the data set (about the 20 – 25%) excluded from the other subsets. Afterwards, the CNN can be applied to new images.

Using CNNs enables one to handle large data sets conveniently for several reasons. While the training can take up to a few days to be completed, processing a single image afterwards requires a fraction of a second, thanks to the employment of Graphics Processing Units (GPUs). Moreover, the feature extraction process that is carried out during the training is completely automated. The algorithm selects the most significant characteristics for achieving the best results without any previous knowledge of the data.

In the following Section, we provide more information about the specific architectures we test in this work and technical details about our training.

2.1. Network architectures

We implement three CNN architectures: a VGG-like network (Simonyan & Zisserman 2015), an inception network (IncNet; Szegedy et al. 2015, 2016) and a residual network (ResNet; He et al. 2016; Xie et al. 2017).

The definition of the final configuration of the networks that we apply to the images is the result of several trials, in which we have tested different hyperparameters for the optimization (such as the learning rate) and general architectures (such as the amount of layers and kernels) in order to find the most suitable arrangement for our classification problem.

2.1.1. VGG-like network

The Visual Geometry Group Network (VGGNet) was first presented in the work by Simonyan & Zisserman (2015). The most significant innovation introduced with this architecture is the application of small convolutional filters, with a receptive field of 3×3. This allowed the construction of deeper models, since the introduction of small filters keeps the number of trainable parameters in the CNN smaller than that of networks that make use of larger filters (e.g., of dimension 5×5 or 7×7). Since the concatenation of multiple kernels of size 3×3 has the same resulting receptive field of larger filters, it is possible to analyze features of larger size, while building deeper architectures.

Our implementation of the VGGNet is composed of ten convolutional layers and five max pooling layers alternating. In particular, we use the leaky rectified linear unit (Leaky ReLU; Xu et al. 2015) at the end of every convolutional layer, in this and in the other two architectures. At the end of each convolutional-pooling block we perform the batch normalization of the output of the block. Batch normalization is a technique employed to accelerate and stabilize the training of deep networks, that consists in the renormalization of the layer inputs (Ioffe & Szegedy 2015). Moreover, the output layer, that is a softmax layer, is preceded by two fully connected layers, that are alternated with dropout layers. The amount of parameters that constitute the architecture amounts to about two millions.

When training on multi-band observations we also add a second input channel to process the Near Infrared Spectrometer and Photometer (NISP) images. Since they are smaller than the Visual Imager (VIS) images (see Table 1), this branch of the network is only eight layers deep. The outputs of the two branches are concatenated before being passed to the output layer. In this configuration our network uses about three million parameters.

In Appendix A, Fig. A.1 shows the VGG-like network configuration we tested on the VIS images (a) and on the multi-band images (b).

2.1.2. Inception network

The reasons for the IncNet architecture were outlined in the work by Szegedy et al. (2015), who applied the ideas of Lin et al. (2013) to CNNs. Trying to improve the performance of a CNN by enlarging its depth and width leads to a massive increase of the number of parameters of the model, favouring overfitting and increasing the requirements of computational resources. Szegedy et al. (2015) suggest applying filters with different size to the same input, making the model learn features on different scales in the same feature maps. This is implemented through the inception module. In the simplest configuration, each module applies filters of several sizes (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 computational cost of the operations. An IncNet is a series of such modules stacked upon each other. A further improvement of the original inception module design is presented in the work by Szegedy et al. (2016): 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. This version of the inception module is the one used in our network implementation.

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

which 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 output layers are both softmax layers. The total number of parameters that compose the model is approximately two millions.

The configuration used to analyze the multi-band images has a secondary branch with one initial convolutional layer and seven inception modules. This branch is characterized by one million parameters, thus leading to a total of three million parameters.

In Appendix A, Fig. A.2 shows the IncNet configuration we tested on the VIS images (a) and on the multi-band images (b).

2.1.3. Residual network

He et al. (2016) introduced residual learning as a way to make the training of deep networks more efficient. The basic idea behind the ResNets is that it is easier for a certain layer (or a few stacked layers) to learn a residual function with respect to the input, rather than the complete, and more complicated, direct mapping. In practice, this is implemented using a residual block with shortcut connections. The input of the block $\boldsymbol{x}$ 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 written as

$$\mathcal{F}(\boldsymbol{x}) := \mathcal{H}(\boldsymbol{x}) - \boldsymbol{x}, \tag{2}$$

where $\mathcal{H}(\boldsymbol{x})$ is the original function and $\mathcal{F}(\boldsymbol{x})$ is the residual function. Thus, the original function can be computed as $\mathcal{F}(\boldsymbol{x})+\boldsymbol{x}$. 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, that can be presented as

$$\mathcal{F}(\boldsymbol{x}) = \sum_{i=1}^{C} \mathcal{T}_i(\boldsymbol{x}) \tag{3}$$

and can serve as the residual function in Eq. (2). Here $\mathcal{T}_i(\boldsymbol{x})$ is an arbitrary function and C is a hyperparameter called *cardinality*, that represents the size of the set of transformations to be aggregated.

The fundamental block of our ResNet is this version of the residual block, with cardinality equal to eight. In particular, the input is initially processed by two convolutional and pooling layers and then passed to four residual blocks alternated with two max pooling layers. Before being passed to the final softmax layer, dropout is performed on the resulting feature maps. Moreover, batch normalization is performed after every max pooling layer. The NISP images are processed by a similar branch, which differs from this one in having only one initial convolutional layer.

The parameters of the model number one million in the VIS configuration and two millions in the multi-band configuration, and thus are significantly fewer than in the other examined architectures. However, this configuration of the ResNet outperformed the other possible ResNets that we assessed when designing our models.

In Appendix A, Fig. A.3 shows the ResNet configuration we applied to the VIS images (a) and on the multi-band images (b).

3. The data set

Training CNNs requires thousands of labeled examples. Since not many observed galaxy-scale lenses are known to date, simulating the events is often necessary. The realism of the simulations is essential for ensuring that the evaluation of the model's performance is indicative of the results we may expect on real observations.

The image simulations use the galaxy and halo catalogs provided by the Flagship simulation (v1.10.11; Castander et al., in prep.) through the CosmoHub portal[1] (Carretero et al. 2017; Tallada et al. 2020).

We construct the images using the following procedure.

- We randomly select a trial lens galaxy from the light cone subject to a magnitude cut of 23 in the $I_{\rm E}$ band. After this, we randomly select a source from a catalog of Hubble Ultra Deep Field (UDF; Coe et al. 2006) sources with known redshifts. We decompose these sources into shapelets for denoising, following the procedure described in Meneghetti et al. (2008, 2010). The mass of the lens is represented by a truncated singular isothermal ellipsoid and a Navarro, Frenk & White (NFW; Navarro et al. 1996) halo.
- We use the GLAMER lensing code (Metcalf & Petkova 2014; Petkova et al. 2014) to perform the ray-tracing. Light rays coming from the position of the observer are shot within a $20'' \times 20''$ square centered on the lens object, with initial resolution of $0''.05$, twice the final resolution of the VIS instrument. We use these rays to compute the deflection angles that will trace the path of the light back to the sources.
- The code detects any caustics in the field and does some further refinement to characterize them. Specifically, more rays are shot in a region surrounding the caustics to constrain their position with higher resolution. If the area within the largest critical curve is larger than 0.2 arcsec^2 and smaller than 20 arcsec^2 the object is accepted as a lens of the appropriate size range.
- The lensed image is constructed using the shapelet source and Sérsic profiles for the lens galaxy and any other galaxy that appears within the field. We take the parameters for these profiles from the Flagship catalog with some randomization. While we place the lens galaxy at the centre of the cutout, the positions of the other galaxies are determined following the Flagship catalogs as well, with some randomization. In this way, the density of galaxies along the line of sight is the same as that of the Flagship simulations, but the sources will have a different angular position.
- We place the galaxy source at a random point on the source plane within a circle surrounding the caustic. We add noise to the images using an approximation of the *Euclid* PSFs (Euclid collaboration et al., in prep.) and noise levels. The noise is simulated with a gaussian random field, to reproduce the noise level expected by the Euclid Wide Survey.
- To increase the number of images, at a low computational cost, we create two other images by rotating the lens galaxy in three dimensions and creating the new source position by randomizing the previous one and moving the companion galaxies.

This procedure is similar to the one used for the Lens Finding Challenges and described in more detail in Metcalf et al. (2019).

The result of these simulations are one hundred thousand *Euclid*-like mock images simulated in the $I_{\rm E}$ band of the VIS instrument and $H_{\rm E}$, $Y_{\rm E}$, $J_{\rm E}$ bands of the Near-Infrared Spectrometer and Photometer (NISP) instrument. The dimensions of the VIS and NISP images are 200×200 and 66×66 pixels, respectively. Given the resolution of the instruments, reported in Table 1, these correspond to $20'' \times 20''$.

[1] https://cosmohub.pic.es/home

Table 1. Main characteristics of the *Euclid* VIS and NISP instruments.

Instrument	Capability	λ range (nm)	Pixel size (arcsec)
VIS	Visual imaging	550 - 900	0.1
NISP	NIR imaging photometry	Y_E (920 - 1146), J_E (1146 - 1372), H_E (1372 - 2000)	0.3

We initially clean the data set by dropping the images with a source at $z > 7$, thus leaving a catalog of 99 409 objects. The images in the data set are considered lenses if they meet the following criteria simultaneously:

$$\begin{cases} \texttt{n\_source\_im} > 0; \\ \texttt{mag\_eff} > 1.6; \\ \texttt{n\_pix\_source} > 20. \end{cases} \quad (4)$$

Here `n_source_im` represents the number of images of the background source, `mag_eff` is the effective magnification of the source, and `n_pix_source` is the number of pixels in which the surface brightness of the source is 1σ above the background noise level. The magnification is computed as the ratio of the sum of all the pixels with a flux above the noise level in the lensed images on the image plane and the unlensed images pixels on the source plane. The same criteria were adopted in the Lens Finding Challenge 2.0[2] (Metcalf et al., in prep.),

In many cases one or more background sources are present in the non-lenses, but they are too faint and/or too weakly magnified to be classified as a lens. For this reason, the parameters `n_pix_source` and `mag_eff` are also considered in the classification criteria (Eq. 4). Of course, depending on the sensitivity of the model, the classification of the borderline images might vary, while the clearest ones should be immediately assigned to the correct category.

By using these conditions, we divide the images we simulated into 19 591 lenses and 79 816 non-lenses, thus obtaining two very unbalanced classes out of the complete data set. This would not guarantee the training procedure to be successful, since it is essential that a similar number of images are assigned to all the considered categories. For this reason, we decide to use all the lenses for the training, and to randomly select only a subsample of 20 000 non-lenses.

In Fig. 1, we report the distribution of some properties of the images in the data set. From top-left to bottom-right, we show the distribution of the redshift of the galaxy lenses and sources, of the magnitude of the galaxy lenses and sources, of the Einstein area of the largest critical curve in the lensing system and of `n_pix_source`. The histograms in each panel refer to the lenses (green) and non-lenses (red) separately, and to the complete data set (blue). The galaxy lenses in the two classes share similar distributions of redshift, magnitude and Einstein radius (top, middle and bottom-left panels, respectively). The sources' redshift distribution, in the top-right panel, is also similar for the two subsets. On the other hand, the magnitude of the simulated sources (middle-right panel) in the non-lenses class is on average fainter than that of the sources in the lenses. This is intuitive, since sources with brighter magnitude will be more evident in the images and it will be more likely that they produce a clear lensing event. A similar argument can be made about `n_pix_source` (bottom-right panel): the higher is the value of this parameter, the clearer will be the distortion of the source's images, hence the lensing system.

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

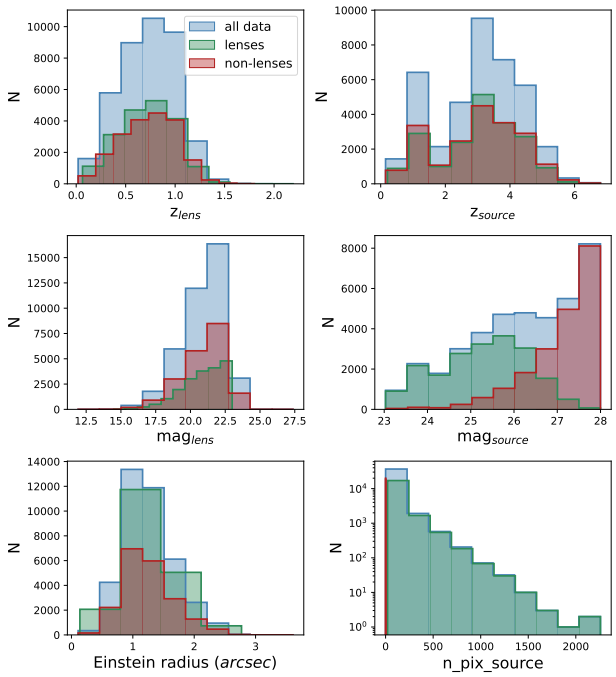

Fig. 1. Distribution of several properties of the simulated images in the data set (blue histograms). The distributions of the same properties in the separate subsets of lenses and non-lenses are given by the green and red histograms, respectively. In the upper- and middle-row panels we show the distributions of lens and source redshifts and magnitudes (in the case of the sources, we refer to the intrinsic magnitude). The bottom panels show the distributions of the areas within the lens Einstein radii and of the number of pixels where the source brightness exceeds 1σ above the background noise level.

4. Results and discussion

4.1. Performance evaluation

We assess the performance of our trained networks by examining the properties of the catalogs produced by the classification of the images in the test set. In particular, we take into consideration four statistical metrics that are immediately derived from the *confusion matrix* (Stehman 1997). A generic element of the confusion matrix C_{ij} is given by the amount of images belonging to the class i and classified as members of the class j. In a binary classification problem, like the one considered here, the diagonal elements indicate the amounts of correctly classified objects, i.e. the True Positives (TP) and the True Negatives (TN), while the off-diagonal terms show the amounts of misclassified objects, i.e. the False Positives (FP) and the False Negatives (FN).

Considering the class of the Positives, the combination of these quantities leads to the definition of the following metrics:

- The precision (P) can be computed as

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (5)$$

  and it measures the level of purity of the retrieved catalog.
- The recall (R) can be computed as

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (6)$$

and it measures the level of completeness of the retrieved catalog.

– The F1-score (F1) is the harmonic average of P and R,

$$\mathrm{F1} = 2\,\frac{P\,R}{P+R}. \tag{7}$$

– The accuracy (A) is the ratio between the amount of correctly classified objects and the total amount of objects,

$$A = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}. \tag{8}$$

The first three indicators can be similarly computed for the class of the Negatives, while the accuracy is a global indicator of the performance.

In addition, we compute the receiver operating characteristic (ROC; Hanley 1982) curve, that visually represents the variation of the True Positive Rate (TPR) and False Positive Rate (FPR) with the detection threshold $t \in (0, 1)$, that is used to discriminate whether an image contains a lens or not. The area under the ROC curve (AUC) summarizes the information conveyed by the ROC: while one would be the score of a perfect classifier, 0.5 indicates that the classification is equivalent to a random choice, hence worthless.

4.2. Data preprocessing

The data preparation consists of a sequence of several steps. We divide each data set into three parts: the training set (70%), the validation set (5%) and the test set (25%). The images in the data set are randomly assigned to one of these subsets, but we check that the validation and test sets are well described by the training set before starting the training. We do this by inspecting the distributions of several parameters that define the characteristics of the lenses and sources in the data set, such as their redshift, magnitude and Einstein radius.

Once the data set is split, we randomly select 20% of the images in the training set and augment them by rotating them of 90°, 180° and 270° and flipping them with respect to the horizontal and vertical axes. The augmentation doubles the amount of images in the training set. Afterwards, we normalize the images in the data set by subtracting the mean and dividing them by the standard deviation of the training set. The reason for this type of normalization is that the computation of the gradients in the training stage of the networks is easier if the features in the training set are in a similar range. Moreover, scaling the inputs in this way makes the parameter sharing more efficient (Goodfellow et al. 2016).

4.3. Training procedure

We implement, train and test our networks using `Keras`[3] (Chollet 2015) 2.4.3 with `TensorFlow`[4] (Abadi et al. 2016) 2.2.0 backend on a NVIDIA Titan Xp Graphics Processing Unit (GPU).

We conduct twenty-four trainings of 100 epochs each on the selections of the data set described in Sect. 4.4, since we train each architecture on each selection of data. Twelve of them use the VIS images, the other twelve use the NISP bands too. We use the Adaptive moment estimation (Adam; Kingma & Ba 2017; Reddi et al. 2019) optimizer with initial learning rate of 10^{-4}.

[3] `https://keras.io/`
[4] `https://www.tensorflow.org/`

We employ the binary cross-entropy $\mathcal{L}$ to estimate the loss at the end of each epoch:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y(\mathbf{x}_i)\ \ln[y_p(\mathbf{x}_i)] + [1-y(\mathbf{x}_i)]\ \ln[1-y_p(\mathbf{x}_i)], \tag{9}$$

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 one as predicted by the network, so that $y_p - 1$ is the probability that the i-th example has label zero.

The trainings are also monitored by estimating the performance of the network on the validation set at the end of every epoch. If the loss function evaluated on this independent subset does not decrease for twenty consecutive epochs, the training will be stopped with the `EarlyStopping`[5] class from `Keras`. This method is particularly useful to avoid overfitting.

At the end of training we use the best models, i.e. those that have the minimum value of the loss function on the validation set, for our tests.

4.4. Experiments

The identification of GGSL events is primarily based on their distinctive morphological characteristics, namely on the distortion of the images of the background source into arcs and rings, and on the color difference between the foreground and background galaxies. However, real lenses can show complex configurations and might be not so easily recognizable. Our experiments aim at evaluating the ability of CNNs to detect the less clear lenses and at assessing their performance on a diversified data set.

We do this by training the networks on four selections of images, named from S1 to S4, that gradually include a greater fraction of borderline objects. These samples consist of approximately two thousand, ten thousand, twenty thousand images, and the complete data set, respectively. They are built to have an approximately equal number of lenses and non-lenses (see Table 2). The criteria we adopt to progressively broaden our selections take into account the features that might be employed by the networks to classify the objects as members of the correct category.

In the case of the non-lenses, the lack of a background source, or the absence of its images, makes the classification more likely to be correct. Therefore, we initially consider a sample of the approximately ten thousand non-lenses without a background source. Specifically, we select one thousand of them in S1, five thousand in S2, and ten thousand in S3. In S4, we broaden our sample by including the images where a background source has been added, but does not correspond to a visible image, extending our selection to the other objects that are classified as non-lenses according to the criteria in Eq. (4).

In the case of the lenses, the definition of an effective criterion to identify the clearest examples in the data set 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 system, since several factors contribute to the actual clarity of the observable features. Among them are the magnitude of the source and the extension of the image produced by the lensing effect. After several tests involving these parameters and others (such as the Einstein area and the magnification of the sources), we deem `n_pix_source` to be an appropriate choice to discriminate between clear and faint lenses.

[5] `https://keras.io/api/callbacks/early_stopping/`

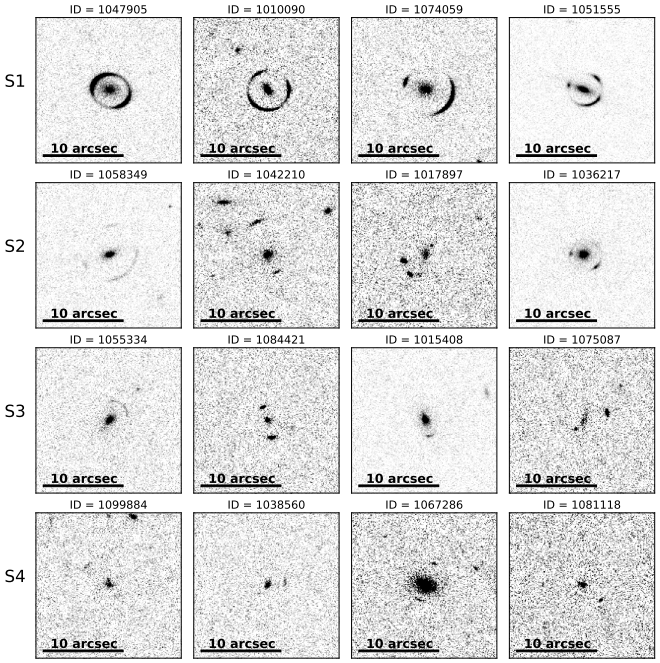

Fig. 2. From top to bottom row, we show four random lenses extracted respectively from S1, S2, S3, S4. Here, it is evident the effect that using different thresholds of the parameter `n_pix_source` has on the selection of images of lenses we use in training phase.

The complete sample of the lenses is characterized by the minimum value `n_pix_source` = 20. From S4 to S1, we increase this threshold to different values, that depend on the number of images we seek to isolate: the higher the value employed, the smaller will be the number of images selected and the clearer the lenses. The thresholds established for the creation of the selections described so far also take into account the necessity to have comparable number of images in each of the two classes, so that the examples analyzed by the networks in the training phase are balanced. In Table 2, we give a summary of the criteria used to identify the images to include in each selection. We also show in Fig. 2 some randomly chosen examples of lenses that are characteristic of each selection, to better illustrate what the effect of using different thresholds for `n_pix_source` is on the training sets we use.

We train and test on these selections of the data set the three architectures, previously discussed: a VGG-like network (Simonyan & Zisserman 2015), an IncNet (Szegedy et al. 2015, 2016), and a ResNet (He et al. 2016; Xie et al. 2017).

The best results of each architecture and each the classification experiment, that are conducted using the I_E band images, are summarized in Table B.1, where the precision, recall, F1-score, accuracy and AUC obtained from the application of our models are reported.

4.5. Discussion

By studying how the metrics depend on the selections, we find that the ability of our networks to correctly classify the images tends to deteriorate as the fraction of included borderline objects increases.

The trend of the accuracy is pictured in Fig. 3. Our three models succeed in the classification of the objects in the selections S1 and S2, where this parameter is in the range $\sim$ 0.9 to $\sim$ 0.96. The IncNet and VGG-like network also perform similarly on S3, while they reach an accuracy level of $\sim$ 0.87 on S4. On the other hand, the ResNet is the worst-performing architecture, with an accuracy of $\sim$ 0.75 on the complete data set.

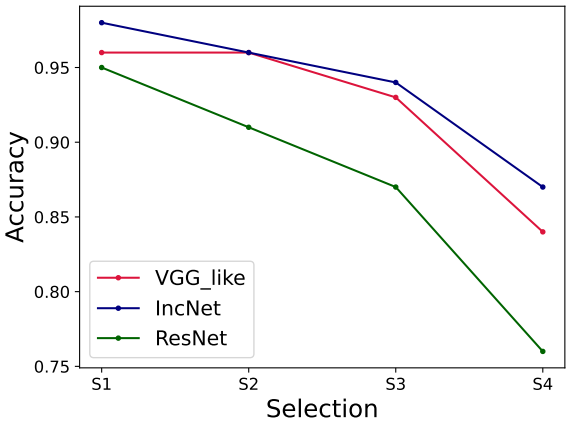

Fig. 3. Trend of the classification accuracy of the VGG-like network (red), the IncNet (blue) and the ResNet (green) tested on the four selections of data.

The precision, recall and F1-score also have similar global trends as that of the accuracy. They are shown in the top, middle and bottom panels of Fig. 4, respectively. These metrics are evaluated separately on the non-lenses (on the left) and on the lenses (on the right), but the same consideration applies to both classes. This suggests that the degradation of the performance does not only affect the identification of the lenses, but it involves the classification of the two categories. In particular, the F1-score, that depends on precision and completeness, peaks at $\sim$ 0.96 on S1 and decreases to $\sim$ 0.87 on S4, with the ResNet being again the worst performing network.

In each panel of Fig. 5, we show the ROC curves of one of our networks, evaluated on the test sets of the selections S1, S2, S3, and S4. Their trends for the IncNet (middle panel) and the ResNet (bottom panel) are similar, with the AUC decreasing by $\sim$ 10% from S1 to S4. It should, however, be pointed out that the IncNet performs systematically better than the ResNet: while the former has an AUC of 0.92 on S1 and 0.81 on S4, the latter AUC ranges from 0.81 on S1 to 0.7 on S4. On the other hand, the ROC of the VGG-like network on S2, S4 has a lower AUC, of $\sim$ 0.57, compared to the other models, and higher AUC values only for the selections S1, S3.

Let us focus on the selection S4, i.e. on the performance of our models on the complete data set. We can see in Fig. 6 nine misclassified non-lenses and Fig. 7 nine misclassified lenses. The images reported in these figures are selected among those misclassified by all three models, therefore they should be characterized by the features that the networks generally find harder to attribute to the correct class.

The False Positives in Fig. 6 are mostly characterized by the coexistence of more than one source in addition to the lens galaxy, that might be mistaken for the multiple images of the same source. The misinterpretation of these objects might be exacerbated by the inclusion of several borderline lenses in the training set. In fact, many of the lenses in the labeled examples do not present clear arcs or rings, and the faint distortions encountered in the feature extraction process are likely to resemble specific morphological features of non-lensed galaxies, such as spiral arms, or isolated, but elongated galaxies.

Table 2. Summary of the criteria adopted to choose the images included in the different selections of lenses and non-lenses for our experiments. 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
	Criterion	Number of images	Criterion	Number of images	
S1	`n_pix_source` >430	1001	Randomly selected objects with `n_sources` = 0	1000	2001
S2	`n_pix_source` >140	5083	Randomly selected objects with `n_sources` = 0	5000	10 083
S3	`n_pix_source` >70	9709	Randomly selected objects with `n_sources` = 0	10 000	19 709
S4	`n_pix_source` >20	19 591	Randomly selected objects with `n_source_im` = 0	20 000	39 591

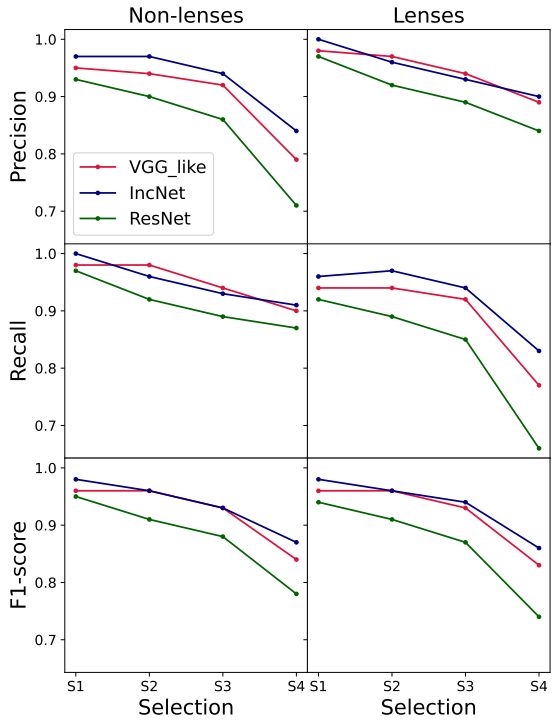

Fig. 4. Trend of the precision (first row), recall (second row) and F1-score (third row) in the classification of the non-lenses (left column) and of the lenses (right column) in the different selections. Different colored lines refer to different networks, as labelled.

The False Negatives in Fig. 7 are in large part not even recognizable as lenses by visual inspection. Although being classified as lenses according to the criteria in Eq. (4), many of these objects do not show evident lensing features. Therefore, if the classification was to be carried out on unlabeled observations, we would not expect the models to be able to identify them as lenses. In some of the images, however, the arc-shaped and ring-shaped sources are evident. Nevertheless, their classification is incorrect, signalling that some clear lenses might also be missed by our classifiers.

In order to further investigate the ability of the networks trained on S4 to actually identify clear lenses, we test them on the images in S2 (test S4/S2). We compare the result of this test with those obtained from training and testing the networks on S2 (test S2/S2): the results of this comparison are reported in Fig. 8 and Table B.2.

The performance of the models trained on S4 in the identification of the lenses in S2 is generally worse than that of the models trained on S2, even though the images that are part of S2 will also inevitably be part of S4, since it consists of the complete data set. While the completeness of the retrieved catalogue of lenses is constant in the two tests, the precision decreases of $\sim 20\%$, passing from ~ 0.95 in the test S2/S2 to ~ 0.73 in S4/S2, with only minor differences between the different architectures. Even though the magnitude of the overall deterioration is not large per se (the accuracy decreases of $\sim 5\%$ for the three networks), it is problematic since it is mostly due to the misclassification of clear lenses, that are also the most useful for scientific purposes.

This result suggests that the performance of the models trained on S4 is worse in general, since the borderline objects are intrinsically harder to classify and a significant fraction of this selection is composed of non-obvious lenses. Moreover, there is a deterioration in the ability of the models to recognize the clearest objects in the data set, as those that are part of S2.

This effect might result from a combination of two complementary factors regarding the characteristics of the images in the data set. First of all, the fraction of clear images in the training set of S4 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 distinguish them. Wide arcs and rings will be recognizable only in a moderate number of images, thus not being as significant as they are in S2 for the classification of the lenses. 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.

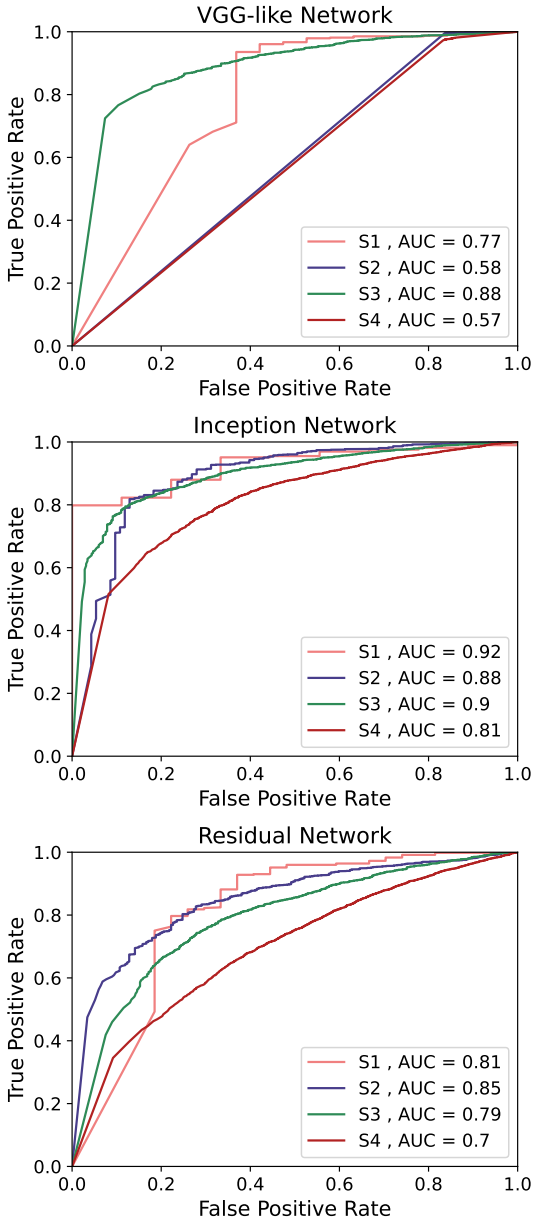

Fig. 5. From top to bottom, each panel of this image shows the ROC curves obtained from the application of the VGG-like network, the IncNet and the ResNet to the test sets of the different selections S1 (pink line), S2 (blue line), S3 (green line) and S4 (red line) of the data set.

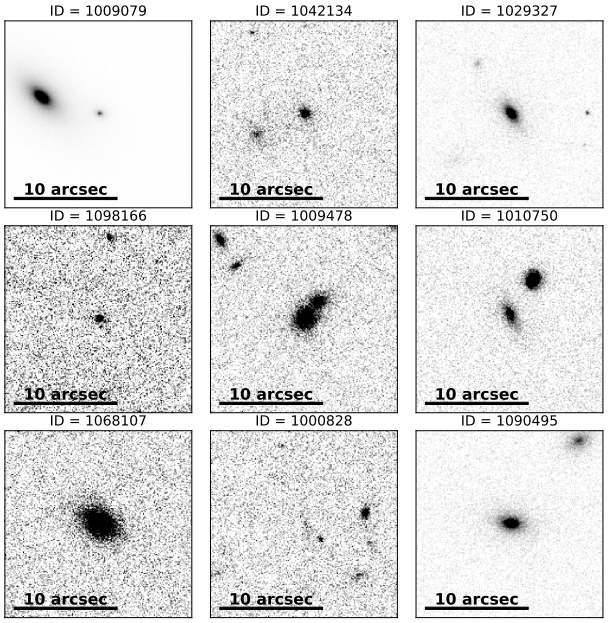

Fig. 6. Example of false positives produced by the three networks applied to the selection S4, here pictured in the I_E band.

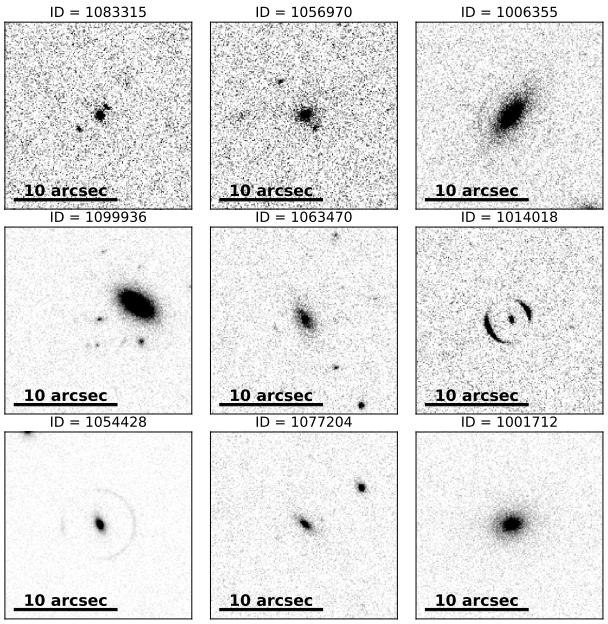

Fig. 7. Example of false negatives produced by the three networks applied to the selection S4, here pictured in the I_E band.

As shown in Fig. 7, the largest fraction of the lenses that are not identified by the networks trained on S4 is actually difficult to identify. However, a certain fraction of evident lenses might also be missed if the training set is extended to include a significant number of borderline objects, as they might be underrepresented.

In addition to this, the architecture of the network appears to be influential in the outcome of the classification only to a certain degree. In particular, when trained and tested on the same selections, the IncNet and the VGG-like network generally perform similarly, when comparing the metrics in Fig. 3 and Fig. 4. The ResNet, on the other hand, performs significantly worse than the others, especially on S4.

4.6. Additional tests

We now test the models trained on S2 on the wider selections S3, S4 (tests S2/S3 and S2/S4, respectively), after removing the objects of these samples that are also included in the training set of S2. This test has the purpose of assessing whether the networks trained on clear examples are flexible enough to detect fainter systems. A deterioration of the performance from S2/S3 to S2/S4 is expected, since CNNs mostly generalize to the images that are similar to those in the data set they have been trained with. Consequently, they might struggle to perform the same task when dealing with images containing features they have never seen before. In the present case, most of the images in the training set of S2 show clear lensing features, while the test sets progressively

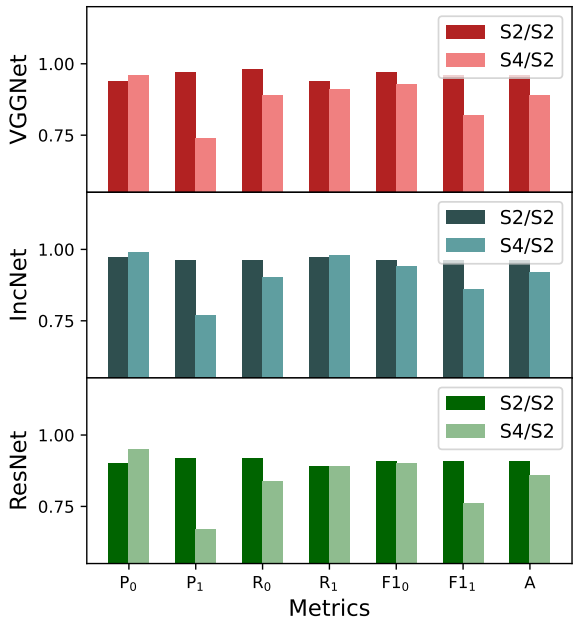

Fig. 8. Comparison of the tests S2/S2 and S4/S2 (darker and lighter histograms, respectively) run with the VGG-like network (top), the IncNet (center) and the ResNet (bottom). In each panel we show the results for the different metrics: from left to right we show the precision on the class of the non-lenses (P_0) and lenses (P_1), the recall on the class of the non-lenses (R_0) and lenses (R_1), the F1-score on the class of the non-lenses ($F1_0$) and lenses ($F1_1$) and the overall accuracy (A).

include a greater fraction of images with new features. Table B.3 summarizes the results of these tests: it is clear that the overall trend is common for the three architectures with a similar performance in both tests.

The general performance of the networks trained on S2 deteriorates on the other broader selections: the accuracy of the classification varies from $\sim$ 0.85 in the case S2/S3 to $\sim$ 0.7 in the case S2/S4. By comparing these results with that of the test S4/S4 in Table B.1, we observe several differences in the precision, recall and F1-score, computed separately for the non-lenses and lenses, as well as in the accuracy.

The purity of the non-lenses decreases when broader selections are used as test sets: the precision reaches the value of $\sim$ 0.64 with S4. On the other hand, the recall is approximately constant at values of $\sim$ 0.96 independently of the considered selection, meaning that the greater part of the objects in this class is correctly identified.

In the case of the lenses, we find a roughly opposite trend. The precision of the classification is roughly constant at $\sim$ 0.94, while the recall decreases drastically from $\sim$ 0.7 in S3 to $\sim$ 0.38 in S4: these values suggest that the models do not manage to recognize a great part of the lenses in the complete data set.

These trends can be interpreted by considering the impact of the inclusion of the borderline objects in the test sets. In particular, the training set of S2 mostly includes clear lenses and images of isolated non-lenses, not surrounded by other sources. When processing the borderline objects, the absence of clear arcs and rings, and more generally the faintness of the lensing features induce a growing fraction of lenses to be classified as non-lenses. Our results highlight the inability of our models to recover a considerable amount of lenses that are not similar to those in S2, leading to a dramatic decrease of more than $\sim$ 20% in the recall of the lenses from S2/S2 to S2/S3 and of the $\sim$ 30% from S2/S3 to S2/S4.

4.7. Training with multi-band images

The correct identification of GGSL events may benefit significantly from color information emerging from the analysis of multi-band data. Indeed, lenses and sources typically have different colors, due to their different spectral energy distributions (and redshifts). For example, the most common sources are star forming galaxies that appear bluer than the lenses, which on the contrary are often early-type passive galaxies. For these reasons, Metcalf et al. (2019) noticed that using multi-band images for training improves substantially the performance of the classifiers, when dealing with mock ground-based data. In this case, however, the morphological differences between lenses and non-lenses are mitigated by the worse spatial resolution achievable from the ground compared to space-based observations.

We evaluate the importance of color information for the identification of the borderline lenses in *Euclid*-like data by repeating the same training discussed so far, this time including the NIR images, also available from the BLF data. We change the architecture of our models to take into account the different sizes of the VIS and NISP images, as explained in Sect. 2, but otherwise keep the same setup as our previous experiments. We report the results of these tests in Table B.4.

By comparing these values to those in Table B.1, we do not observe a significant improvement of the models' performance when training with multi-band data. This is expected for the smaller selections, limited to the clearest lenses, whose correct identification through their characteristic morphology is relatively easy. Thus, in these cases, color information is expected to be less relevant. When looking at the broader selections, in which the morphology of the lenses is less clear, however, we might expect to see some improvement in the classification performance when feeding the models with color information. Surprisingly, we do not notice any significant variations of the metrics that quantify the model performance.

We interpret this result as follows. First, the wavelength range covered by the VIS instrument (see Table 1) does not include the wavelengths at which the color difference between the background and foreground galaxies is particularly evident, i.e. the blue wavelengths of the optical spectrum. Besides, the images in the NIR bands are characterized by lower resolution than those in the I_E band (also see Table 1), thus the morphological information is degraded in these channels.

5. Conclusions

The future observations of the *Euclid* telescope will offer the opportunity to increase the number of known galaxy-scale strong lenses by orders of magnitude, as long as potential candidates are efficiently identified. This will be feasible by using reliable methods for the analysis of large data sets and DL techniques are very promising for doing this.

In this work, we presented a detailed analysis of the performance of three CNN architectures in the identification of GGSL events. We did this by using a data set of forty thousand images simulated by the Bologna Lens Factory to mimic the data quality expected by the *Euclid* space mission. The classification was primarily based on the morphology of the systems, since we

mainly conducted our experiments with the images simulated in the I_E band, but we also evaluated the importance of color information by using multi-band images. We trained and tested our CNNs on four selections of the data set that gradually include a greater fraction of borderline objects, that are characterized by faint lensing features and will be more difficult to recognize. We evaluated the outcome of the classification by estimating the precision, recall and F1-score of the catalogs of obtained lenses.

We noticed that the morphological characteristics of the lenses included in the training set influence in a crucial way the ability of our CNNs to identify the lenses in a separate test set, whether they show clear or faint lensing features. In fact, we found that the inclusion of a large fraction of borderline images deteriorates the performance of our models, causing a decrease in the overall accuracy of the $\sim$ 10%, from $\sim$ 0.95 to $\sim$ 0.85 for the IncNet and VGG-like network, and even greater for the ResNet, that reaches an accuracy of $\sim$ 0.74. Moreover, we also found that it impacts the ability of our models to identify the most evident lenses, as they become under-represented in the training set.

These results prove that the identification of lenses with different morphologies 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 multiclass classification problem, distinguishing the clear and probable lenses from the possible and evident non-lenses. In this last case, however, the distinction between obvious and borderline objects should be further investigated and quantified.

We also retrain our models on the same selections of the data set, this time including a separate channel for processing the NIR images in addition to those in the I_E band, thus assessing how relevant the color information is for the identification of the borderline lenses. We do not find a significant improvement of the performance of any of our networks. We suggest that this might depend on a combination of two factors: firstly, the images in the I_E band have higher resolution than those in the NIR bands and secondly, the I_E band covers a wavelength range in which the color difference between lens and source galaxies might not be important (see Table 1).

Finally, we highlight that the three architectures retrieve catalogs with similar characteristics in terms of completeness and precision, when applied to the same selections of images. The only exception to this is the ResNet, whose accuracy on the full data set is $\sim$ 10% worse than that of the others. Because of the faster training and the easier implementation, however, the VGG-like network might be considered the best architecture among those tested.

In the future, we could improve our selection method by testing a combination of physical parameters to differentiate between faint and clear lenses, instead of using `n_pix_source`, that we have as a result of our simulations, but is not a property of the galaxies. It would also be useful to study whether there is a bias in the properties of the lenses found by our models, to better understand what kind of systems are most likely to be found or missed.

Acknowledgements. The authors acknowledge the Euclid Consortium, the European Space Agency, and a number of agencies and institutes that have supported the development of *Euclid*, in particular the Academy of Finland, the Agenzia Spaziale Italiana, the Belgian Science Policy, the Canadian Euclid Consortium, the French Centre National d'Etudes Spatiales, the Deutsches Zentrum für Luft- und Raumfahrt, the Danish Space Research Institute, the Fundação para a Ciência e a Tecnologia, the Ministerio de Ciencia e Innovación, the National Aeronautics and Space Administration, the National Astronomical Observatory of Japan, the Netherlandse Onderzoekschool Voor Astronomie, the Norwegian Space Agency, the Romanian Space Agency, the State Secretariat for Education, Research and Innovation (SERI) at the Swiss Space Office (SSO), and the United Kingdom Space Agency. A complete and detailed list is available on the *Euclid* web site (`http://www.euclid-ec.org`). We acknowledge support from the grants PRIN-MIUR 2017 WSCC32, PRIN-MIUR 2020 SKSTHZ and ASI n.2018-23-HH.0. MM acknowledges financial support from INAF "mainstream" 1.05.01.86.20: "Deep and wide view of galaxy clusters (P.I.: M. Nonino)" and INAF "main-stream" 1.05.01.86.31 "The deepest view of high-redshift galaxies and globular cluster precursors in the early Universe" (P.I.: E. Vanzella). This work has made use of CosmoHub. CosmoHub has been developed by the Port d'Informació Científica (PIC), maintained through a collaboration of the Institut de Física d'Altes Energies (IFAE) and the Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT) and the Institute of Space Sciences (CSIC & IEEC), and was partially funded by the "Plan Estatal de Investigación Científica y Técnica y de Innovación" program of the Spanish government.

References

Abadi, M., Barham, P., Chen, J., et al. 2016, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 265
Allison, J. R., Moss, V. A., Macquart, J. P., et al. 2017, MNRAS, 465, 4450
Angora, G., Rosati, P., Brescia, M., et al. 2020, A&A, 643, A177
Barnabè, M., Czoske, O., Koopmans, L. V. E., Treu, T., & Bolton, A. S. 2011, MNRAS, 415, 2215
Bengio, Y. 2009, Foundation and Trends in Machine Learning, vol. 2, 1
Bergamini, P., Rosati, P., Vanzella, E., et al. 2021, A&A, 645, A140
Bolton, A. S., Burles, S., Koopmans, L. V. E., Treu, T., & Moustakas, L. A. 2006, ApJ, 638, 703
Cabanac, R. A., Alard, C., Dantel-Fort, M., et al. 2007, A&A, 461, 813
Carretero, J., Tallada, P., Casals, J., et al. 2017, in Proceedings of the European Physical Society Conference on High Energy Physics. 5-12 July, 488
Chollet, F. 2015, keras, `https://github.com/fchollet/keras`
Coe, D., Benítez, N., Sánchez, S. F., et al. 2006, AJ, 132, 926
Collett, T. E. 2015, ApJ, 811, 20
de Jong, J. T. A., Verdoes Kleijn, G. A., Boxhoorn, D. R., et al. 2015, A&A, 582, A62
Desprez, G., Richard, J., Jauzac, M., et al. 2018, MNRAS, 479, 2630
Fluri, J., Kacprzak, T., Lucchi, A., et al. 2019, Phys. Rev. D, 100, 063514
Gavazzi, R., Marshall, P. J., Treu, T., & Sonnenfeld, A. 2014, ApJ, 785, 144
Gavazzi, R., Treu, T., Marshall, P. J., Brault, F., & Ruff, A. 2012, ApJ, 761, 170
Ghosh, A., Urry, C. M., Wang, Z., et al. 2020, ApJ, 895, 112
Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (The MIT Press)
Grillo, C. 2012, ApJ, 747, L15
Gupta, N. & Reichardt, C. L. 2020, ApJ, 900, 110
Gwyn, S. D. J. 2012, AJ, 143, 38
Hanley, J. V. & McNeil, B. 1982, Radiology, 143, 29
He, K., Zhang, X., Ren, S., & Sun, J. 2016, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770
He, Z., Er, X., Long, Q., et al. 2020, MNRAS, 497, 556
Hebb, D. O. 1949, The organization of behavior: A neuropsychological theory (Wiley)
Ho, M., Rau, M. M., Ntampaka, M., et al. 2019, ApJ, 887, 25
Impellizzeri, C. M. V., McKean, J. P., Castangia, P., et al. 2008, Nature, 456, 927
Ioffe, S. & Szegedy, C. 2015, in Proceedings of Machine Learning Research, Vol. 37, Proceedings of the 32nd International Conference on Machine Learning, ed. F. Bach & D. Blei (Lille, France: PMLR), 448
Jacobs, C., Collett, T., Glazebrook, K., et al. 2019a, ApJS, 243, 17
Jacobs, C., Collett, T., Glazebrook, K., et al. 2019b, MNRAS, 484, 5330
Jacobs, C., Glazebrook, K., Collett, T., More, A., & McCarthy, C. 2017, MNRAS, 471, 167
Jauzac, M., Klein, B., Kneib, J.-P., et al. 2021, MNRAS, 508, 1206
Kingma, D. P. & Ba, J. 2017, Adam: A Method for Stochastic Optimization, arXiv:1412.6980
Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, arXiv:1110.3193
LeCun, Y., Boser, B., Denker, J. S., et al. 1989, Neural Computation, 1, 541
Li, R., Napolitano, N. R., Tortora, C., et al. 2020, ApJ, 899, 30
Liew-Cain, C. L., Kawata, D., Sánchez-Blázquez, P., Ferreras, I., & Symeonidis, M. 2021, MNRAS, 502, 1355
Lin, M., Chen, Q., & Yan, S. 2013, arXiv:1312.4400
LSST Science Collaboration, Abell, P. A., Allison, J., et al. 2009, arXiv:0912.0201
Maturi, M., Mizera, S., & Seidel, G. 2014, A&A, 567, A111
McCulloch, W. & Pitts, W. 1943, Bulletin of Mathematical Biophysics, 5, 115
Meneghetti, M., Davoli, G., Bergamini, P., et al. 2020, Science, 369, 1347
Meneghetti, M., Melchior, P., Grazian, A., et al. 2008, A&A, 482, 403
Meneghetti, M., Ragagnin, A., Borgani, S., et al. 2022, A&A, 668, A188
Meneghetti, M., Rasia, E., Merten, J., et al. 2010, A&A, 514, A93

Merten, J., Giocoli, C., Baldi, M., et al. 2019, MNRAS, 487, 104
Metcalf, R. B., Meneghetti, M., Avestruz, C., et al. 2019, A&A, 625, A119
Metcalf, R. B. & Petkova, M. 2014, MNRAS, 445, 1942
Minor, Q., Gad-Nasr, S., Kaplinghat, M., & Vegetti, S. 2021, MNRAS, 507, 1662
Myers, S. T., Jackson, N. J., Browne, I. W. A., et al. 2003, MNRAS, 341, 1
Napolitano, N. R., Li, R., Spiniello, C., et al. 2020, ApJ, 904, L31
Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, ApJ, 462, 563
Negrello, M., Amber, S., Amvrosiadis, A., et al. 2017, MNRAS, 465, 3558
Negrello, M., Hopwood, R., De Zotti, G., et al. 2010, Science, 330, 800
Nightingale, J. W., Massey, R. J., Harvey, D. R., et al. 2019, MNRAS, 489, 2049
Oguri, M., Rusu, C. E., & Falco, E. E. 2014, MNRAS, 439, 2494
Pan, S., Liu, M., Forero-Romero, J., et al. 2020, Science China Physics, Mechanics, and Astronomy, 63, 110412
Pasquet, J., Bertin, E., Treyer, M., Arnouts, S., & Fouchez, D. 2019, A&A, 621, A26
Petkova, M., Metcalf, R. B., & Giocoli, C. 2014, MNRAS, 445, 1954
Petrillo, C. E., Tortora, C., Chatterjee, S., et al. 2019a, MNRAS, 482, 807
Petrillo, C. E., Tortora, C., Vernardos, G., et al. 2019b, MNRAS, 484, 3879
Reddi, S. J., Kale, S., & S., K. 2019, On the Convergence of Adam and Beyond, arXiv:1904.09237
Rojas, K., Savary, E., Clément, B., et al. 2022, A&A, 668, A73
Rumelhart, D., Hinton, G. E., & Williams, R. J. 1986, Nature, 323, 533
Savary, E., Rojas, K., Maus, M., et al. 2022, A&A, 666, A1
Schuldt, S., Chirivì, G., Suyu, S. H., et al. 2019, A&A, 631, A40
Seidel, G. & Bartelmann, M. 2007, A&A, 472, 341
Shuntov, M., Pasquet, J., Arnouts, S., et al. 2020, A&A, 636, A90
Simonyan, K. & Zisserman, A. 2015, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings
Sonnenfeld, A., Chan, J. H. H., Shu, Y., et al. 2018, PASJ, 70, S29
Stacey, H. R., McKean, J. P., Robertson, N. C., et al. 2018, MNRAS, 476, 5075
Stehman, S. V. 1997, Remote Sensing of Environment, 62, 77
Suyu, S. H., Hensel, S. W., McKean, J. P., et al. 2012, ApJ, 750, 10
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818
Szegedy, C., Wei Liu, Yangqing Jia, et al. 2015, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1
Tallada, P., Carretero, J., Casals, J., et al. 2020, Astronomy and Computing, 32, 100391
The Dark Energy Survey Collaboration. 2005, arXiv:0510346
Treu, T. & Koopmans, L. V. E. 2004, ApJ, 611, 739
Tu, H., Limousin, M., Fort, B., et al. 2008, MNRAS, 386, 1169
Vegetti, S., Despali, G., Lovell, M. R., & Enzi, W. 2018, MNRAS, 481, 3661
Wu, J. F. & Boada, S. 2019, MNRAS, 484, 4683
Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. 2017, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5987
Xu, B., Wang, N., Chen, T., & Li, M. 2015, arXiv:1505.00853
Zhou, Y.-T. & Chellappa, R. 1988, in IEEE 1988 International Conference on Neural Networks, Vol. 2, 71
Zhu, X.-P., Dai, J.-M., Bian, C.-J., et al. 2019, Ap&SS, 364, 55

Appendix A: Network architectures

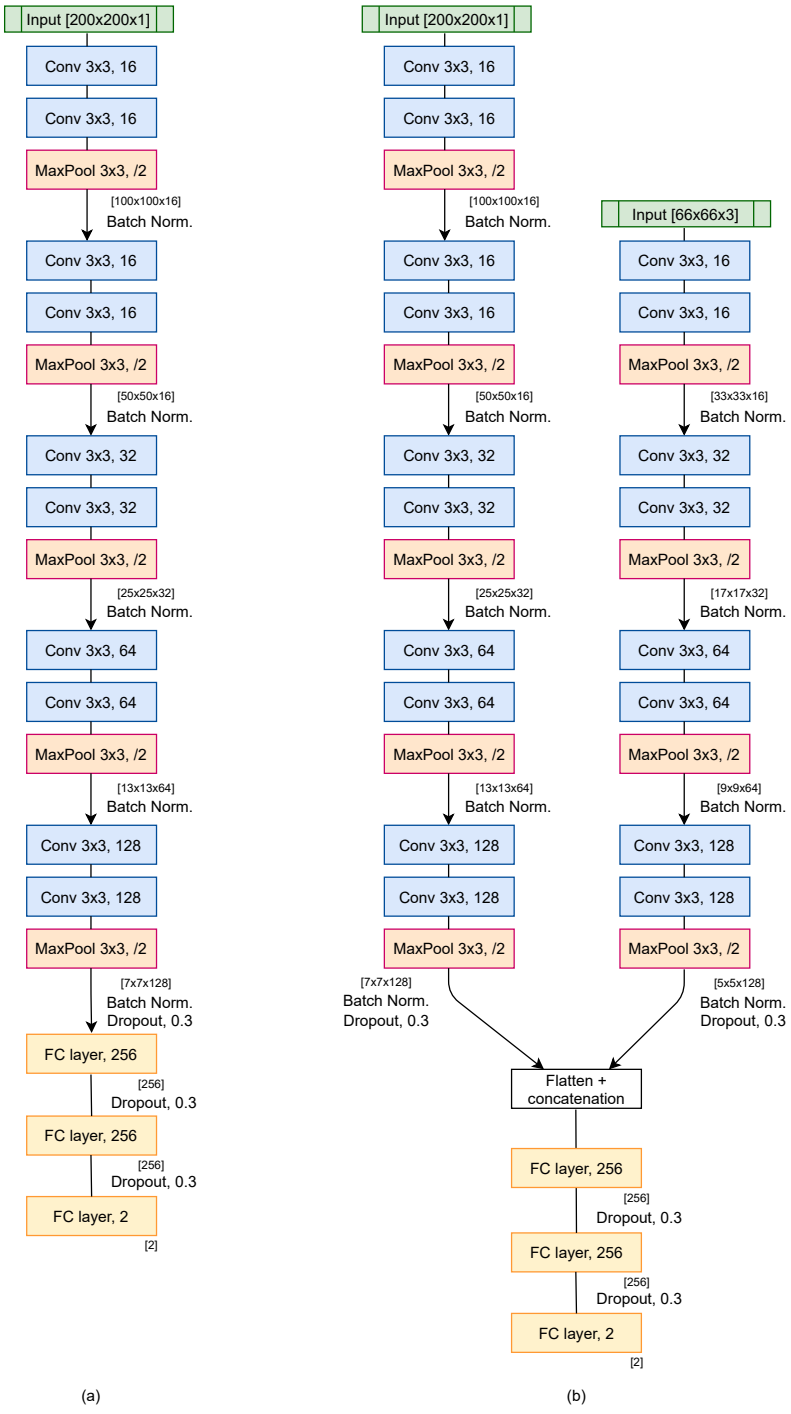

Fig. A.1. VGG-like network configurations tested on the VIS images (a) and on the multi-band images (b). We report the dimension (D) and amount (F) of the filters used in the convolutional layers in the format D×D, F. We also indicate the pooling region (R) and the strides (S) in the pooling layers in the format R×R, /S. The numbers in the square brackets indicate the dimension and amount of the feature maps obtained as output of the layers in the format [D×D×F].

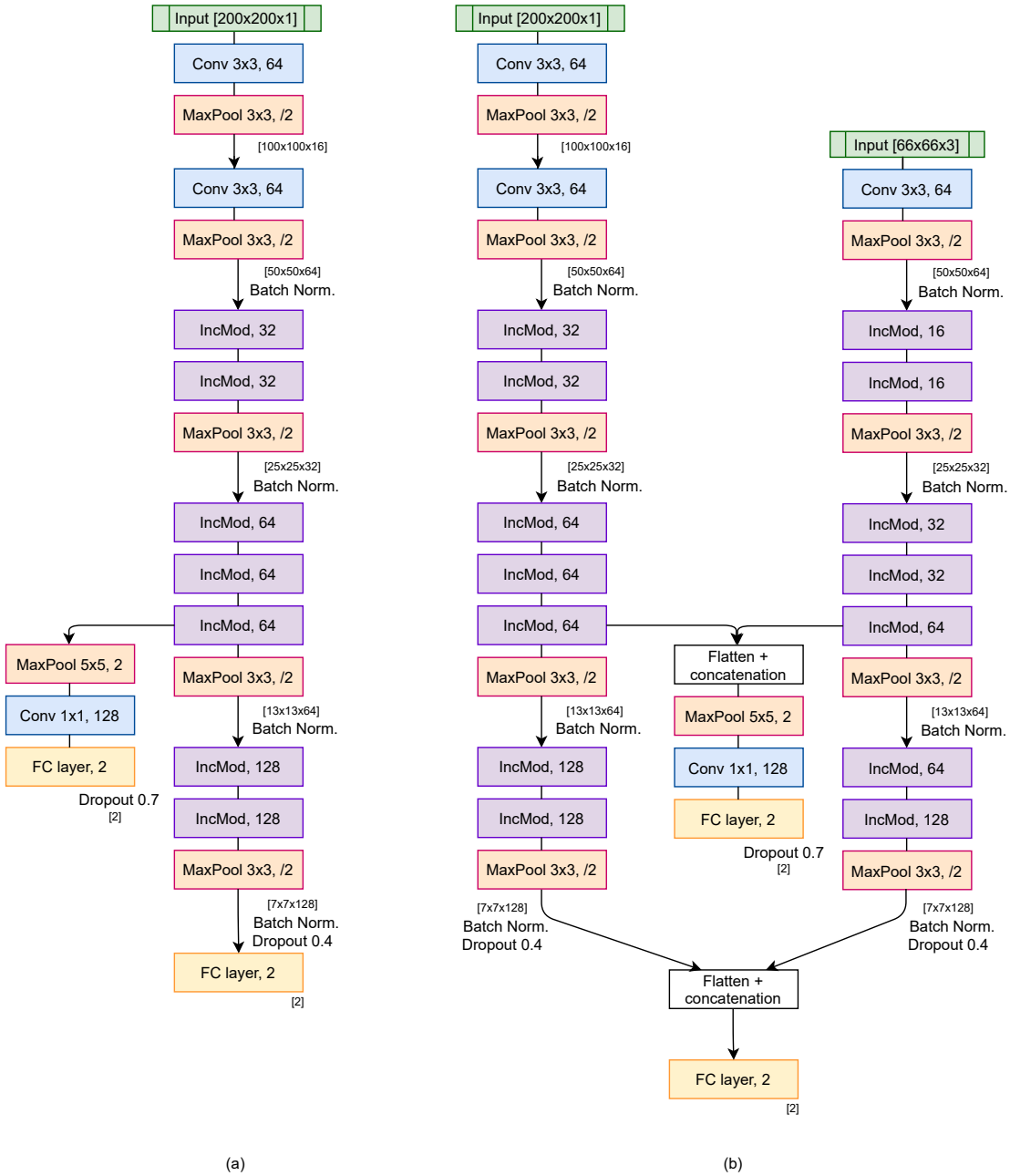

Fig. A.2. Inception Network configurations tested on the VIS images (a) and on the multi-band images (b). These diagrams use the same notation as those in Fig. A.1.

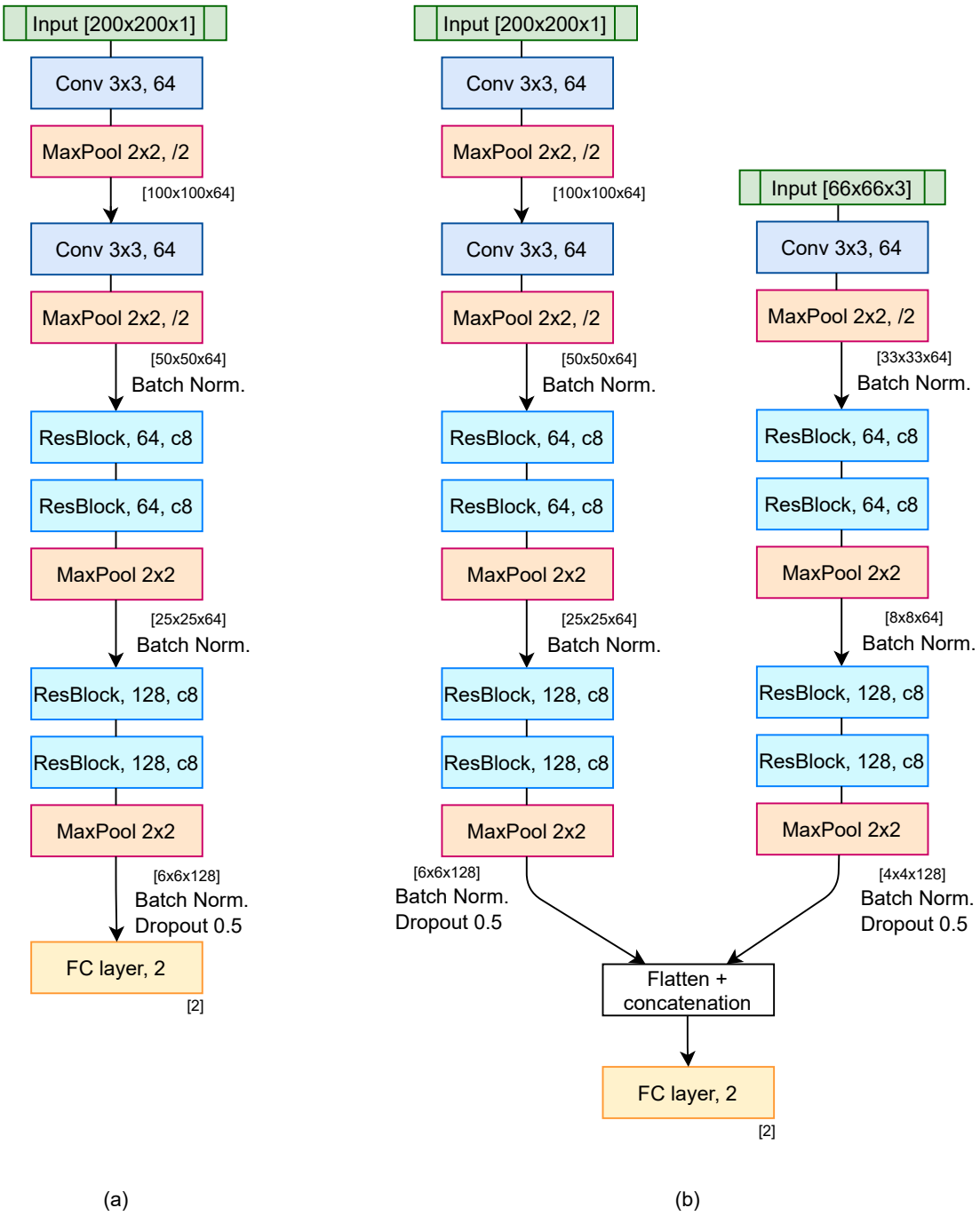

Fig. A.3. Residual Network configurations tested on the VIS images (a) and on the multi-band images (b). These diagrams use the same notation as those in Fig. A.1.

Appendix B: Tables

Table B.1. Summary of the performance of the VGG-like network, the IncNet and the ResNet in the classification of the objects of the four selections of images in the I_E band. The precision, recall and F1-score are evaluated on the class of the non-lenses (0) and of the lenses (1) separately, while accuracy and AUC are global quantities.

VGG-like network								
	S1		S2		S3		S4	
Class	0	1	0	1	0	1	0	1
Precision	0.95	0.98	0.94	0.97	0.92	0.94	0.79	0.89
Recall	0.98	0.94	0.98	0.94	0.94	0.92	0.90	0.77
F1-score	0.96	0.96	0.96	0.96	0.93	0.93	0.84	0.83
Accuracy	0.96		0.96		0.93		0.84	
AUC	0.77		0.58		0.88		0.57	
Inception Network								
	S1		S2		S3		S4	
Class	0	1	0	1	0	1	0	1
Precision	0.97	1.0	0.97	0.96	0.94	0.93	0.84	0.90
Recall	1.0	0.96	0.96	0.97	0.93	0.94	0.91	0.83
F1-score	0.98	0.98	0.96	0.96	0.93	0.94	0.87	0.86
Accuracy	0.98		0.96		0.94		0.87	
AUC	0.92		0.88		0.90		0.81	
Residual Network								
	S1		S2		S3		S4	
Class	0	1	0	1	0	1	0	1
Precision	0.93	0.97	0.90	0.92	0.86	0.89	0.71	0.84
Recall	0.97	0.92	0.92	0.89	0.89	0.85	0.87	0.66
F1-score	0.95	0.94	0.91	0.91	0.88	0.87	0.78	0.74
Accuracy	0.95		0.91		0.87		0.76	
AUC	0.81		0.85		0.79		0.70	

Table B.2. Comparison between the metrics of tests on the selection S2 with the models trained on S2 (top) and on S4 (bottom). Class 0 refers to the non-lenses, while class 1 refers to the lenses.

S2/S2						
	VGG-like network		**Inception Network**		**Residual Network**	
Class	0	1	0	1	0	1
Precision	0.94	0.97	0.97	0.96	0.90	0.92
Recall	0.98	0.94	0.96	0.97	0.92	0.89
F1-score	0.96	0.96	0.96	0.96	0.91	0.91
Accuracy	0.96		0.96		0.91	
AUC	0.58		0.88		0.85	
S4/S2						
	VGG-like network		**Inception Network**		**Residual Network**	
Class	0	1	0	1	0	1
Precision	0.96	0.74	0.99	0.77	0.95	0.67
Recall	0.89	0.91	0.90	0.98	0.85	0.89
F1-score	0.93	0.82	0.94	0.86	0.90	0.76
Accuracy	0.89		0.92		0.86	
AUC	0.51		0.88		0.75	

Table B.3. Summary of the performance of the VGG-like network, the Inception Network and the Residual Network, trained on the selection S2, in the classification of the objects that are part of the selections S3, S4. 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				**Inception Network**				**Residual Network**			
	S2/S3		S2/S4		S2/S3		S2/S4		S2/S3		S2/S4	
Class	0	1	0	1	0	1	0	1	0	1	0	1
Precision	0.77	0.97	0.62	0.95	0.82	0.96	0.65	0.93	0.75	0.88	0.64	0.85
Recall	0.98	0.68	0.98	0.33	0.97	0.76	0.97	0.42	0.92	0.67	0.94	0.40
F1-score	0.86	0.80	0.76	0.48	0.89	0.85	0.78	0.58	0.83	0.76	0.76	0.55
Accuracy	0.83		0.68		0.87		0.71		0.80		0.69	
AUC	0.57		0.52		0.81		0.7		0.78		0.65	

Table B.4. Same as in Table B.1, but using images in the VIS and NISP bands.

VGG-like network								
	S1		S2		S3		S4	
Class	0	1	0	1	0	1	0	1
Precision	0.99	0.97	0.98	0.97	0.91	0.96	0.81	0.91
Recall	0.97	0.99	0.97	0.98	0.96	0.91	0.92	0.79
F1-score	0.98	0.98	0.98	0.98	0.94	0.93	0.86	0.84
Accuracy	0.98		0.98		0.93		0.85	
AUC	0.65		0.87		0.67		0.62	
Inception Network								
	S1		S2		S3		S4	
Class	0	1	0	1	0	1	0	1
Precision	0.98	0.96	0.97	0.98	0.96	0.96	0.87	0.91
Recall	0.96	0.98	0.98	0.96	0.96	0.96	0.91	0.87
F1-score	0.97	0.97	0.97	0.97	0.96	0.96	0.89	0.89
Accuracy	0.97		0.97		0.96		0.89	
AUC	0.77		0.9		0.92		0.84	
Residual Network								
	S1		S2		S3		S4	
Class	0	1	0	1	0	1	0	1
Precision	0.96	0.95	0.92	0.94	0.86	0.92	0.74	0.85
Recall	0.94	0.96	0.94	0.92	0.92	0.87	0.87	0.71
F1-score	0.95	0.95	0.93	0.93	0.90	0.89	0.80	0.77
Accuracy	0.95		0.93		0.90		0.78	
AUC	0.81		0.88		0.81		0.72	