Detection of Strong Gravitational Lenses with Convolutional Neural Networks

Machine Learning (CS-433), Department of Computer and Communication Sciences,

École Polytechnique Fédérale de Lausanne

Daniel Forero Sanchez
daniel.forerosanchez@epfl.ch

Andrei Variu
andrei.variu@epfl.ch

Yaroslav Ilichenko
yaroslav.ilichenko@epfl.ch

Abstract—**The search for strong gravitational lenses is very relevant to modern cosmology where future large-scale surveys such as Euclid will expand our knowledge about the universe. In this paper, we rely on simulated Euclid data for lensing systems and train two neural network architectures to classify images into lenses and non-lenses. We obtain accuracies of 0.708 and 0.687 for the LASTRO and the Residual neural networks, respectively. We also find that objects with clear arch-like features such as spiral galaxies tend to confuse the networks, as well as pictures with scattered objects.**

I. Introduction

Gravitational Lensing is a phenomenon that occurs due to the deflection of light in a gravitational field of a massive object. The name comes from the fact that this effect is observationally similar to light passing through an optical lens. Moreover, the source of the gravitational field is called gravitational lens or simply lens.

Strong Gravitational Lensing (hereon SL) is a more specific case of this phenomenon. It can occur when a light source (such as a quasar or a galaxy) is behind a galaxy or a cluster of galaxies (the lens). Depending on the position of the source relative to the lens, one can observe multiple images of the source around the lens, luminous arcs and rings (also called Einstein Rings), and often the images are magnified.

Lensing systems are scarce and difficult to detect [6, 7], but very valuable for cosmological studies. Rough estimations show that as few as one in 100 thousands observed galaxies will show strong lensing features, making their detection a challenge. Consequently, large astronomical surveys such as Euclid[1] are required to observe more lenses.

The Euclid survey is set to start in 2022 with the launch of the satellite and it should map 15 000 of the sky, thus yielding huge amounts of data. So far, lens detection has mainly been done at small scale and by human eye inspection (such as the Citizen Science project launched by Hyper-Suprime Cam survey[2]); however, the volume of Euclid's data shows the necessity of a tool to detect such systems on a much larger scale. In order to develop detection tools for this survey, challenges such as Gravitational Lens Finding Challenge[3] are organized based on Euclid-like simulations.

In this report, we present the results of our search of strong gravitational lenses for the previously presented challenge, using two architectures of Convolutional Neural Networks (CNNs). In section 2, we outline the data set used to train the networks. In section 3, we present the preprocessing techniques applied on our data set. In section 4, we describe the two CNN architectures. In section 5, we discuss the results and in the last section we conclude.

II. Data set

The proposed challenge is based on lensing simulations that mimic the data that will be gathered by Euclid. The provided catalog did not contain labels for lenses and non-lenses, but candidates without any added sources (`n_sources==0`) were flagged as non-lenses and the rest were considered lenses. We did not exactly follow the suggestion of the official instructions[4] (i.e. lenses = objects with effective magnitude $\mu_{\rm eff} > 2$ and non-lenses = objects with `n_sources==0` or $\mu_{\rm eff} < 1.2$) due to a misunderstanding and we have realized this issue too late to perform all tests from beginning. The used data set consists of 89 995 lenses and 10 005 non-lenses, but the correct one should have 5000 more non-lenses and and about 35 thousands less lenses. However, we have performed brief tests to verify the impact of this error.

We preprocessed (see section III) the whole data set and we split it in a train and validation set (≈ 67000 objects for the training set and the rest are in the validation set), but we favored a higher number of small-scale tests over fewer large-scale tests. Therefore we used 6000 images per epoch for training and 2955 per epoch for validation. Each image (`ID`) contains 4 spectral bands: one in the visible (VIS) and 3 near-infrared ones (NIR), with resolutions of 200×200 and 60×60 respectively.

Another aspect is that at each epoch the set of 6000 objects can change, because for each epoch there are 600 steps and for each step a random batch of 10 images is selected from the entire training set. Theoretically, this can actually

[1] https://www.euclid-ec.org/

[2] https://www.zooniverse.org/projects/aprajita/space-warps-hsc

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

[4] http://metcalf1.difa.unibo.it/DATA3/evaluation.pdf

be advantageous by reducing the chance of overfit and by reducing the effect of the wrong definition of lenses.

It is worth noting that the catalog was recently updated and we did *not* update ours. However, the update was due to duplicate `IDs` in the catalog, which were removed by us from the begining.
A more detailed description of the data set and the lens catalog can be found in the challenge's webpage[5].

III. Preprocessing

In order to preprocess the data, we sought to enhance the signal as well as to cast the pixel values into a range easy to manage by the computer. The initial images (top row in figures 5 and 6, in the appendix) had a very weak signal and the pixel values were of the order of 10^{-9} to 10^{-12}. To enhance the signal, we attempted two ways to preprocess the data. On one hand, we performed a log-stretch and then a normalization to the interval $[0, 1]$ (middle row in the same figures). On the other hand, (bottom row), we initially clipped the figure before the above transformations in the following manner. The interval limits for the 0.25% and 99.75% percentiles (`vmin`, `vmax` respectively) were extracted for each band. Each was then clipped at -0.7 `vmin` below and at `vmax`. The resulting array was normalized by `vmax` and then log-stretched. The overall transformation got rid of most of the noise (see figure 5 in appendix) and results in a stronger signal. Additionally, in both cases, all provided color bands were saved in one multispectral image , where the pictures in the infrared bands were upsampled to the size of the ones in the visible band to avoid losing the visible band's resolution.

IV. Architectures

We implemented two neural network architectures: a Residual Network (ResNet) [3, 4] and an architecture inspired by the submission of the Laboratory of Astrophysics (LASTRO) to the first challenge [5, 7], which we refer to as LASTRO. Both networks were fed with batches of 10 images containing half lenses and half non-lenses to remove the bias from the data set and both NNs use the binary cross-entropy loss. In addition to preprocessing, we tested the influence of data augmentation on both networks. More specifically, we have used random horizontal and vertical flips of images to augment the data.

A. ResNet

The ResNet used in this project was inspired from reference [3] and from the Keras official website[6]. The complete architecture is modular, as one can observe in figure 8 (left panel). A block of five layers – 2D convolution, batch normalization, activation, 2D convolution and batch normalization – is repeated nine times, always being preceded by an activation layer and succeeded by an addition layer. The addition layer adds the input of a block (which is called residual) with its output. After the last addition layer, an activation, an average 2D pooling, a drop-out, a flatten and a dense layers are built. The drop-out layer had not been initially part of the architecture, however it has been introduced because the ResNet has started to overfit when the training sample has been increased from 1000 objects to 6000.

Using a residual, the depth of the NN can be substantially increased [3], thus the current architecture has 75 layers. The kernel size of the convolution layer is three pixels, except for the two convolution layers applied on the residuals (the two layers depicted with rotated written names) whose kernel sizes are one pixel. The pool size of the average 2D pool layers is 8 pixels and the activation function is the Rectified Linear Unit (ReLU), except the one used after the dense layer which was a Sigmoid function. This architecture performed very well on CIFAR-10[7] training data set according to reference [3]. Moreover, residual nets achieved one of the highest accuracy in the first Gravitational Lens Finding Challenge [7].

B. LASTRO

The network architecture, inspired by references [7, 5], consists of a 25-layer network shown in figure 8 (right panel). Our final architecture is composed by convolutional layers with kernel sizes of 5 for the initial ones and 3 for the following ones, as well as a pool window of width 3 for the max-pooling layers succeeding the convolutions. The number of filters was doubled in each convolutional layer pair. We used ReLU activation in the convolutional layers and sigmoid in the output dense one. For this architecture we tested the impact of using the *spatial dropout* [2] instead of (standard) *dropout* [1]. The motivation is that adjacent pixels in images are highly correlated so masking individual pixels through dropout is not expected to have a big effect, while masking entire feature maps using the spatial dropout could potentially have a higher impact. However, in the case of fully connected layers, we keep the standard dropout.

V. Results

A. Metrics

In this work, we evaluate the models using the Accuracy, Receiving Operating Characteristic curve (ROC) and the Area Under (ROC) Curve, AUC. The ROC is the curve obtained by plotting the True Positive Rate (TPR) versus the False Positive Rate (FPR). These are defined as

$$\mathrm{TPR} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}, \qquad (1)$$

$$\mathrm{FPR} = \frac{\text{False Positives}}{\text{True Negatives} + \text{False Positives}}. \qquad (2)$$

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

[6] https://keras.io/examples/cifar10_resnet/

[7] https://www.cs.toronto.edu/~kriz/cifar.html

A perfect classifier would yield $\text{TPR} = 1$ and $\text{FPR} = 0$ leading to $\text{AUC} = 1$, while a random classifier would have $\text{AUC} = 0.5$.

B. ResNet

We have started training the ResNet using only the visible band and no augmentation, with the non clipped data set. The NN has learnt up to the 80th epoch and then it slowly overfitted, yielding a maximum precision of ≈ 0.65. The clipped data set has yielded lower precision and lower AUC, thus it was not used for this architecture, see figure 1.

Furthermore, using the random horizontal and vertical flips to augment the data, we have obtained the best results for ResNet, i.e. an accuracy of 0.687 and an AUC of 0.746 (see figures 2 and 4). Moreover, the NN was continuously learning until the 130th epoch, thus we have increased the number of epochs up to 230 for the same configuration in order to further train and test it . Unfortunately, no improvement has been observed and the accuracy quickly plateaued after the 130th epoch. We have also attempted to train the same configuration on a sample with 12 thousands objects and no improvement has been observed. In the last configuration, we have included the three NIR bands, which has drastically decreased the performance of the ResNet, as one can observe in figure 3.

C. LASTRO

The network presented in [5] was modified to suit our needs. While the architecture remained roughly the same, hyperparameters such as the kernel size and the pool window width were tuned. Various tests varying such parameters were made where the models did not learn enough (or fast enough) and they were no better than the random classifier after 20 to 50 epochs. However, the NN has started learning using kernel sizes of 5 for the first layer and 3 for the subsequent ones and a pool window of 3 pixels wide. We tested the differences in both kinds of preprocessing, the ROC obtained for the tests is shown in figure 1. After the 20 epochs in this particular test, the ROC with clip had better metrics, which led us to use the clipped data set with this network.

Subsequently, the impact of augmentation was tested. The results are shown in figure 2. Notice how the ROC for the network that uses augmentation is slightly better, with an $\text{AUC} = 0.59$, 11% better than the case without random flips ($\text{AUC} = 0.53$). Even if the inclusion of augmented data is a well known strategy in the training of networks, we did not expect to see a significant difference given that the data set is mostly composed of pictures with a bright galaxy in the center and companions/lensed sources scattered around; which could be regarded as roughly symmetric under the dihedral group. However, future work could evaluate the impact of not only vertical and horizontal flips but of the whole group. Nonetheless, this improvement could still be caused due to the small training set we used.

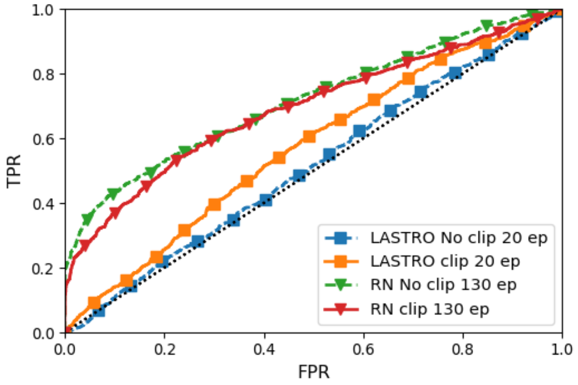

Figure 1. ROC of the test for the impact of the clip step the architectures.

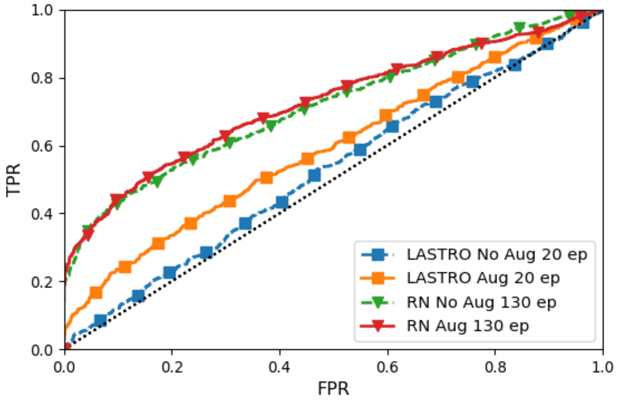

Figure 2. ROC of the test for the impact of augmentation on the architectures.

There was, in principle, no good justification to use the NIR bands of the data, given their poorer resolution and the extra work it meant to input a multispectral image to the network. However, for this network, the inclusion of the 3 NIR bands implies an increase in accuracy of 7% and an increase in AUC from $\text{AUC} = 0.7$ to $\text{AUC} = 0.75$. The results for the test with and without the NIR bands are shown in figure 3.

Having tuned these hyperparameters, we decided to use augmentation, clipping and the NIR bands for LASTRO's network. It was then initially trained for up to 230 epochs and early stopping. The network using standard dropout did stop earlier in the training than the one using spatial dropout. This shows a more constant improvement of the latter. However, both networks stopped before the 230 epochs. The learning rate was then changed to 10^{-4} for up to 20 epochs and then changed again to 10^{-5} and left to run until a plateau was reached. In the end, the network with spatial dropout

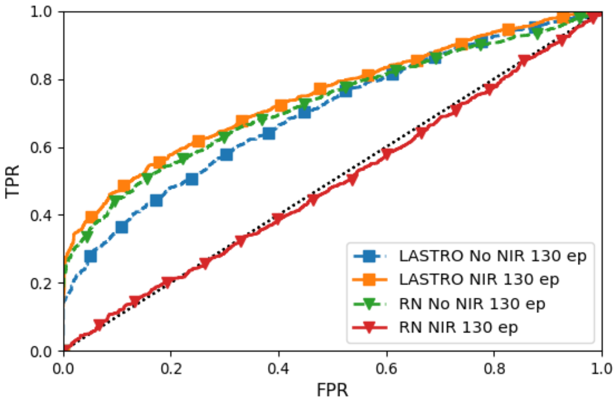

Figure 3. ROC of the test for the impact of the inclusion of NIR bands on the architectures.

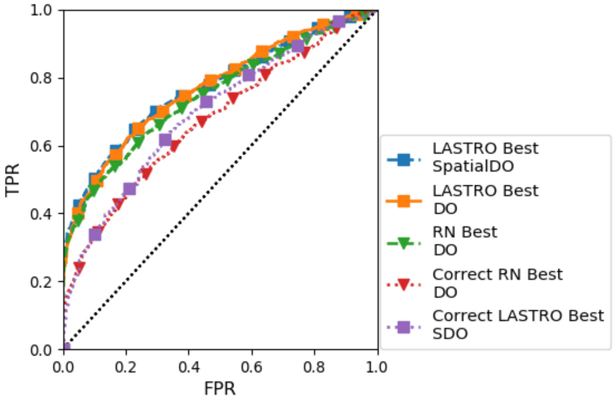

Figure 4. ROC of out best models. Dashed lines show our initial results while dotted lines show the ones obtained after shortly retraining the networks with the recomputed (marked as "correct") labels. Triangles mark the ResNet architecture and squares, LASTRO.

trained for 255 epochs and the one with usual dropout for 199. Dropout rate was kept at 0.2 throughout the networks. Given that the training followed the same steps, we can see that spatial dropout seems to allow the network to train for longer. Nevertheless, the end results as seen in figure 4 show that the accuracy and AUC are comparable, differing in 0.1% in AUC and 1.4% in accuracy. We use the model with spatial dropout, that yielded an accuracy of $\mathrm{ACC} = 0.71$ and an $\mathrm{AUC} = 0.77$ as our best model.

In order to better understand the networks, we extracted some of the false positives for each of our architectures which are shown in the images in the appendix (figure 7). Images composed by a galaxy surrounded by small companions look like the dominant cause for false positives. However, it is also evident that large spiral galaxies with very distinct arms tend to confuse the network as well. Our results are summarized in table I

Table I
METRICS FOR THE MODELS THAT SHOWED BEST PERFORMANCE.

Network	ACC	AUC
LASTRO (SpatialDropout)	0.708	0.77
LASTRO (Dropout)	0.698	0.769
ResNet	0.687	0.746
Corrected LASTRO (SpatialDropout)	0.633	0.698
Corrected ResNet	0.576	0.673

The results presented previously are all obtained with the different definition of lenses and non-lenses. Therefore, we have performed rapid tests on the correct catalog of lenses and non-lenses to have an estimation of the impact. The tests were performed with the hyperparameters that previously yielded the best results, but the number of steps and epochs were different (keeping the same batch size of 10 images). For ResNet, we have set 500 steps per epoch and only 50 epochs (at each epoch, the train set could be different) and for LASTRO we have used exactly 5126 images in the training sample and 95 epochs (each epoch had the same train set). Consequently, these results (see figure 4 and table I) cannot be directly compared to the previous ones. However, one can observe that the accuracy and AUC are not far off compared to the previous case. While LASTRO started overfitting – which might be caused by the fixed set of 5126 images – the learning curve of ResNet showed similar trend to the case with the different definition of lenses. The last observation on ResNet might also show that by changing the training at each epoch, the results could more robust to changes.

VI. Conclusion

The results obtained with both of our approaches show maximum accuracy of $\sim 70\%$. This is still not enough to comfortably classify a real data set, but it shows that the networks have the capacity of learning from this sample. The networks should be improved and further testing is necessary. The current run-time for our models was unexpectedly large, often surpassing the $10\,\mathrm{h}$ mark, probably due to the limited GPU time available in Lesta, the cluster of the Observatory of Geneva, and poor optimization of our code. This made testing and tuning various hyperparameters time-consuming, while the size of the data set made it difficult and still time-expensive to switch machines. A deeper architecture could have the complexity necessary to better fit the data, at the cost of training time and risk of overfitting. ResNet is designed precisely to be able to create deep models while still gaining accuracy, so further testing in this direction is promising. We also observed that objects that present clear arch-like structures, such as spiral galaxies, tend to confuse the networks.

Furthermore, we performed a test using the correct definitions for lenses and non-lenses to asses the impact of this change and observed that the results might not be drastically different. However, further training of the NNs should be done on the correct catalog.

References

[1] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.

[2] Jonathan Tompson et al. “Efficient object localization using Convolutional Networks”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 07-12-June-2015 (2015), pp. 648–656. ISSN: 10636919. DOI: 10.1109/CVPR.2015.7298664.

[3] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (2016), pp. 770–778. ISSN: 10636919. DOI: 10.1109/CVPR.2016.90.

[4] Kaiming He et al. “Identity mappings in deep residual networks”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9908 LNCS (2016), pp. 630–645. ISSN: 16113349. DOI: 10.1007/978-3-319-46493-0_38.

[5] C. Schaefer et al. “Deep Convolutional Neural Networks as strong gravitational lens detectors”. In: (May 2017). DOI: 10.1051/0004-6361/201731201.

[6] C. Jacobs et al. “An Extended Catalog of Galaxy–Galaxy Strong Gravitational Lenses Discovered in DES Using Convolutional Neural Networks”. In: *The Astrophysical Journal Supplement Series* 243.1 (2019), p. 17. ISSN: 0067-0049. DOI: 10.3847/1538-4365/ab26b6.

[7] R. B. Metcalf et al. “The strong gravitational lens finding challenge”. In: *Astronomy and Astrophysics* 625 (2019). ISSN: 14320746. DOI: 10.1051/0004-6361/201832797.

Appendix

A. Effect of preprocessing on the images

B. False Positives

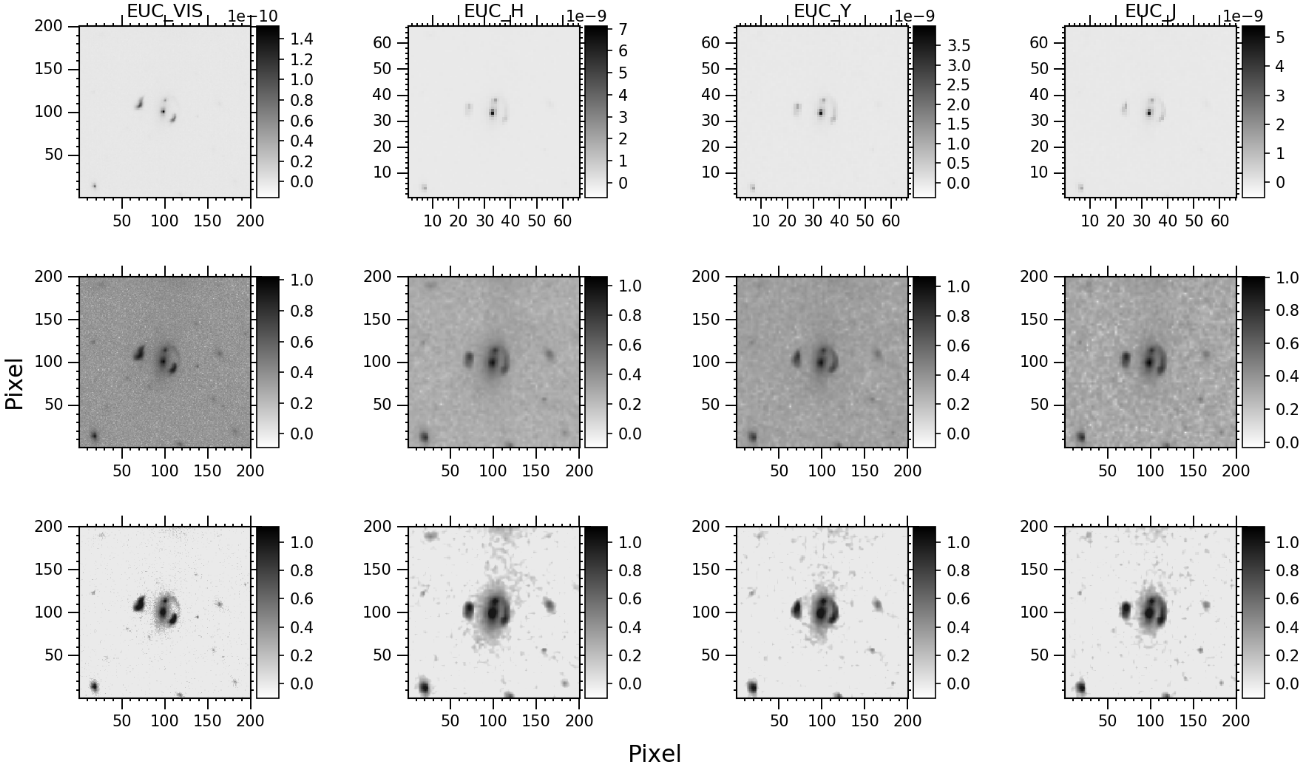

Figure 5.   Preprocessing results for a lens example (`ID 200001`) to evaluate the impact of the clip step in the preprocessing. Top row shows each band without preprocessing, middle row shows the bands with a log stretch and rescaling to the $[0, 1]$ interval (from Astropy's visualization module). Bottom row shows the images when clipping the noise.

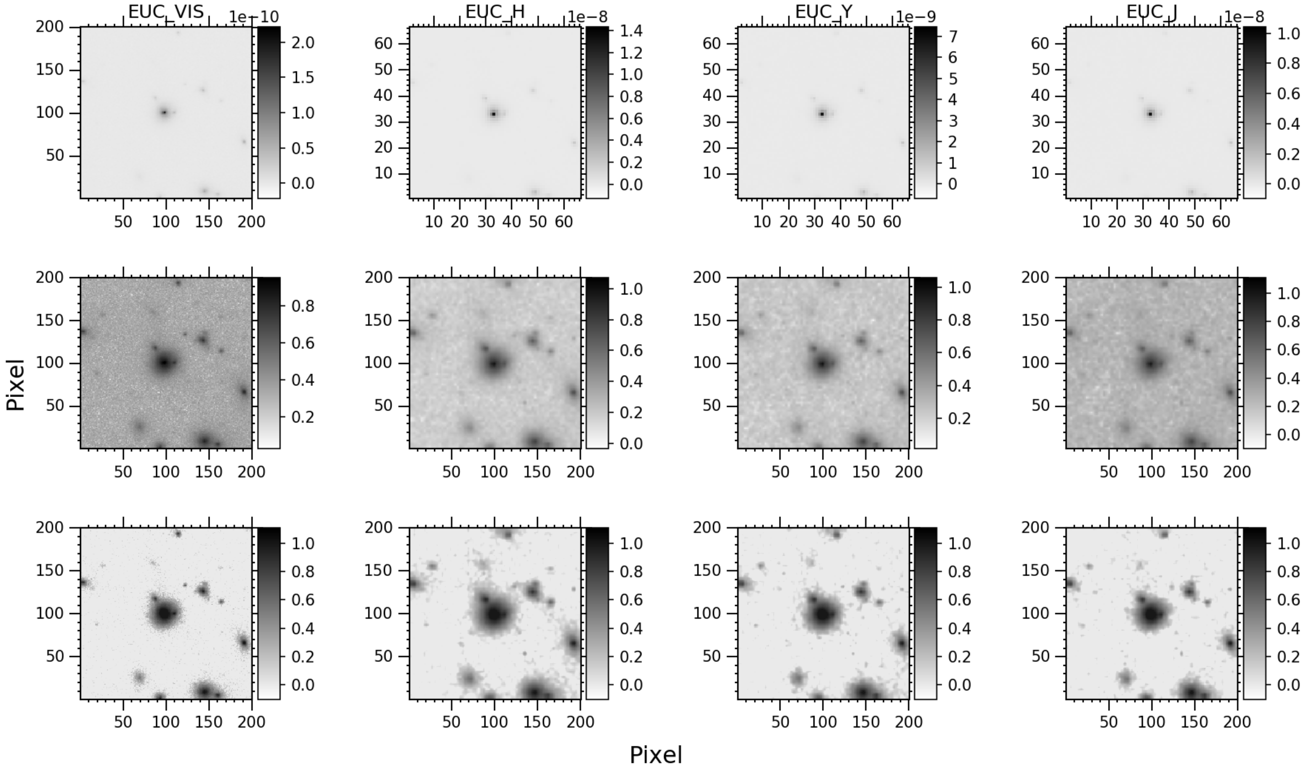

Figure 6. Preprocessing results for a non-lens example (`ID 200003`) to evaluate the impact of the clip step in the preprocessing. Top row shows each band without preprocessing, middle row shows the bands with a log stretch and rescaling to the $[0, 1]$ interval (from Astropy's visualization module). Bottom row shows the images when clipping the noise.

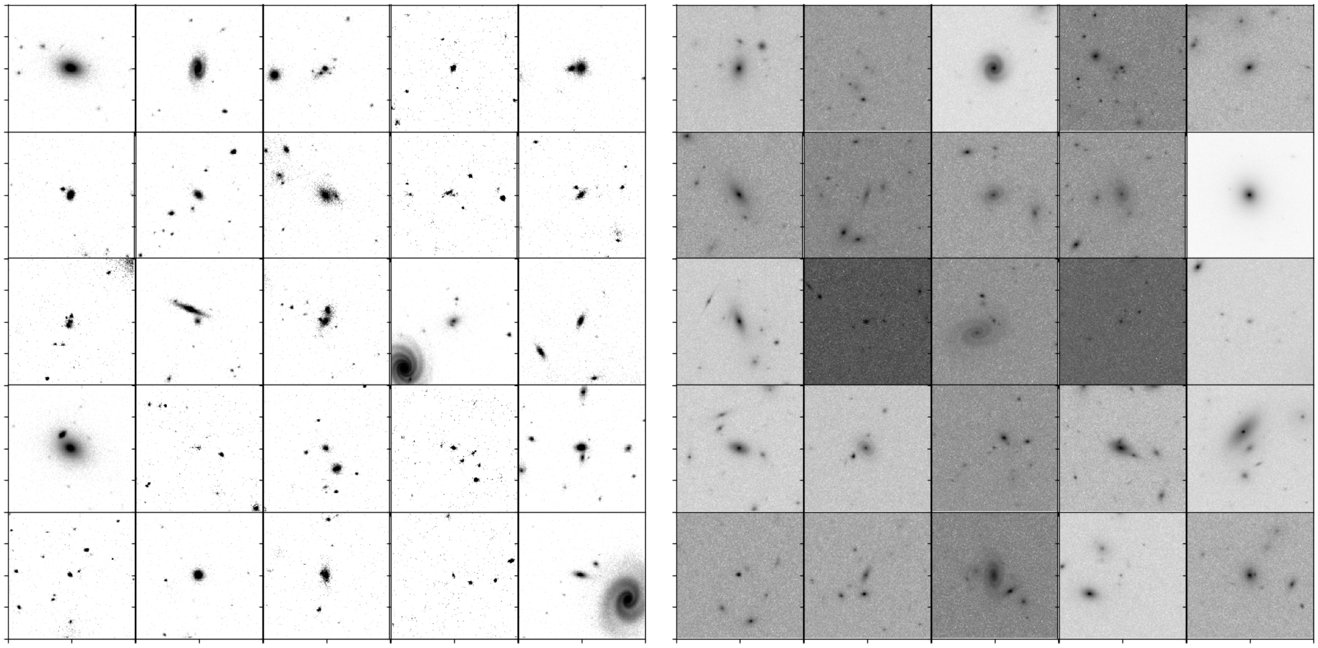

Figure 7. Left: False-positive adversarial examples for the LASTRO network. Most show a relatively high number of companions or are spiral galaxies with distinct arms. Right: False-positive adversarial examples for ResNet implementation,

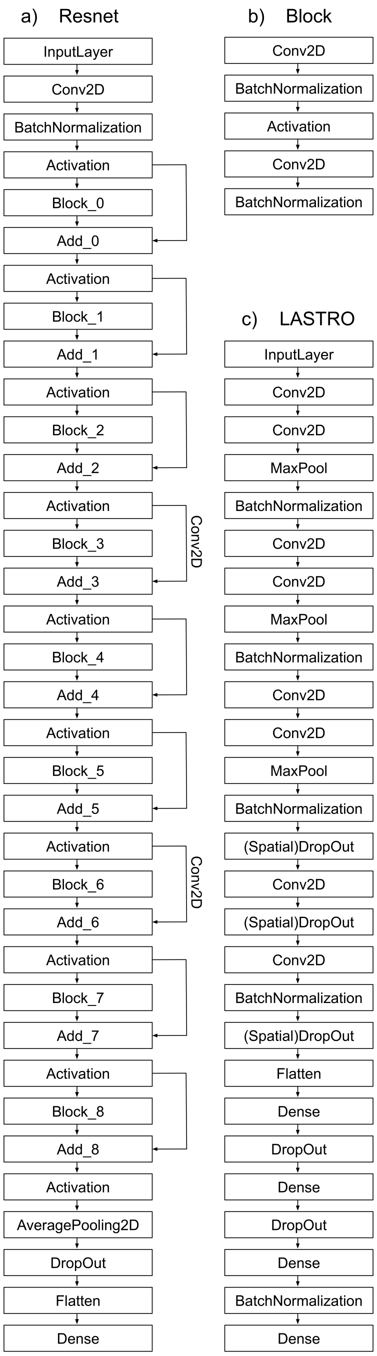

Figure 8. Architectures of the Neural Networks used in this project. a) ResNet which uses the block from b). c) LASTRO architecture. See Section 4 for more details.