Deep Learning as an alternative to the deconvolution of images of galaxies captured with the Hubble Space Telescope

Luis Enrique Ramírez Peláez

International University of La Rioja, Logroño (Spain)

01/02/2023

ABSTRACT

The anomalies and artifacts of the real galaxy image captures cause the presence of noise that hinders the work of observation and investigation of astronomers. Poisson noise and the Point Spread Function (PSF) represent two typical cases that are usually treated for attenuation or suppression. Advances in Artificial Intelligence have enabled the construction of models that can be trained to reconstruct galaxy images, mitigating the aberrations inherent in the acquisition process. In this project, real images from the Hubble Space Telescope will be used to train three neural network architectures, AEPP2 (based on Autoencoders), U-Net and DIDN, and perform a reconstruction of them, eliminating the effect of the Point Spread Function, as would be done with a deconvolution algorithm. The tests carried out provide data with which it can be stated that the U-Net network model is the one that offers the best results in this context.

KEYWORDS

Astronomical imaging, COSMOS survey, deconvolution, Deep Learning, DIDN, Hubble Space Telescope, point spread function, U-Net.

I. INTRODUCTION

In the acquisition of astronomical images of galaxies, anomalies caused by observing instruments and atmospheric turbulence are introduced. These anomalies, called artifacts, affect the correct interpretation of the data received, causing errors of appreciation that affect scientific studies to a greater or lesser extent.

There are mainly two aberrations that affect captured images of galaxies: Poisson noise and the point scattering function (PSF). Poisson noise [1] is characteristic of electronic sensors associated with optical instruments used in image acquisition and, due to the random movement of photons striking the sensor, causes fluctuations in capture. This uncertainty value in the signal causes a certain level of noise in the final image. The point spread function (PSF) [2] defines the response of an optical sensor to pulsed light, and is specific to each picking instrument, manifesting itself visually in a certain degree of blur.

The correction of errors caused by Poisson noise and PSF has been approached from different points of view with different results. The advances in computer engineering, and in particular, Artificial Intelligence, has allowed the introduction of new ways of working, which are in continuous development and from which significant advances are being made.

This project will address the use of neural networks for the reconstruction of images of galaxies affected by the point scattering function. A dataset large and complete enough to perform the training has been developed, built from the "COSMOS real galaxy dataset" [3], consisting of 118,992 images of galaxies captured by the Hubble telescope in the COSMOS program (The Cosmic Evolution Survey) [4]. The dataset contains the PSF values of each image allowing you to perform a deconvolution to reconstruct what the original image would look like without noise.

The tests have been carried out by training three neural networks: The AEPP2 architecture [5] based on an autoencoder model; the U-Net architecture [6] configuring different depth levels and the U-Net-based DIDN architecture [7] to which additional threads are added.

The evaluation of the tests has allowed to determine that the U-Net network is the one that offers the best results for the reconstruction of images of galaxies, obtaining performance values according to the expected standards.

II. STATE OF THE ART

The effect of the point dispersion function

Real images captured with a space telescope are distorted by the effect of the PSF. If this PSF is considered invariant with respect to the position in the plane of the object, the final image is formed with the convolution of the real and the PSF. In the case of astronomical images, the convolution operates on the real image X and the matrix H corresponding to the PSF, generating the observed image Y. To the convolution process is also added noise of Gaussian type, represented as N [8]

$$Y = H * X + N$$

Since the captured images contain the aberrations produced by noise and PSF, if what is intended is to reverse the scattering of light by relocating it in its original place, what is done mathematically is the inverse operation, called deconvolution [9] where the key is to know the corresponding PSF, whose calculation, however, It's nothing trivial.

Deconvolution techniques

Classical deconvolution techniques such as those described in [10] are based on post-processing of the captured image using the known PSF. These include the CLEAN method, the Richardson-Lucy algorithm, the Wiener filter and Bayesian approximations. Other techniques described in [11] use maximum entropy, gradients, the R-L-EM algorithm or FMPE. However, it is not always possible to have the PSF image or its estimation is complex. In these cases, blind deconvolution can be done [12] where, without knowing the PSF, an estimate of this is made based on the use of certain assumptions, which may not generate optimal results.

Deconvolution with deep learning

Deep learning techniques have brought a new perspective in solving the problem of capturing astronomical images. The training of neural network architectures with enough captured image - target pairs allow the prediction of how a real image will be given its capture, without the need to know its PSF, neither real nor estimated, which saves precisely the most complex part of the image reconstruction.

[13] describes the different neural network architectures that are commonly used for image deconvolution problems. One option is the use of classical CNNs (Convolutional Neural networks) [14] that alternately employ convolution layers with a certain kernel size, and max pooling layers to obtain singular characteristics. Although initially they could be considered as valid for obtaining images without aberrations, in practice it has been proven that they are not so successful, mainly due to the use of small kernels. CNN architectures such as AlexNet or VGG also incur poor results when the background of the images is not uniform, adding blurred areas.

Another type of architecture is GAN networks [15] which involve a completely different approach. Used for deconvolution, the generating network creates a false image that resembles as closely as possible the original without noise, and the discriminator network is trained to distinguish images with noise from those without noise. The process ends when the generator can construct an image that the discriminator considers free of aberrations. The biggest difficulty is learning the nature of the existing noise. In the proposal of [16] a CNN is used as a generating network for the GAN, improving the prediction. Another common drawback of GAN networks, and that affects this scenario, is their difficulty for training, which requires numerous tests and adjustments.

The use of Autoencoder architectures also allows to approach the process of image deconvolution, where the first phase of encoder transforms the input image into a compact representation trying to eliminate noise. Later in the decoder phase , the original image is recomposed without the removed characteristics. The difficulty lies in finding a balance between the depth of the network and the results in the reconstruction of the original image.

A modification to the use of Autoencoders that improves their problems is proposed by [17]. It is a very deep convolutional network of Autoencoders type that consists of a coding stage and a decoding stage. The particularity lies in the use of a characteristic of ResNet networks, the existence of skip-connections, which connect non-adjacent layers of both stages, which propagates the information in both directions, avoiding the loss of characteristics of the input image.

Other architectures with a philosophy like Autoencoders are U-nets, which are adapted to deconvolution processes and have their origin in medical image segmentation tasks [6]. They consist of a first stage of coding in which characteristics of the image are extracted, and a second stage of decoding, symmetrical to the first. In both phases there is a communication between symmetrical layers, transferring information, and avoiding that characteristics are lost in the encoding. In [8] a U-Net architecture is used to eliminate aberrations in simulated galaxy images. Based on the simulated image and the PSF, a deconvolution process is carried out using two algorithms, Tikhonov and ADMM. This generates a training dataset with which the U-net network learns the appropriate parameters to, in the prediction phase, be able to eliminate the aberration of a test image. The use of the Tikhonov algorithm and the U-Net network is called the Tikhonet network.

An evolution and improvement of the Tikhonet architecture is the proposal called ShapeNet that takes into account the shape of the structure of galaxies, adding to the loss function a new term related to the degree of ellipticity of these.

In other works, hybrid architectures such as SeeingGAN [18] are built, where ResNet and U-Net are combined and some additional changes are also implemented. On the other hand, the loss function is modified using a combination of Wasserstein loss [19] and perceptual loss (loss on the feature maps of the convolutional layers).

U-Net architectures have positioned themselves as very effective in situations where there is a shortage of data, achieving good results in reduced datasets. This quality is also explored with the Learnlet architecture [20] which is based on the use of gradient descent functionality to enhance the expressive power of wavelets. This study compares with traditional U-Nets with varying degrees of depth. These architectures are also studied for the deconvolution of images of galaxies, such as the work carried out by [21] in which the results are contrasted with Tikhonet, Learnet and U-Net64, and modifications are introduced to evaluate the metrics, based on the use of a window in the space of the image where the error calculations are performed. The authors conclude that the latter two models, and especially U-Net, obtain superior results by being able to generalize better than Tikhonet.

Architectures such as DIDN [17] extend the complexity of models with promising results. This proposal includes additional sub processing stages that operate like U-Nets. The iterative process carried out manages to improve the elimination of noise in the images.

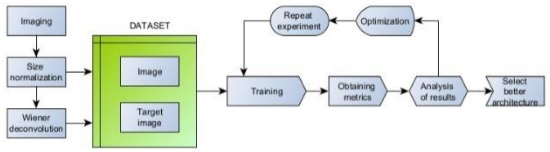
III. Objectives and Methodology

The main objective is the reconstruction of images of galaxies captured with the Hubble telescope affected by the dot scattering function. As secondary objectives, it is intended to:

1. Build a homogeneous and valid image dataset for training, validation and testing.
2. Achieve an efficient mechanism for the treatment and storage of images.
3. Identify and study the existing deconvolution techniques and their parameterizations to obtain the target image with which to train.
4. Study the neural network architectures that can be used successfully, as well as the best parameterization.
5. Determine the error metrics that best suit this type of studies.
6. Identify the most appropriate neural network for this area.

For the development of the project, a quantitative empirical methodology has been followed, following the phases of problem statement, formulation of hypotheses, data analysis and extraction of conclusions. Specifically, and particularly to the work context, the design of the experiments has been carried out as shown in Figure 1.

Figure 1 – Design of the experiments.



IV. Contribution

According to the definition of the main and secondary objectives, a comparison of valid architecture for the reconstruction of images of galaxies has been proposed. A set of captures of which its PSF is available has been sought, with which a deconvolution will be made, which will be the objective image to be predicted by the neural network. In practical terms the neural network will perform a process like a blind deconvolution.

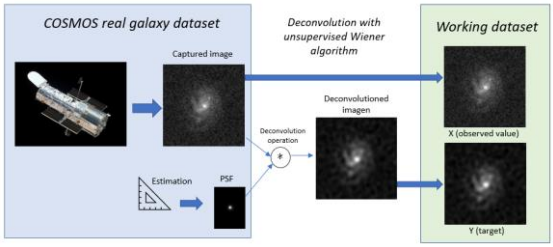
Construction of the working dataset

From the collection of images "COSMOS real galaxy dataset" has been built a working dataset consisting of 9,625 images of dimensions 128x128 pixels. Hubble's captures range in sizes from a few pixels to 900x900. All those between 110x110 pixels and 140x140 pixels have been selected, adapting to the final dimension by eliminating rows and columns from the ends or adding them with a value equal to 0, thus respecting the image information.

For training, the image values are normalized to the range [0,1] without affecting the image information.

The deconvolution process has been carried out with the Unsupervised Wiener algorithm that makes an automatic estimation of the regularization parameter necessary for the deconvolution, which allows the process to be automated, making it independent of the visual criterion of the scientist.

Figure 2 - Working dataset creation process.



Evaluation metrics

The evaluation of the error in the predictions is carried out with the following metrics:

Mean squared error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{i-}\, x_i)^2$$

Peak Signal-to-Noise Ratio (PSNR)

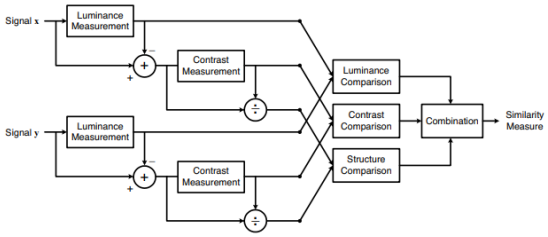
$$PSNR = 10log_{10}\left(\frac{MAXi}{MSE}\right) = 20log_{10}\left(\frac{MAXi}{\sqrt{MSE}}\right)$$

where MAXi is the maximum value of the pixels, represented by B bits per sample, and $MAXi = 2^B - 1$

PSNR measures the ratio of the maximum possible energy of a signal to the noise affecting it. Measured in decibels, normal values range from 30 dB to 50 dB, the higher they are better.

Structural Similarity Index (SSIM). This measure proposed by [21] approximates what human vision does to compare two images. Instead of using the raw data of the image values, it employs a combination of luminance, contrast, and structure to assess the structural similarity between neighboring pixels.

Figure 3 – Diagram of the SSIM calculation. Image:[22]



Each of these three elements is calculated as follows:

$$Luminance(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$Contrast(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$Structure(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

μ_x *is the average of the pixels in x image*
μ_y *is the average of the pixels in y image*
σ_x^2 *is the variance of x image*
σ_y^2 *is the variance of y umage*
σ_{xy} *is the covariance*
$C_1, C_2\ y\ C_3$ *are stabilization coeficients for divisions*

In this way, the SSIM calculation is expressed as follows:

$$SSIM(x,y) = l(x,y)^{\alpha} \cdot c(x,y)^{\beta} s(x,y)^{\gamma}$$

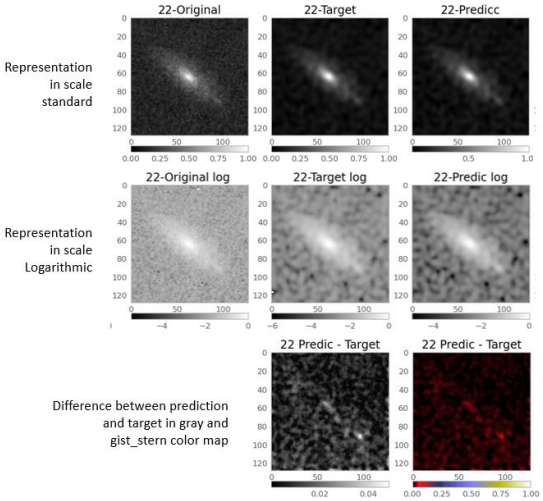
α, β, γ *are weights to weight each of the components*

The SSIM value ranges from 0 to 1, with 1 being 1 when two images are identical. Although it depends on each case, a value greater than 0.97 indicates that the images are very similar. This metric is very sensitive and a small change in value can mean a significant improvement or worsening. For the scope of this project, the use of SSIM

allows to evaluate the simulated deconvolution process in the neural network, that is, to determine if the image calculated by the network from the original is sufficiently like the target.

It will also provide a mechanism for a visual evaluation of the results, comparing the calculated image with the target image. On the one hand, the result of the execution of the model on the test images on a logarithmic scale is shown (Figure 4), which allows to better appreciate some structures of the galaxies.

Figure 4 - Results of a test. First row and from left to right, original image, objective and predicted. Center row and from left to right, original images, objective and predicted in logarithmic representation. Bottom row from left to right, differences between predicted image and target image, with grayscale color maps and "*gist_stern* ".



On the other hand, a subtraction of values of the objective image and the one predicted by the model (residual value) has been carried out, showing itself in a visually significant color map, specifically the *"gist_stern*" map, which allows small variations to be easily appreciated. a good prediction, such as the one shown in Figure 4, will have a residual image with black tones. The worst predictions will appear in reddish tones going to bluish and yellowish if these are very incorrect.

Hyperparameters

The same values have been used in all models to be able to compare under similar execution conditions and thus determine the best architecture.

Optimizer: The Adam optimizer has been used *with* learning rate *values of* 1.0E-3, 1.0E-4, 1.0E-5 and 1.0E-6 according to the particularities of each training.

Loss function: The structural similarity index (SSIM) is used as it is an appropriate metric to compare images, inverting their value, that is, subtracting from 1.

Metric: The metric used in training is *Mean Absolute Error* (MAE). This function allows you to assess the learning of the network, something that will happen when this value gradually decreases.

Early Stop: The stop occurs when the *loss* rate is less than 0.02, or what is the same, when the SSIM reaches the value of 0.98.

Epoch: Training cycles of 100, 150 and 200 epochs, adapting to the characteristics of each network and the preliminary results.

Batch size: Sizes of 32, 63, 128 and 256.

Distribution of training, validation, and test data : All models have been trained with the same training and validation images, although in each run they have been mixed at the beginning using the *Keras shuffle* parameter. All architectures have been tested with identical images and thus be able to efficiently compare the results. Specifically, 70% of training data, 17% validation data and 13% test data have been used.

Hardware and software resources

The training tests have been run in two environments: local and online: Local environment: Computer with AMD RYZEN 5 5600X CPU and NVIDIA RTX 3060Ti 8GB GPU. Online environment: Google Colab. NVIDIA A100-SXM4 40GB GPU

All the programming part has been done in Python v3.9.12 using the following libraries: TensorFlow v2.10.0, Keras v2.10.0, Scikit-image v0.19.2, Numpy v1.21.5, Matplotlib v3.5.1, Astropy 5.0.4

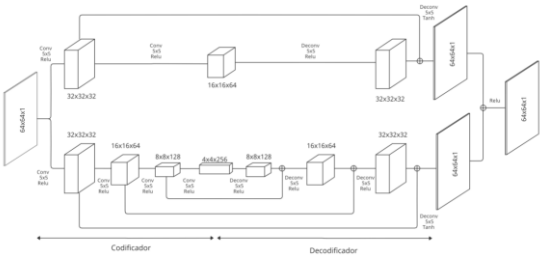
Architectures used

AEPP2 architecture

It is a model based on the proposal of [23] that uses autoencoders in two branches, in which convolution and deconvolution operations are performed. The AEPP2 architecture was employed by [5] considering some modifications for a better adaptation to noise elimination.

This model consists of 2 branches, the upper one includes 2 convolution operations (coding or compression) and another 2 deconvolution operations (decoding or decompression); The lower one consists of 3 operations of each type. The idea is to transform the original image into a compact representation by trying to eliminate noise (compression) and then recompose the original image without the removed features (decompression). The existence of 2 branches is justified in being able to include in the same process two levels of noise elimination. Thus, the lower branch produces greater compression, eliminating more noise, but remembering fewer details. The upper branch, with a lower compression capacity, eliminates less noise, but allows the original image to be better recovered.

Figure 5 - AEPP2 architecture. Image de [5]



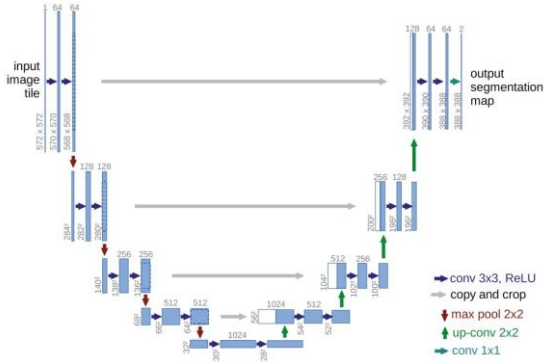
To avoid the loss of information caused by the use of *autoencoders*, skip-connections *are included*, which are connections between layers of the beginning and end to preserve the characteristics of the images. This problem stems from using too much depth in processing, which can cause gradient fading. In fact, the starting model proposed

by [23] limits this depth.

U-Net architecture

This model uses a philosophy similar to Autoencoders by executing two stages, one of coding where the characteristics of the image are extracted, and a second stage of decoding, symmetrical to the first, and where the original image is reconstructed. There is a communication between symmetrical layers, which transmits the information from the input to the output, avoiding that too many characteristics are lost. Maxpooling is also performed after each stage, reducing dimensionality without losing important information.

Figure 6 - U-Net architecture. Image: [6]

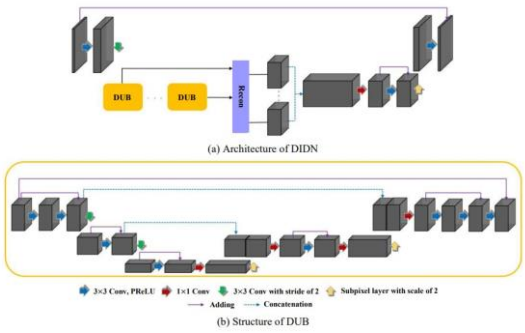


For the tests, 6 models with different levels of depth have been built (Table 2 of the Annex).

DIDN architecture

This model proposed by [7] starts from the structure of a U-Net but adds an iterative sub processing of the convolution process (DUB), fine-tuning and improving noise removal. In this case, the network consists of 4 stages: feature extraction, iterative reduction, and scaling (DUB threading), rebuilding and improvement. The iterative process performed with the structure is essentially a U-Net network, and its iteration improves the elimination of noise in the images. For the tests, two simplified variants of the original model have been built, consisting of 16 filters in input, with 1 and 2 DUB sub processing units.

Figure 7 - DIDN architecture. General scheme (a) and internal threading (b) – Image: [7]



V. Evaluation and Results

The evaluation of the results has included the analysis of training statistics, the study of prediction metrics and the visual comparison of images, of which some examples have been included in the Appendix.

Table 1 - Parameters, training metrics and prediction of the best models of each architecture.

MODEL AND ARCHITECTURE			EVALUATION TRAINING			PREDICTIONS METRICS			
MODEL	EPOCH	BATCH SIZE	LOSS (1-SSIM)	METRIC (MAE)	T (SEG)	MSE	RMSE	PSNR	SSIM
AEPP2	100	128	0,0571	0,0154	819	37,8199	6,1498	32,3536	0,9540
UNET4	200	32	0,0426	0,0159	1226	57,0713	7,5546	30,5666	0,9638
UNET8	100	64	0,0409	0,0149	648	32,1079	5,6664	33,0647	0,9642
UNET16	200	32	0,0448	0,0173	2690	61,4367	7,8382	30,2465	0,9670
UNET32	100	256	0,0316	0,0125	674	35,0267	5,9183	32,6868	0,9710
UNET64	100	32	0,0369	0,0137	1666	28,5203	5,3404	33,5793	0,9703
UNET128	100	128	0,0359	0,0147	3638	50,3681	7,0971	31,1092	0,9707
DIDN16 DUB1	150	128	0,0690	0,0176	1523	50,2390	7,0879	31,1204	0,9225
DIDN16 DUB2	150	32	0,0576	0,0170	2361	49,4050	7,0289	31,1931	0,9320

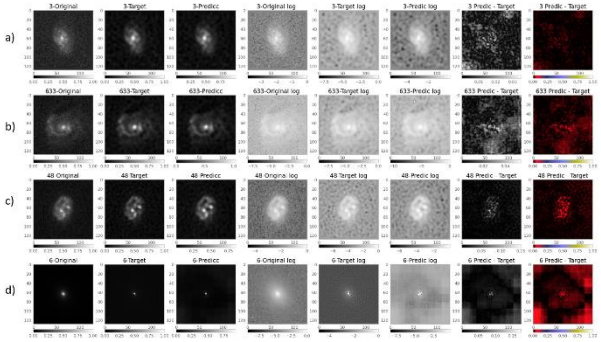
AEPP2 architecture evaluation

There have been 8 experiments with each value of batch size: 32, 64, 128 and 256 in two batches of 100 and 200 batch, observing that, generally, from the epoch 40 convergence was practically reached. It is also noted that with smaller batch sizes the model produces some overfitting, an aspect that is corrected with larger sizes.

In all tests the results are very similar, with values of 0.95 for SSIM (with differences of thousandths) and between 31 and 32 for PSNR. The model with the best behavior is the one executed with batch size equal to 128 since it also offers the best figures for PSNR and MSE.

Regarding the visual result and in line with the SSIM values obtained, the reconstructions carried out are generally good, although with differences according to the type of galaxy in question. However, it is not trivial to find a pattern for better or worse results depending on the physiognomy of the galaxy. Figure 8 shows some examples of model behavior. In 8.a) a very good reconstruction is carried out. The residue that remains when making the difference between the predicted image and the objective image shows low values, without appreciating highlighted structures of the galaxy.

Figure 8 - Examples of predictions. a) optimal reconstruction. b) Efficient reconstruction of the structure of the galaxy. c) Poor reconstruction of the structure of the galaxy. (d) Grid effect. From left to right, original image, target, and prediction. Their corresponding views on logarithmic scale. Finally, it differentiates between prediction and target, in grayscale and in color map "gist_stern".



In 8.b) the reconstruction is worse because a residue with more information remains. In 8.c) the structure of the galaxy is negatively affected by the prediction, something that can be seen with the naked eye observing both images and is ratified with the residual value. In 8(d) an anomaly is observed that has been called "grid effect". The

prediction generates a grid throughout the image, which is best seen on the logarithmic scale and in the residual image. This effect occurs in most predictions, although it is usually very attenuated and only by making a bulky enlargement can it be appreciated. This grid is formed by elements of 16x16 pixels, in total 8x8 mini blocks. It so happens that the architecture uses *kernel* gradually until reaching a minimum image level of 16x16, which seems to be the reason for this irregularity.

This behavior resulting from the architecture of the neural network causes the relatively good data of the metrics to be overshadowed, leading to the conclusion that this model is not appropriate to the work context.

UNET architecture evaluation

35 experiments have been carried out with the 6 variations of the initial architecture, with the common hyperparameter configuration, but choosing certain values based on the preliminary results, since sometimes the model did not converge, and the training plan had to be varied. In general, the greater the complexity of the neural network, the lower the learning rate to ensure that the training was carried out without errors. It has also been observed that in the most complex models (Unet32, Unet64 and Unet128), trained in Google Colab, the execution of 200 epoch made the model fall into overfitting, so the tests have been carried out only with 100 epochs.

In general, the results are good, exceeding in almost all cases the figure of 0.95 for the SSIM, reaching 0.96 on numerous occasions and reaching 4 times a value of 0.97, in which figures between 30 and 33 are also obtained for the PSNR. In absolute terms the best model corresponds to Unet-64 with batch size equal to 32 (Table 1).

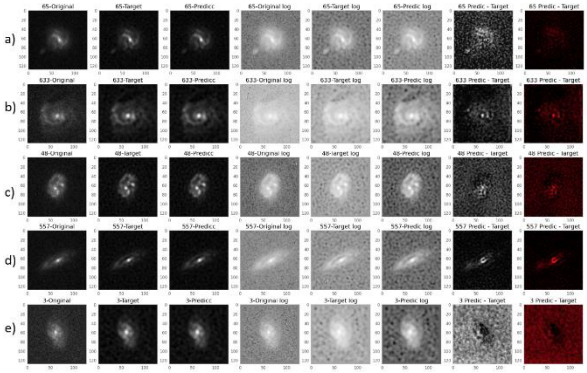
There is a tendency to obtain better results as the complexity of the architecture increases, but, although the value of SSIM is very sensitive to small changes and the objective of the tests is to maximize it, the increase in the associated training cost is significantly greater than the gain in SSIM and PSNR. Simple models offer more homogeneous results, while complex models get very good metrics, but also inferior even to simple ones.

A feature of this architecture is that convergence is achieved quickly. In most cases with just over 30 epoch acceptable values close to the optimal are achieved.

In some trainings there has been overfitting, this circumstance has taken place in all types of models, regardless of their complexity, but only on some occasions has been a detrimental factor, being on the contrary a very slight overfitting.

The visual evaluation of the results offers a positive balance. In general, the reconstructions carried out by the different configurations of the U-Net architecture are good. Figure 9 shows some examples.

Figure 9 - Examples of predictions made with the U-Net network. From left to right, original image, target and prediction. Their corresponding views on logarithmic scale. Finally, it differentiates between prediction and target, in grayscale and in "gist_stern" color map.



In 9.a) an optimal reconstruction is achieved. In 9.b) and c) the differences are minimal, respecting the structure of the galaxies. In 9.d) the process is not so perfect because the body of the galaxy contains some differences that are denoted by the residual image. In 9.e), although apparently the result is worse, the structure of the galaxy is reconstructed quite well, showing the residual image black tones in the center.

With U-Net the occurrence of the so-called "grid effect" is not appreciated at any time.

DIDN architecture evaluation

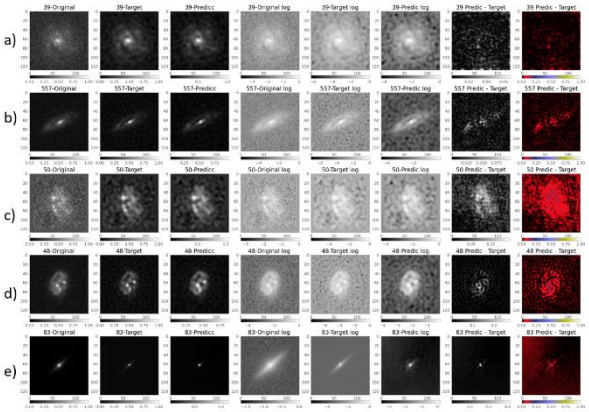
The experiments with this architecture have been carried out with the 2 models of 1 and 2 units of sub processing DUB.

The results do not reach a remarkable quality, since in no case is the value of 0.93 for SSIM exceeded. The model with 2 blocks of DUB sub-processing behaves slightly better, being the test with batch size equal to 32 and trained with 150 epoch the one that obtains the best results (Table 1).

A common feature of all workouts is that convergence is somewhat slow but stable, and no tendency to overfitting has been observed. The values of the loss rate gradually decrease to approximately 100 epochs, at which point this rate practically stops improving.

The visual analysis of the predicted images allows to appreciate in a generalized way defects to a greater or lesser extent. Even so, sometimes these predictions are satisfactory. Figure 10 shows some results. The appearance of the "grid effect" observed with the AEPP2 architecture has been verified. This behavior is not very frequent and has appeared only in the model with 1 DUP block. Training with the 2 DUB model does not show this particularity. In addition, the effect is not the same, because on this occasion, the anomaly is formed by one or two stripes as a frame of the image.

Figure 10 - Examples of predictions made with the DIDN network. From left to right, original image, target, and prediction. Their corresponding views on logarithmic scale. Finally, it differentiates between prediction and target, in grayscale and in "gist_stern" color map.



In Figure 10.a) the prediction is optimal. In 10.b) the residual image indicates that the reconstruction fails in the structure of the galaxy. This behavior manifests itself immediately in 10(d). In 10.c) the residual value indicates quite differences that affect the entire image. In 10.e) there is apparently not much residual value, however, the halo of the galaxy is affected and lost substantially.

VI. Discussion

The U-Net architecture stands as the best proposal in this comparison. The quantitative results obtained from the SSIM and PSNR metrics for some of its configurations are the best, also achieving a very acceptable figure considering the nature of these evaluation measures. The goal of deconvoluting images of galaxies captured with the Hubble Space Telescope is possible with this neural network model. It should be noted that the U-Net architecture greatly respects the structure and physiognomy of the images, in this case galaxies, not introducing anomalies or deformations that pose a damage from the scientific point of view. The results in the predictions that are sometimes not satisfactory imply a decrease in the quality of the reconstruction, but do not introduce alterations such as the "grid effect" observed with the DIDN network and to a greater extent with the AEPP2 network.

The great flexibility and easy scalability of the U-Net network makes it possible to experiment and execute trainings that allow determining the best parameterization to maximize the best results. All this helps to counteract the relative variability of the models, which, as explained, means that some tests do not offer satisfactory results. Also noteworthy is the rapid convergence of this network model, which also implies vigilance to avoid overfitting.

The AEPP2 architecture, although it offers satisfactory results, does not reach the quality level of the U-Net. As described, sometimes the reconstruction of the images is not completely satisfactory, even with the naked eye. In addition, the appearance of the anomaly described as "grid effect" is a major problem impossible to ignore. With all this it can be affirmed that the use of the ARPP2 architecture is not valid for the deconvolution of images of galaxies.

Regarding the DIDN model, the results are the worst of the three alternatives, obtaining images generally poorly reconstructed. The small improvement of using two sub processing units suggests that the inclusion of a greater number of them will improve these results. Another aspect that the DIDN network has highlighted is its slow convergence, making it necessary to execute many cycles or epoch to obtain acceptable metrics. The positive feature of this model is its good performance against overfitting, the best of the three architectures tested.

VII. Conclusions

The problem of aberrations produced in the capture of astronomical images has motivated the development of techniques that allow their elimination and help astronomers in their scientific studies. The reconstruction of these images with mathematical algorithms has given way to new techniques based on artificial intelligence, which are proving their validity as an alternative. The use of supervised machine learning with neural networks for the deconvolution of astronomical images is an important alternative to the procedures commonly used. The possibility of reconstructing images to how they would really be without having the point dispersion function is like the blind deconvolution process, with the advantage of having learned to perform this process with previously validated data.

In this project, captures from the Hubble Space Telescope have been used to train different configurations of three neural network architectures. The UNet model is positioned as a robust option for the effective deconvolution of astronomical images. The process of compression and decompression extracting significant characteristics, while restoring imperfections, has shown good performance, which, together with the particularities of the architecture, mainly scalability and rapid convergence, make it of great help to the scientific community for the study of images of galaxies free of aberrations.

Proposals for improvement

1.- Increase in the size of the training dataset: One of the limitations that has been had is the number of images with which to train. Although the source dataset contains more than 118,000 captures, those used in the tests have been limited to 9,625. The reason for this limitation has been the choice of an image size of 128x128 that has been considered visually representative, and compatible with the built architectures. However, the starting dataset contains mostly smaller images, which could also be used to adapt the network. Ideally, we should be able to use the largest number of images of galaxies, scaling them to a higher fixed size, processing them with the neural network and reducing them again. The key would be to study how this operation affects the use of the PSF for the deconvolution of the target image.

2.- Use of real images as a target: The working images have been obtained from the astronomical observation campaign COSMOS, The Cosmic Evolution Survey. This same campaign has been planned to take over the new James Webb Space Telescope, in the COSMOS-Web study, which will capture the same sector of the sky. This capture cycle has already begun but will be completed during the years 2023 and 2024. Once completed, since the captured galaxies will be the same and are identified and cataloged, a training dataset could be built consisting of Hubble images and Webb images that would be the *target*

to have a higher quality and resolution. In this way, predictions of new catches would be based on a trained model with real information of higher quality.

3.- Use of GAN (Generative Adversarial Network). This type of neural networks implies a different approach to training, generating approximations of the objective image until it is considered valid. For this use case, as it is an unsupervised learning, the reconstruction would be carried out without any additional information to the captured image, which avoids having to deconvolute previously to obtain the target. The results obtained by this type of architecture are proving surprising and useful in the areas where they are experimented with and makes them worthy of testing in the reconstruction of astronomical images.

APPENDICES

Tabl3 2 - Number of filters and parameters of trained neural networks.

MODEL	Nº SUCCESSIVE FILTERS	TOTAL PARAMETERS
AEPP2	Upper branch: 32, 64, 32 Lower branch: 32, 64, 128,256, 128, 64, 32	1.444.674
U-Net 4	4, 8, 16, 32, 64, 32, 16, 8	121.725
U-Net 8	8, 16, 32, 64, 128, 64, 32, 16, 8	485.813
U-Net 16	16, 32, 64, 128, 256, 128 64, 32, 16	1.941.093
U-Net 32	32, 64, 128, 256, 512, 256, 128 64, 32	7.760.069
U-Net 64	64, 128, 256, 512, 1024, 512, 256, 128 64	31.031.685
U-Net 128	128, 256, 512, 1024, 2048, 1024, 512, 256, 128	124.109.573
DIDN16 1DUB	16, (16, 32, 64, 128, 32, 64, 16, 16), 16	1.507.360
DIDN16 2DUB	16, (16, 32, 64, 128, 32, 64, 16, 16) x2, 16	2.005.792

Figure 11 - Results of the predictions of all the models of the three trained architectures.

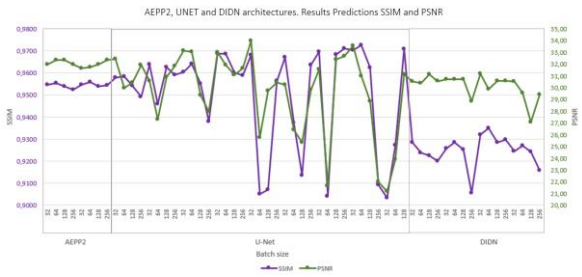
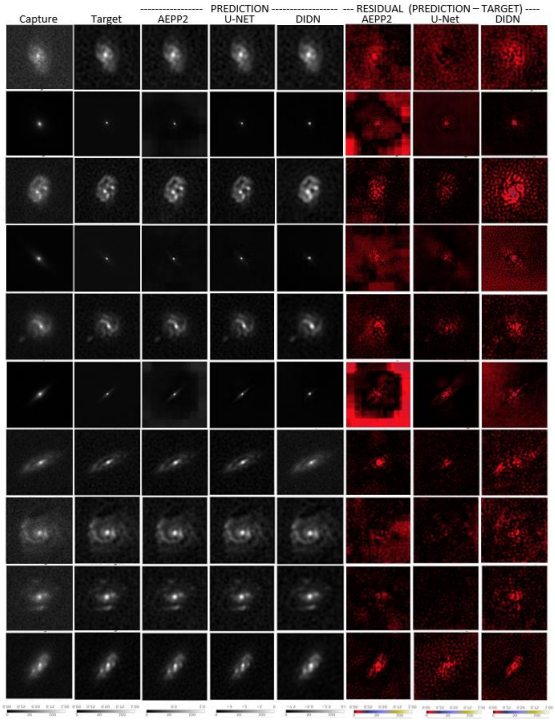


Figure 12 - Examples of predictions made with the best models of each architecture. From left to right, captured image, *target*, prediction with AEEP2, Unet, DIDN and residual image in AEPP2, Unet and DIDN.



REFERENCES

[1] S. W. Hasinoff, «Photon, Poisson Noise», en *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 608-610. doi: 10.1007/978-0-387-31439-6_482.

[2] V. Sacek, «Point spread function PSF», *Amateur telescope optics*, 14 de julio de 2006. https://www.telescope-optics.net/diffraction_image.htm (accedido 20 de enero de 2023).

[3] R. Mandelbaum, C. Lackner, A. Leauthaud, y B. Rowe, «COSMOS real galaxy dataset». Zenodo, 1 de enero de 2012. doi: 10.5281/zenodo.3242143.

[4] N. Scoville *et al.*, «The Cosmic Evolution Survey (COSMOS): Overview», *ApJS*, vol. 172, n.º 1, p. 1, sep. 2007, doi: 10.1086/516585.

[5] J. Hernández Afonso, «Redes Neuronales Convolucionales para la Reconstrucción de Imágenes de Galaxias», TFM, Universidad Internacional de La Rioja, 2022.

[6] O. Ronneberger, P. Fischer, y T. Brox, «U-Net: Convolutional Networks for Biomedical Image Segmentation», en *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234-241. doi: 10.1007/978-3-319-24574-4_28.

[7] S. Yu, B. Park, y J. Jeong, «Deep Iterative Down-Up CNN for Image Denoising», en *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, jun. 2019, pp. 2095-2103. doi: 10.1109/CVPRW.2019.00262.

[8] F. Sureau, A. Lechat, y J.-L. Starck, «Deep learning for a space-variant deconvolution in galaxy surveys», *A&A*, vol. 641, p. A67, sep. 2020, doi: 10.1051/0004-6361/201937039.

[9] G. Sparacino, G. De Nicolao, G. Pillonetto, y C. Cobelli, «Deconvolution», en *Modelling Methodology for Physiology and Medicine (Second Edition)*, E. Carson y C. Cobelli, Eds. Oxford: Elsevier, 2014, pp. 45-68. doi: 10.1016/B978-0-12-411557-6.00003-3.

[10] J. L. Starck, E. Pantin, y F. Murtagh, «Deconvolution in Astronomy: A Review», *PASP*, vol. 114, n.º 800, p. 1051, oct. 2002, doi: 10.1086/342606.

[11] R. Molina, J. Nunez, F. J. Cortijo, y J. Mateos, «Image restoration in astronomy: a Bayesian perspective», *IEEE Signal Processing Magazine*, vol. 18, n.º 2, pp. 11-29, mar. 2001, doi: 10.1109/79.916318.

[12] S. M. Jefferies, J. C. Christou, S. M. Jefferies, y J. C. Christou, «Restoration of Astronomical Images by Iterative Blind Deconvolution», *ApJ*, vol. 415, p. 862, oct. 1993, doi: 10.1086/173208.

[13] M. Makarkin y D. Bratashov, «State-of-the-Art Approaches for Image Deconvolution Problems, including Modern Deep Learning Architectures», *Micromachines*, vol. 12, n.º 12, p. 1558, dic. 2021, doi: 10.3390/mi12121558.

[14] K. O'Shea y R. Nash, «An Introduction to Convolutional Neural Networks», nov. 2015. doi: 10.48550/arXiv.1511.08458.

[15] I. Goodfellow *et al.*, «Generative adversarial networks», *Commun. ACM*, vol. 63, n.º 11, pp. 139-144, oct. 2020, doi: 10.1145/3422622.

[16] H. Y. Lee, J. M. Kwak, B. Ban, S. J. Na, S. R. Lee, y H.-K. Lee, «GAN-D: Generative adversarial networks for image deconvolution», en *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, oct. 2017, pp. 132-137. doi: 10.1109/ICTC.2017.8190958.

[17] X. Mao, C. Shen, y Y.-B. Yang, «Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections», en *Advances in Neural Information Processing Systems*, 2016, vol. 29. Accedido: 8 de febrero de 2023. [En línea]. Disponible en: https://proceedings.neurips.cc/paper/2016/hash/0ed9422357395a0d4879191c66f4faa2-Abstract.html

[18] F. K. Gan, K. Bekki, y A. Hashemizadeh, «SeeingGAN: Galactic image deblurring with deep learning for better morphological classification of galaxies». arXiv, 28 de marzo de 2021. doi: 10.48550/arXiv.2103.09711.

[19] M. Arjovsky, S. Chintala, y L. Bottou, «Wasserstein Generative Adversarial Networks», *Proceedings of the 34th International Conference on Machine Learning*, pp. 214-223, jul. 2017.

[20] Z. Ramzi, K. Michalewicz, J.-L. Starck, T. Moreau, y P. Ciuciu, «Wavelets in the Deep Learning Era», *J Math Imaging Vis*, vol. 65, n.º 1, pp. 240-251, ene. 2023, doi: 10.1007/s10851-022-01123-w.

[21] U. Akhaury, J.-L. Starck, P. Jablonka, F. Courbin, y K. Michalewicz, Eds., «Deep learning-based galaxy image deconvolution», *Frontiers In Astronomy And Space Sciences*, vol. 9, 2022, doi: 10.3389/fspas.2022.1001043.

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, y E. P. Simoncelli, «Image quality assessment: from error visibility to structural similarity», *IEEE Transactions on Image Processing*, vol. 13, n.º 4, pp. 600-612, abr. 2004, doi: 10.1109/TIP.2003.819861.

[23] P.-Y. Liu y E. Y. Lam, «Image Reconstruction Using Deep Learning». arXiv, 27 de septiembre de 2018. doi: 10.48550/arXiv.1809.10410.