Generation of synthetic satellite streaks for streaks segmentation

Servane Lunven
Supervised by Mathieu Salzmann and the SSA team
Autumn 2022

Submitted in partial satisfaction of the requirements for the degree of:
Computational Science and Engineering (CSE)
EPFL

1 Abstract

With the number of space debris, figure 3, growing day by day, it is essential to be able to monitor them to ensure safety on earth and in space. Deep Learning provides efficient tools for this but a lot of data are often required to ensure the efficiency of the methods. What is proposed in this project is an automated method to improve the reality of synthetic satellite streaks in astronomical images using a deep learning method: SimGAN [1]. It is a type of generative adversarial network, specifically designed for image-to-image translation tasks. For this, pictures of space taken with an OmegaCAM camera on the VLT telescope (ex: figure 2 and 1) in Chile are used. The efficiency of the refinement is tested by comparing the segmentations obtained with a UNet [3] trained on synthetic images and one trained with synthetic images refined by the SimGAN model. Small improvements are observed for the segmentation of bright streaks. For streaks that do not stand out from the background no improvement is observed.

Figure 1: Image with a real satellite streak

Figure 2: Image with a real satellite streak

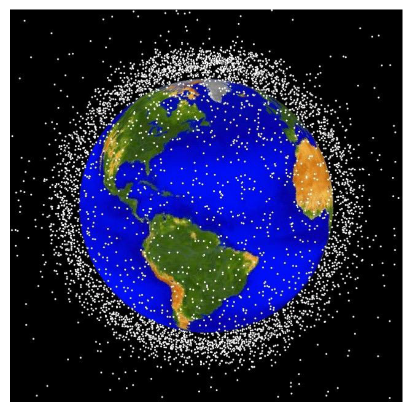

Figure 3: Space debris and human spacecraft

2 Introduction

A satellite streak is a long, thin trail of light that appears in the night sky and is created by the reflection of sunlight off the surface of a satellite. They can be difficult to detect. Indeed they are only visible when the satellite is in the observer's field of view and due to their velocity are sometimes visible only a few seconds. Moreover, it can be hard to distinguish them from other celestial objects. For all these reasons, pictures of satellite streaks are rare. It is necessary to get more of them to train deep learning models whose role are to detect them. Indeed the number of space debris is increasing rapidly, it is urgent to keep track of them. The goal of this project is to generate synthetic satellite track images, indistinguishable from real tracks, in order to expand the small satellite track dataset. For this purpose, artificial satellite tracks will be generated and then the reality of these tracks will be improved using the SimGAN model [1], a type of generative adversarial network specially designed for image-to-image translation task. Finally, the efficiency of the refinement will be tested by comparing the results of the segmentation of real traces obtained with two UNet [3]. One trained with unrefined synthetic images and one trained with refined synthetic images.

3 Reals satellite streaks

For this project, astronomical images belonging to LASTRO lab at EPFL are used. These images were taken with the OmegaCAM camera on the VLT telescope at Paranal in Chile with exposure time equal to 320s. Among these images, a hundred contained satellite streaks such as the ones in figures 2 and 1. To isolate these streaks, a csv file belonging to LASTRO, where the coordinates of the two extreme points of each streak were available, has been used. With these information, it was then possible to extract 64 x 64 patches around the streaks by computing their slopes. These streaks being long, 1200 patches of real streaks are obtained. Examples of these streaks are visible in figure 4.

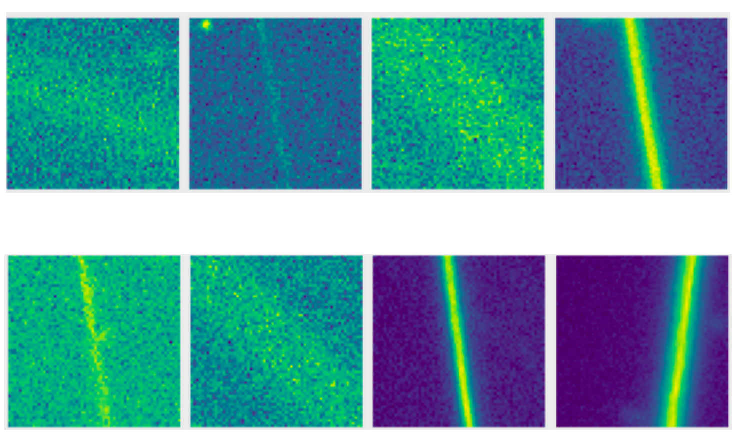

Figure 4: Patches of real satellite streaks.

4 Synthetic satellite streaks

The second step of this project is to generate images of synthetic satellite streaks. To achieve this, Yann Bouquet's method has been used and slightly modified. It consists in artificially generating streaks which are then overlayed on real astronomical images.

To get these backgrounds, fits images from LASTRO, taken with the same telescope as in section 3 are used. They do not have any streak in them. These fits images are mosaics, like in figure 5, of 16000 x 16000 pixels, composed of 32 blocks of 2000 x 4000 pixels separated by NaN values.

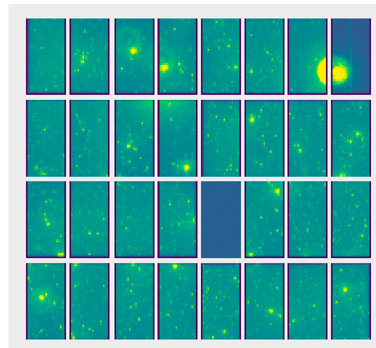

Figure 5: Fits image: Mosaic of 32 blocks.

The pixel intensities i in these blocks present a wide range of values: $i \in [0, 65535]$, consequently the images appear black, as visible in the left image of figure 6. To tackle this problem, a zscale transformation is applied for each image. It consists in displaying the pixel values near the median value of the image without the time consuming process of computing a full image histogram. The different objects in space, including the streaks, are then visible (right figure 6). Finally, these images are cut into 64x64 patches and are randomly flipped vertically and horizontally.The flips allow a patch to be reused multiple times.

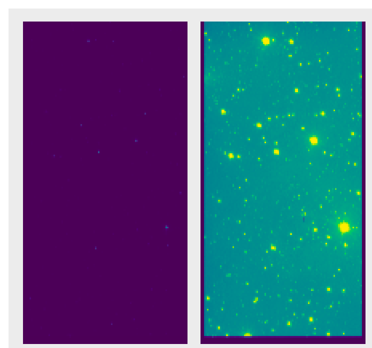

Figure 6: left: Before right: After z scaling

To create a satellite streak, a lign with a random width is artificially created. Its length is such that the streak crosses the entire patch. A Gaussian blur is applied on the streak, real ones having very changing intensities along the streak. A streak is created only a certain percentage of the time,so that the final dataset is composed of 4300 images with and without streak. In figure 7, are visible patches of real backgrounds before and after a streak has been overlayed on them.

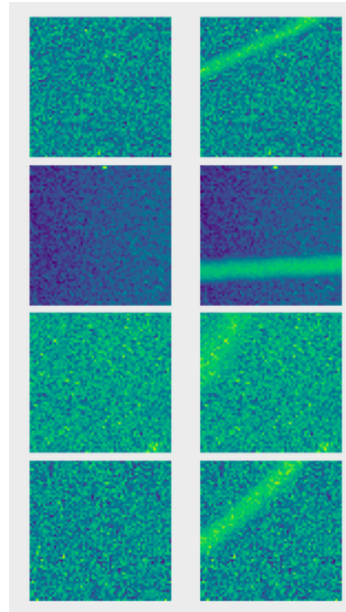

Figure 7: Examples of real backgrounds and created synthetic streaks.

5 Refinement of synthetic images with SimGAN

The purpose is to improve the reality of the generated synthetic images so that they look closer to real images of satellite streaks. For this, SimGAN, a type of generative adversarial network (GAN) specifically designed for image-to-image translation tasks, is used.

5.1 What is SimGAN

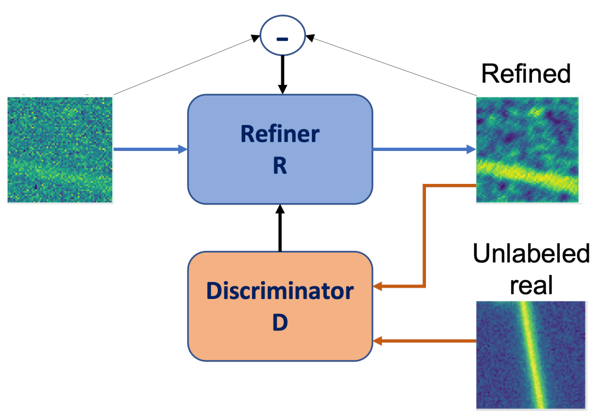

Figure 8: Scheme of the SimGAN model [2].

It is composed of two neural network models:

1. The **generator** is trained to produce new synthetic images that are similar to the given input synthetic images. It consists in a fully convolutional network with ResNet blocks. Details of the network are given in [1].

2. The **discriminator** is trained to distinguish between the synthetic images produced by the generator and real images from a target dataset. It consists in a fully convolutional network. Details of the network are given in [1].

The two models are trained together in an adversarial process, in which the generator tries to produce images that the discriminator cannot distinguish from real images, and the discriminator tries to correctly identify which images are synthetic and which are real.

1. The **discriminator** seeks to minimize a local cross entropy error for a two class problem:
$$L(\phi) = -\sum_i log(D_\phi(\widetilde{x}_i)) - \sum_j log(1 - D_\phi(y_j)) \tag{1}$$
where ϕ is the discriminator parameter, $D_\phi(.)$ the probability estimated by the discriminator that the input is a synthetic image, $\widetilde{x}_i$ a refined image and y_j a real image. The training of the discriminator presents two particularities:

   - Instead of classifying the entire image, the discriminator actually minimizes the sum of cross entropy losses over local patches.
   - To improve the stability of the training, the loss is computed using the prediction of 2/3 of the images coming from the current refiner network and 1/3 from a buffer containing past refined images.

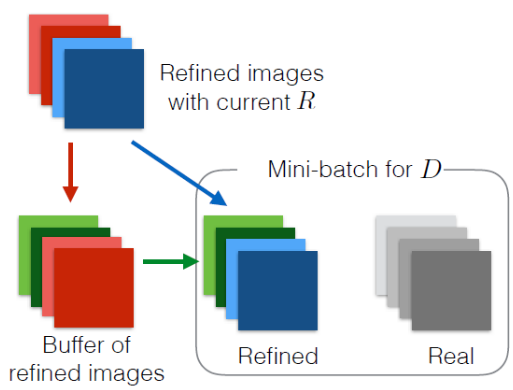

Figure 9: Scheme of the mini batch a discriminator receives for training [2].

2. The **refiner model** is trained to minimize the combination of an adversarial loss and a self regularization term, which penalizes large changes between the synthetic and refined images.
$$L(\phi) = -\sum_i log(1 - D_\phi(R_\theta(x_i))) + \rho||R_\theta(x_i) - x_i||_1 \tag{2}$$
where $R_\theta(.)$ is the refiner network with parameter θ and x_i a synthetic image.

Thus, the particularity of SimGan is that the generative model is trained using a combination of simulated and unsupervised learning techniques.

5.2 Results

In the project, the input dataset is composed of the generated synthetic satellite streaks. The patches containing only background are not taken as inputs, being already real. Thus, around 2200 images of synthetic streaks are used as inputs. The target dataset is composed of around 1200 real streaks.
For the refinement to be optimal, one needs to fine tune parameters like the learning rate l_r and the weight ρ given to the self regularization term. Three values have been tested for l_r: $10^{-3}, 10^{-4}$ and 10^{-5} and four values for ρ: $10^{-5}, 10^{-6}$ and 10^{-7} and 10^{-8}. The higher ρ is, the closer the refined image should be to the initial image. The number of epochs is 22. The loss being close for these different parameters, the comparison is really only possible with the segmentation studied in the next section. In figure 10 can be found an example of refined images for $l_r = 10^{-4}$ and in figure 11 an example of refined images for $l_r = 10^{-3}$.

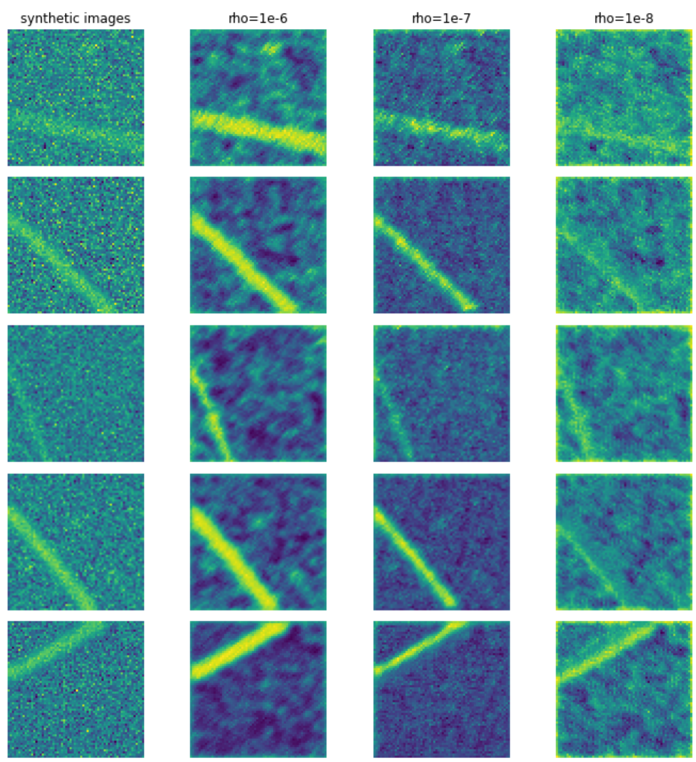

Figure 10: Refined images for different values of ρ and $l_r = 10^{-4}$.

It is noticeable that for each ρ the refiner blurs the entire patch. Even though the background is real, it is changed and seems to get darker, particularly when $\rho = 10^{-6}, 10^{-7}$. The streaks are brighter too. It seems like the image get closer to the last images of figure 4 for $\rho = 10^{-6}, 10^{-7}$. Except a big blurring, no big change is noticeable between the initial images and refined images with $\rho = 10^{-8}$.

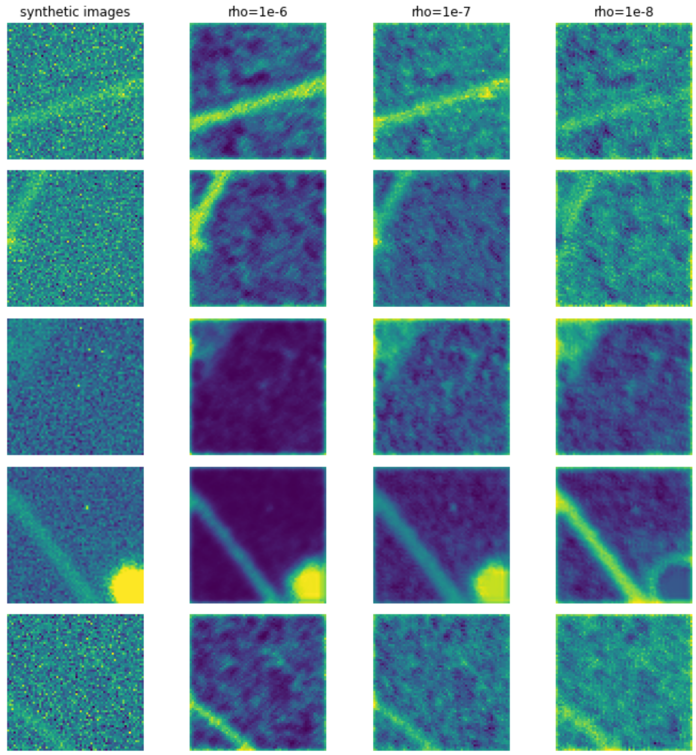

Figure 11: Refined images for different values of ρ and $l_r = 10^{-3}$.

For $l_r = 10^{-3}$, the refiner also blurs the entire patch for each ρ. The same comments as for $l_r = 10^{-4}$ can be made, except that the streak can get darker for $\rho = 10^{-6}$.

6 Semantic segmentation of satellite streaks with a UNet

The purpose of this section is to check if the refinement of the synthetic streaks does have an effect on the segmentation of real streaks. Indeed, if the synthetic images after refinement really got closer to real images, then we should observe a better result for the segmentation of real streaks when the network is trained with refined images.

6.1 What is a UNet?

The UNet is the network chosen to perform semantic segmentation of tracks. Its ability to generate high quality segmentation masks and its relatively simple architecture are the two main reasons explaining this choice. It [3] is a convolutional neural network that is designed to process images and predict the class of each pixel in the image. The name "UNet" comes from the fact that the model is built using a "U" shaped architecture (figure 12, with a series of convolutional and pooling layers in the encoder portion of the network and a series of transposed convolutional layers in the decoder portion of the network.The U-shaped architecture of the model allows it to capture both local and global context in the input image. The model is trained to predict the class label of each pixel in the input image, which can be used to segment the image into different objects or regions.

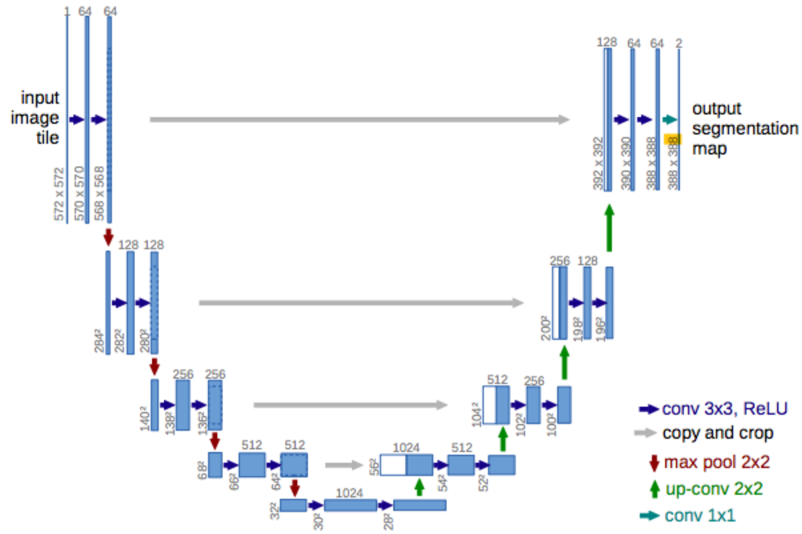

Figure 12: Scheme of the UNet [3].

6.2 Segmentation of satellite streaks

To verify the efficiency of the SimGAN model, a UNet has been trained with different datasets: with initial synthetic images and with refined images for the different values of ρ and l_r. The main purpose is to compare the results between the initial synthetic images and the refined ones but it is also to see which hyper parameters of the SimGAN model lead to the best result. The training has been done with a learning rate equal to: $lr_{unet} = 10^{-3}$ and a number of epochs equal to 12.

For the synthetic images, the targets are available, however targets of real streaks are not available. Thus, when testing each model with real streaks, it is not possible to quantify the accuracy. Only a visual comparison is possible.

In Figure 13, one can see the segmentation results for different values of ρ and for $l_r = 10^{-4}$.

In Figure 14, one can see the segmentation results for different values of ρ and for $l_r = 10^{-3}$.

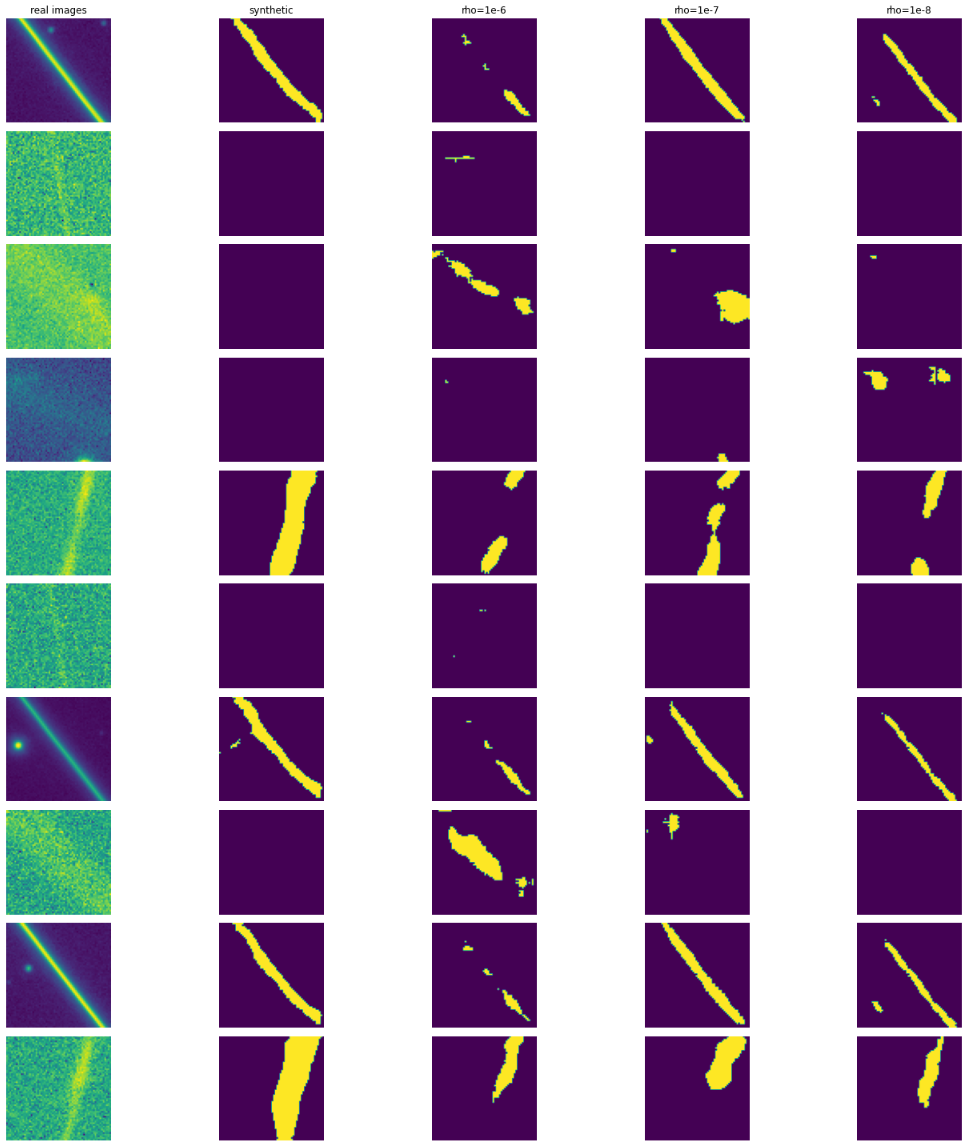

Figure 13: Segmentation results with UNet trained with refined images obtained with different values of ρ and $l_r = 10^{-4}$.

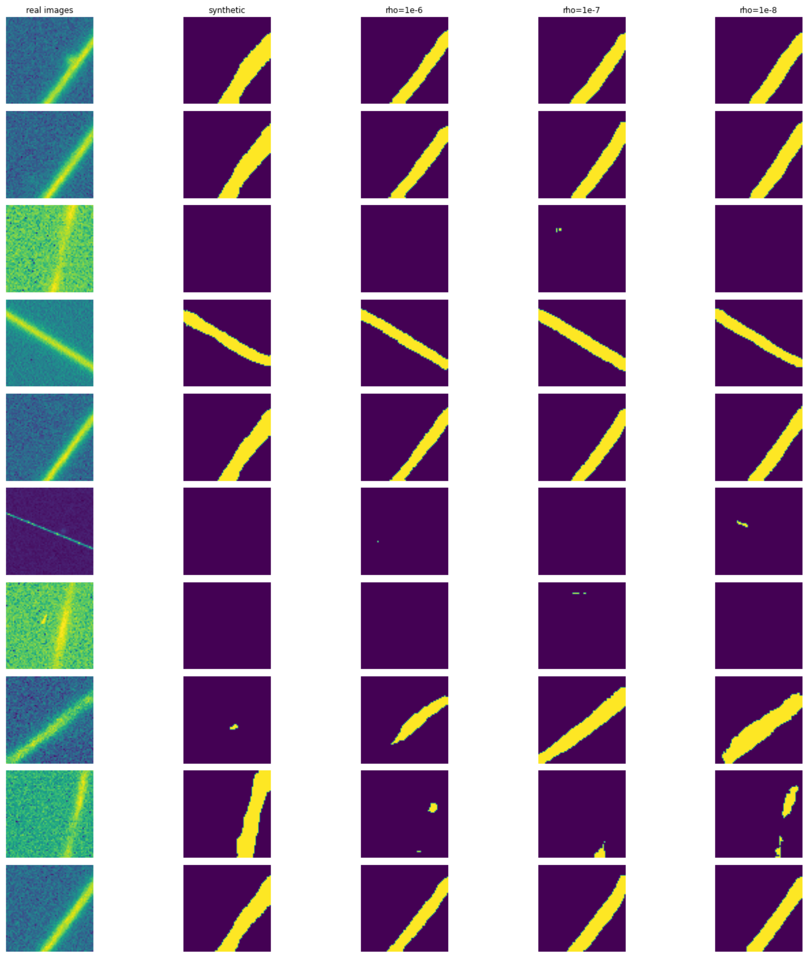

Figure 14: Segmentation results with UNet trained with refined images obtained with different values of ρ and $l_r = 10^{-3}$.

First of all, it is noteworthy that no configuration has succeeded in predicting the presence of a streak every time. However, all the test images have them. The segmentation is successful for clear and bright traces of average thickness for refined and non refined images (example in figure 15).

Figure 15: Segmentation of bright streak. In order: real images, synthetic, $\rho = 10^{-6}, 10^{-7}, 10^{-8}$.

Only with refined images and rho equal to 10^{-6}, the model perceives the presence of large and not very bright tracks (example in figure 16). However, it does not manage to segment the whole of these traces.

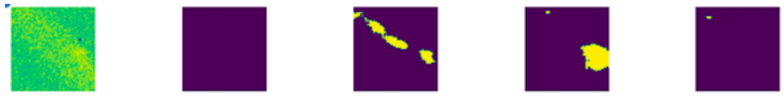

Figure 16: Segmentation of a large and dark streak. In order: real images, synthetic, $\rho = 10^{-6}, 10^{-7}, 10^{-8}$.

Moreover, all models seem to struggle to segment streaks which have very discontinuous intensities (example in figure 17). For the model trained with non refined images, if it manages to segment this kind of streak then the predicted width is too large. The others perceive the subtleties of the trace better but do not segment the entire length.

Figure 17: Segmentation of a streak with discontinuous intensity.In order: real images, synthetic, $\rho = 10^{-6}, 10^{-7}, 10^{-8}$.

6.3 F1 score for bright streaks

The targets of real images are not available. However for bright streaks, it is possible to obtain an idea of the shape of the streak by using filters. The first filter that is applied to the patches is a threshold filter, only pixel values higher than the threshold are kept. The value of the threshold is proper to each patch. Then, an opening filter is applied on each patch to get rid of the small zones not eliminated by the threshold but which are not part of the streak. This technique is only effective with traces that stand out from the background. It does not work with patches that have blobs, as these are not removed by the two successive filters.
It is thus possible to calculate the F1 score obtained with 32 real images and their obtained target. Most of the time these images verify the conditions: no blobs and bright streaks. Results of the segmentation for these streaks are visible for $l_r = 10^{-3}$ and $l_r = 10^{-4}$ respectively in figures 18 and 19 in the Annex. The second column corresponds to the grountruths obtained with the two filters just described. In the table below, one can see the F1 scores obtained for the different combinations of hyper parameters:

	synth	$\rho = 10^{-6}$	$\rho = 10^{-7}$	$\rho = 10^{-8}$
$l_r = 10^{-3}$	0.41	0.56	0.49	0.45
$l_r = 10^{-4}$	0.41	0.43	0.54	0.47

Table 1: Table of F1 scores of the segmentation with different datasets used for training the UNet.

7 Discussion

From the F1 scores, one can see that the results of the segmentation are better for bright streaks when the images are refined. The best score: $F1 = 0.56$ is obtained when the SimGAN model was trained with $l_r = 10^{-3}$ and the weight of the self regularization loss equal to $\rho = 10^{-6}$. From figures 18 and 19, one can see that the UNet trained with refined images segments far better thin streaks such as the ones of ligns 6,7,9. We can not deduce anything for steaks that do not really stand out from the background, but from figure 13 and 14 it seems that refinement or not the segmentation is not successful. Eventhough there is improvement compared to before refinement, 0.56 is still a low score.

One of the main reason why the SimGAN model was not successful may be that the number of target images, so the number of real images was too low. Indeed 1200 real patches were available, coming from 100 streaks. Thus, a lot of them were similar and the diversity of streaks were far too low for the model to learn more about the real distribution of the data. For example, in the original paper [1], they had more than 200 000 images as target. This may be a reason why the background seems to get darker with the refinement. More images, so more backgrounds, would have certainly allow a better generalization. Moreover, another weakness is that the target of refined images is the same before and after refinement, even though the streak might have been modified. Even if the streaks often did not completely change, the UNet might have struggle getting the real distribution. Finally, our interpretations are limited by the fact that targets of real images were not available, particularly for dark streak. Thus, there is definitely room for improvement.

8 Conclusion

The purpose of this project was to generate synthetic satellite streaks as close to reality as possible. To achieve this, the SimGAN model [1] was used to improve the reality of simple generated traces. Its particularity is that it learns from simulated and unsupervised images through adversarial learning. It has been trained for different values of learning rate and weight of the self regularization term in the refiner loss. The results of the segmentation with a UNet trained with refined images are slightly better compared to the results without refinement for bright streaks. However, no clear improvement has been observed for dark streaks, their segmentation still failing. One of the main reason is that the dataset of real streaks, targets in the SimGAN model, was far too small. To improve the obtained results, one would need far more real streaks, so as to ensure a better generalization.

References

[1] *Learning from Simulated and Unsupervised Images through Adversarial Training*, Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, Russ Webb , CVPR 2017, `https://openaccess.thecvf.com/content_cvpr_2017/papers/Shrivastava_Learning_From_Simulated_CVPR_2017_paper.pdf`

[2] *SimGAN- scheme*, `https://sh-tsang.medium.com/review-simgan-learning-from-simulated-and`

[3] *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Olaf Ronneberger, Philipp Fischer, Thomas Brox, Mai 2015 `https://arxiv.org/pdf/1505.04597.pdf`

[4] *Deep learning for satellite imagery via image segmentation* , Arkadiusz Nowaczynski,2017 `https://blog.deepsense.ai/deep-learning-for-satellite-imagery-via-image-segmentation/`

[5] *Optimal and Efficient Streak Detection in Astronomical Images*, Guy Nir, Barak Zackay, Eran O. Ofek, 2018, `https://arxiv.org/abs/1806.04204`

[6] *CodeUNet*,`https://keras.io/examples/vision/oxford_pets_image_segmentation/`

[7] *Code-SimGAN*, `https://github.com/mjdietzx/SimGAN`

9 Annex

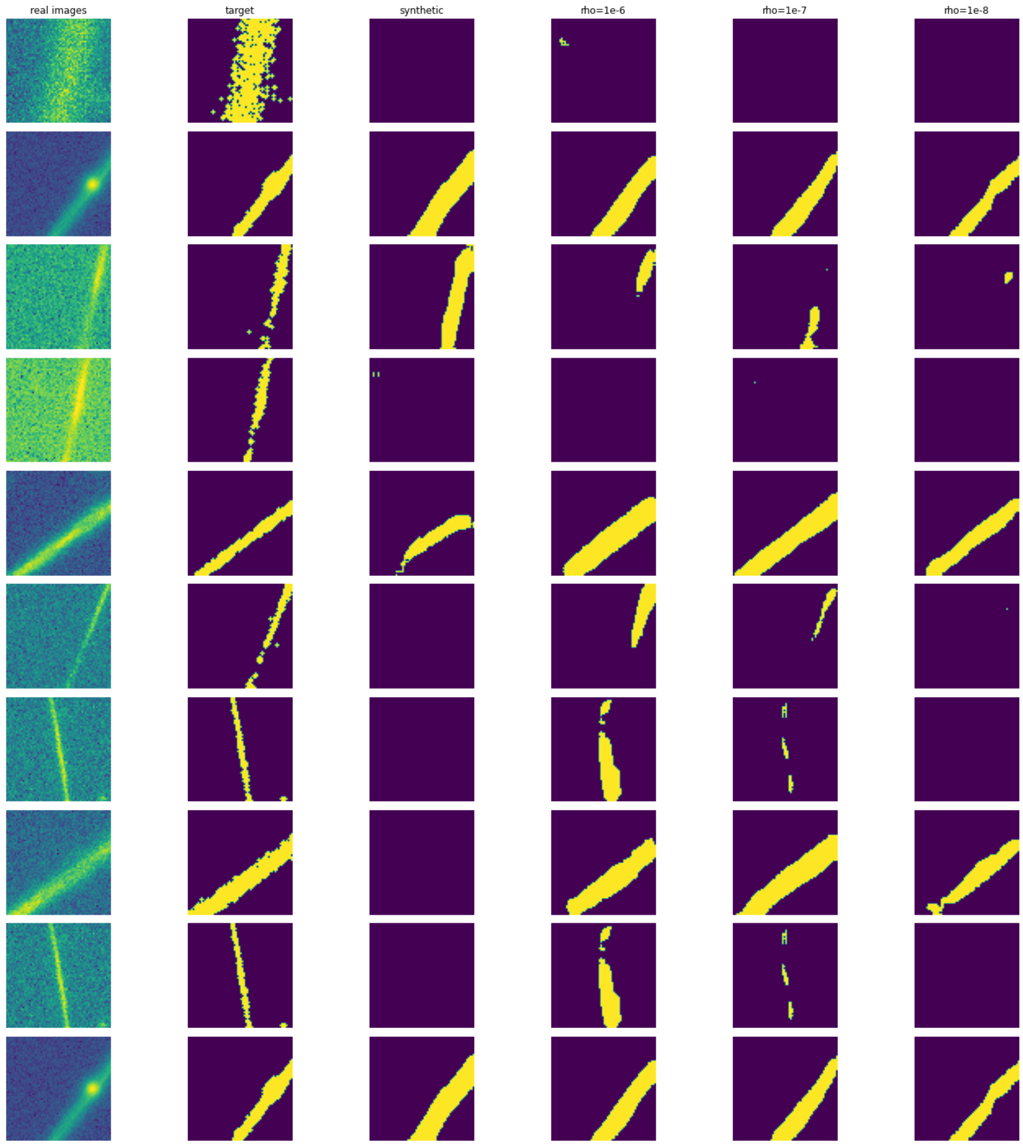

Figure 18: Segmentation results with UNet trained with refined images obtained with different values of ρ and $l_r = 10^{-3}$.

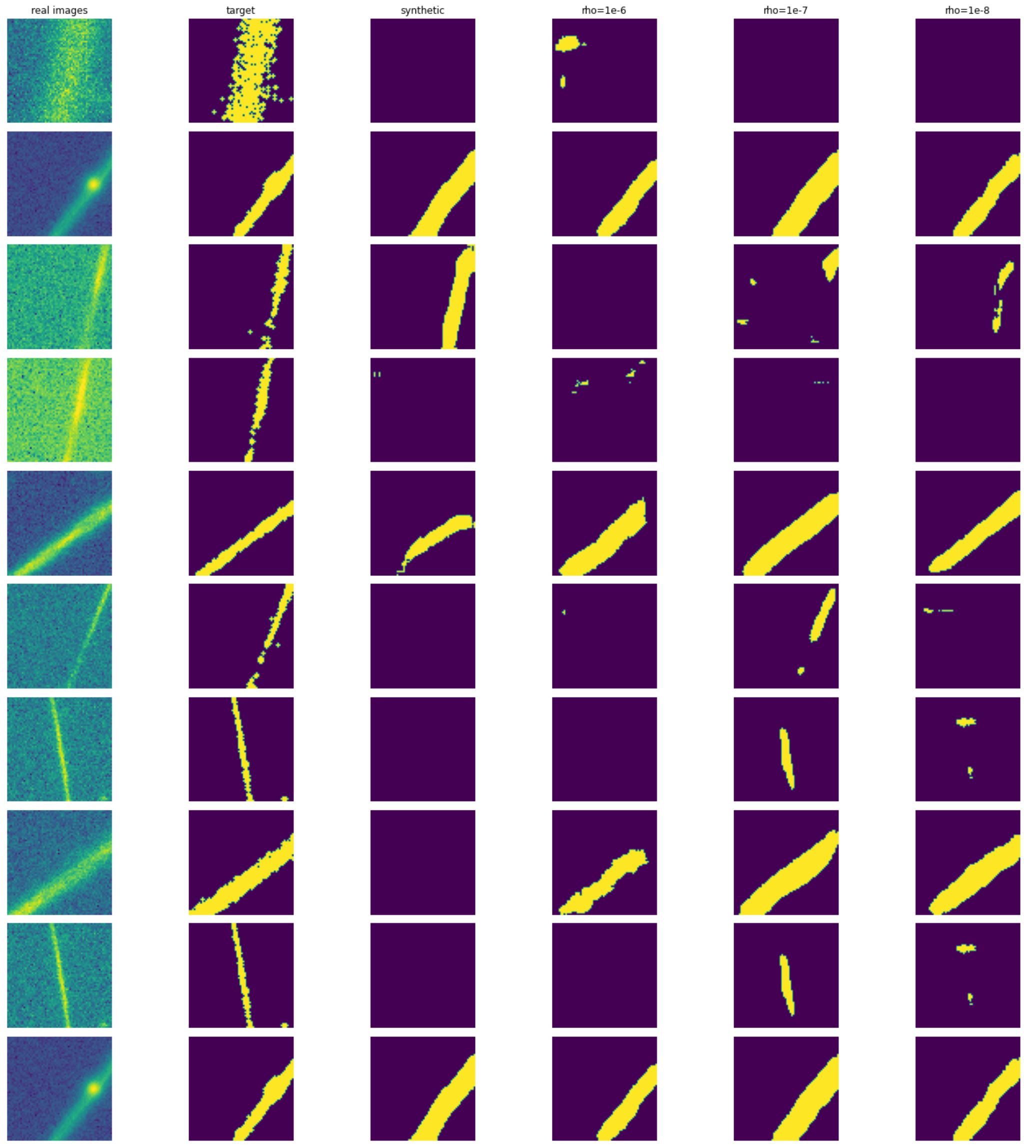

Figure 19: Segmentation results with UNet trained with refined images obtained with different values of ρ and $l_r = 10^{-4}$.