UNIL | Université de Lausanne
Faculté des géosciences
et de l'environnement

Travail de Bachelor

Generative adversarial networks for the super-resolution of satellite images

Lucas Schmutz

Sous la direction du Prof. Grégoire Mariethoz

Table of Contents

Table of Figures

Table of Tables

Abstract. This project explores the capabilities and limitations of generative adversarial network (GAN) based single image super-resolution (SISR) for the downscaling of satellite images. It demonstrates the need of specific training by confronting a model trained on Sentinel-2A images to its generically trained counterpart. Furthermore, it highlights the importance of a bespoke framework by not being able to downscale Landsat 8 images satisfactorily with the well-performing specifically trained model.

1 Introduction

Single image super-resolution (SISR) is a challenging task that aims to obtain a high-resolution (HR) image from a single low-resolution (LR) counterpart. It is a highly ill-posed problem because depending on the scaling factor, there might be a very large number of HR patches that correspond the same LR patch. Therefor the process of downscaling an LR patch has generally multiple solutions. The target of super-resolution is to reduce the distance between the super-resolved patch and the corresponding HR patch. (Agustsson & Timofte, 2017; Haut et al., 2018; Laghrib et al., 2017; Ledig et al., 2017; Timofte et al., 2018; X. Wang et al., 2019; Zhongyuan Wang et al., 2020)

In the context of remote sensing, super-resolution offers the possibility to obtain higher spatial resolution images from limited resolution sensors. These super-resolved (SR) images can in turn be used for a more accurate representation of environmental processes, land-cover use, biomes distribution and connectivity, or geomorphic processes to cite only a few examples. (Atkinson, 2013; Benediktsson et al., 2012; Haut et al., 2018; Thornton et al., 2006)

In this project, I will explore the capabilities of deep learning based SISR, and more precisely the generative adversarial network architecture with state-of-the-art model ESRGAN[1] (X. Wang et al., 2019).

[1] https://github.com/xinntao/ESRGAN

1.1 Deep learning and artificial neural networks

Deep learning is a set of machine learning methods attempting to model data with a high level of abstraction by using architectures composed of multiple layers of non-linear processing. As an example, artificial neural networks (ANNs) are computing systems that uses multiple connected layers of neurons to process data. A neuron is composed of a node (data holder) and connections to the previous and next layer. (Arnold et al., 2011; Deng & Yu, 2014; I. Goodfellow et al., 2016)

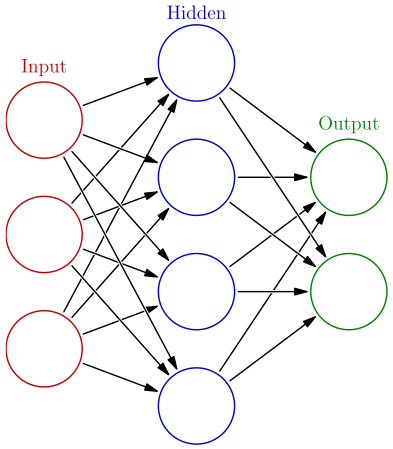

Figure 1 : Representation of an artificial neural network. (By Glosser.ca - Own work, Derivative of File: Artificial neural network.svg, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=24913461)

Each node can be modeled as: $A_j^m = f\{\sum_j A_j^{m-1} w_j + \beta\}$, where A represents a node, m the m^{th} layer of the network, j the j^{th} node of a layer, w the weight associated to the connection with a node of the previous layer, β a bias associated to the node and f the activation function used for this layer. These nodes are organized in connected layers to form a neural network as shown in Figure 1.

During training, ANNs learn by computing the loss function, which represent the distance between the network answer and the expected answer. Then using back-propagation, we compute the gradient of the loss function and adjust the weights with an optimizer like the stochastic gradient descent. (Arnold et al., 2011; Becker & LeCun, 1988; I. Goodfellow et al., 2016; Rumelhart et al., 1986)

1.2 Convolutional neural networks

Convolutional neural networks (CNNs) are a class of deep neural network that uses convolution to process efficiently input data with a grid-like topology, for example in digital images, with the use of kernels. It keeps a layered architecture but the weights are replaced by kernels of fixed size and the nodes by feature maps that are the result of the convolution between a kernel and the feature maps of the previous layer. (Deng & Yu, 2014; I. Goodfellow et al., 2016; Lecun, 1989)

CNN based SR was first proposed by Dong, Loy, He, & Tang (2014) with a simple 3 layers network and was improve in various ways for example with deeper network (Dong, Loy, He, et al., 2016; Lim et al., 2017), removing redundant parameters (Kim et al., 2016c), introducing skip connections by adding a residual architecture that is inspired by

cortical layer VI neurons (Huang et al., 2017; Thomson, 2010; Tong et al., 2017), by learning directly from the LR image instead of an interpolated image of the same resolution as the output resolution, greatly reducing computing cost (Kim et al., 2016a). However, because theses architecture uses pixel-wise mean squared error as their optimization target, it results in particularly blurry SR images that are not perceptually convincing. (Lugmayr et al., 2019; Takano & Alaghband, 2019; X. Wang et al., 2019; Zhongyuan Wang et al., 2020)

1.2.1 Generative adversarial networks

Generative adversarial networks (GANs) are a class of neural network where one of the terms of the loss function is another neural network. The generator (G) has the goal to produce data undiscernible from real data and the discriminator (D) aims at flagging the fake data appropriately (I. J. Goodfellow et al., 2014). They have wide variety of applications ranging from music (Guimaraes et al., 2017), to natural language processing (Yang et al., 2018), image synthesis (Jetchev et al., 2016; Jiang et al., 2020), and super-resolution (Ledig et al., 2017; Lugmayr et al., 2019; X. Wang et al., 2019; Zhongyuan Wang et al., 2020).

Since the publication of SRGAN (Ledig et al., 2017), GANs have gained a lot of attention for SISR. They provide results with finer textured details compared to CNNs based methods and are more perceptually convincing (Zhang et al., 2018). However, they are prone to produce more artefacts by creating high frequency textures where the HR might not have them (X. Wang et al., 2019).

1.3 Goals of this project

The goal of this project is to answer three main questions.

1. Are GANs based SR methods a viable technic for the super-resolution of satellite imagery?
2. Does a specifically trained model perform better than its generically trained counterpart?
3. if the answers to the two previous questions were satisfactory: Is the specifically trained model able to down-sample real-world data, namely Landsat 8 images.

In the next section I will present to you the methods I followed in order to answer these questions.

2 Methods

2.1 Dataset

In order to satisfy the goals of this project, I chose to use Sentinel-2 MSI: MultiSpectral Instrument, Level-2A, True Color Images, and only the TCI_R, TCI_G, TCI_B bands. (European Space Agency, 2015) This was justified by the design of the ESRGAN neural network made to handle standard RGB images. The Sentinel images offer a 10 meters resolution for the high-resolution image and 40 meters for the low-resolution image after a scaling by a factor of x4. This is lower than the 30 meters resolution of Landsat 8 RGB bands, offering the best chances at downscaling them later.

2.1.1 Downloading the base images using Google Earth Engine (GEE)

A total of 87 images where downloaded from GEE [1]. The process was partially manual as I wanted to ensure the quality of the images. Indeed, despite the use of filters for cloudiness and missing/defective/saturated pixels, some images were still very cloudy or missing a lot of pixels. Secondly, to maximize the diversity of objects the neural network would be trained on, I selected images on land from every populated continent and obtained a large variety of scenes. [2] Most of the images have a dimension of 10980*10980, which approximately corresponds to 12'000 km^2 tiles.

2.1.2 Generating the dataset

Following the framework of ESRGAN (X. Wang et al., 2019), SRGAN (Ledig et al., 2017) and the 2018 Perceptual Image Restoration and Manipulation (PIRM) Challenge (Blau et al., 2018), I wrote a MATLAB script[3] that takes the input images downloaded in the previous step and cuts them in tiles of the desired dimensions (in this case 480*480 pixels) to form the ground truth data. The low-resolution pairs are computed using the imresize function and a scale factor of 0.25 with bicubic interpolation as well as an anti-aliasing filter are used to downsample.

A total of 41'052 pairs were generated and then divided in validation and training datasets. This was done by randomly extracting 8200 images from the respective GT and LR folders using the shuf[4] command from the GNU Core Utilities.

[1]The code is accessible at this address:
https://code.earthengine.google.com/6e67a98c76181899166f1883026ac59f?noload=1
[2] The base images are accessible at /datasets/baseimg
[3] The code is found in /codes/resize.m
[4] Source code available at : https://github.com/coreutils/coreutils/blob/master/src/shuf.c

Therefore, the dataset[1] used in this project is composed of two pairs of folders, one for training (32'852 images) and one for validation/testing (8'200 images).

2.2 Training

In order to assess the ability of GANs to downscale satellite images, I chose state-of-the-art, and winner of the PIRM 2018 Challenge (Blau et al., 2018) with the best perceptual index, model of Wang et al. (2019) : ESRGAN. The complete code[2] was downloaded from GitHub but has recently been merged in a more complete image editing toolbox[3]. However, the experiments in this project were only performed with the old version of the code for continuity.

2.2.1 ESRGAN

The network architecture used in this project is based on SRGAN (Ledig et al., 2017) and improves upon it to further increase perceptual quality (X. Wang et al., 2019). The general architecture of the generator is presented on Figure 2 and is based on the SRResNet (Ledig et al., 2017). The major part of the computation is done in the LR feature space which reduces computational complexity massively. The basic blocks are illustrated in Figure 3 and make use of dense connections as well as multi-level residual network (skip connections). The kernels used in this architecture are 3x3 (X. Wang et al., 2019) .

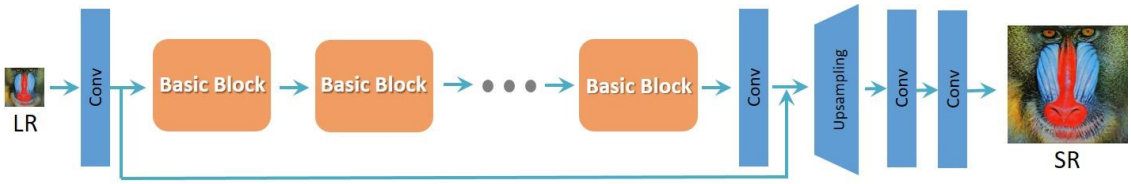

Figure 2 : Representation of the architecture of the generator based on SRResNet. (X. Wang et al., 2019)

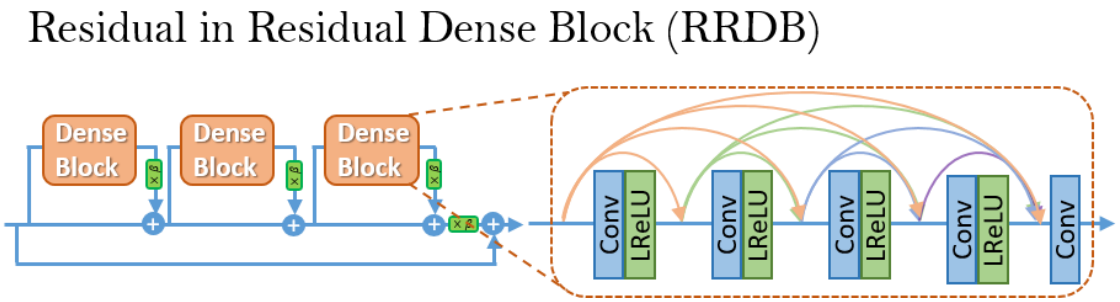

Figure 3 : Representation of the Residual in Residual Dense Block used as basic blocks in SRResNet. (X. Wang et al., 2019)

[1] The dataset is accessible at: /datasets/
[2] The code is accessible at: /mmsr-master
[3] https://github.com/open-mmlab/mmediting

The adversarial training is based on a relativistic average GAN (RaGAN) that speeds up the training considerably, while also stabilizing it compared to standard GANs (Jolicoeur-Martineau, 2019).

2.2.2 Hardware and software setup

The various experiments took place on the octopus [1] supercomputer of the Swiss Geocomputing Centre at the university of Lausanne. Depending on the availability of the nodes, the models were trained either on NVIDIA Titan X or NVIDIA V100 NVLink graphics processing unit. Software wise, the cluster uses CentOS 6.9 and the models were trained using python 3.7.4 and CUDA 9.0. GPU, multi-GPU parallelization was tried but not used as it did slow down the training.

2.2.3 Training process

Due to instabilities and bugs in the code, the training process was less streamlined than expected. They caused most of the runs to not complete and the root of the problem is still not clear yet. Moreover, despite the possibility to set the desired number of iterations in the configuration training file [2] the training systematically stopped at 400'000 iterations. The workaround used here was to set the learning rate accordingly to the end of the last run, and to use the final generator of the last run as the pretrained model for the next.

For the final four models that will be presented in the first section of the results, a pretrained generator that successfully finished 400'000 iterations was used. In order to explore the influence of training parameters I used three different batch size : 8, 16, 32; as well as one model to test the influence of ADAM (Kingma & Ba, 2015) exponential decay rate : beta1 was set to 0.99, for both the discriminator as well as the generator, instead of 0.9. It was planned to have more models trained for each parameter change to obtain a better understanding of the influence of these on the performance. However, due to most of them failing to complete the training, only the performance of the four that completed the training is presented.

2.3 Performance evaluation and metrics

As is typically done in the field of SISR , I used PSNR and SSIM to measure the quality of the SR images (Bosch et al., 2018; Chu et al., 2019; Dong, Loy, He, et al., 2016; Dong,

[1] https://wp.unil.ch/geocomputing/octopus/
[2] /mmsr-master/codes/options/train/train_ESRGAN.yml

Loy, & Tang, 2016; Haris et al., 2018; Haut et al., 2018; Hayat, 2018; Kim et al., 2016b, 2016c; Ledig et al., 2017; Lim et al., 2017; Lugmayr et al., 2019; Tong et al., 2017; X. Wang et al., 2019; Zhongyuan Wang et al., 2020). However, according to Zhang, Isola, Efros, Shechtman, & Wang (2018), both metrics fail to evaluate the perceptual quality of images and as the network architecture is designed to maximize it in parallel of pixel-wise MSE. To measure this performance as well, I added the Learned Perceptual Image Patch Similarity (LPIPS)[1] (Zhang et al., 2018).

The PSNR (peak signal-to-noise ration) is derived from the mean square error (MSE). It has a range of [0, +∞] measured in dB and higher is better (Huynh-Thu & Ghanbari, 2008). The SSIM (structural similarity) is a perception-based model that contrary to the PSNR measure relative error. It has a range of [-1, 1] and higher is better (Zhou Wang et al., 2004). The LPIPS (learned perceptual image patch similarity) is a deep learning based metrics that most closely represents human perception. It uses deep features map and has a range of [0, 1], where lower is better (Zhang et al., 2018) .

2.4 Testing on Landsat 8 images

Finally, to challenge the performance of the specifically trained model with real world data, I downscaled Landsat 8 Surface Reflectance Tier 1 images. The images were downloaded from GEE[2] and then split using the MATLAB script previously used for the Sentinel-2A dataset. They were then downscaled using the specifically trained model and a scaling factor of 4, consequently super-resolving to a ground sampling distance of 7.5m.

The results are presented in section 3.3 and no performance metric is used as the ground truth data is not available.

3 Results

3.1 The influence of the training parameters

In the first part of this project, I trained multiple models in parallel with some variations in the training parameters in order to gain a deeper understanding of the behavior of the neural network and with the goal of finding the best performing setup.

[1] https://github.com/j/PerceptualSimilarity
[2]The script was kindly provided by Dr. Mathieu Gravey:
https://code.earthengine.google.com/6e67a98c76181899166f1883026ac59f?noload=1

The four models presented in Table 1 all started from the same pretrained network, with only the batch size changing for the three first one, and a batch size of 32 and the $beta_1$ of the ADAM optimizer set to 0.99 for the last one.

As we can quantitively see in Table 1, there is not one clear winner over the range of metrics. Moreover, the differences between the models are very small, which means they perform similarly. On Figure 4 we can qualitatively see the performance of the four models compared to a bicubic interpolation (top left) and the HR (bottom left). We can observe that the models give slightly different SR images but that they all have artefacts compared to the HR image. This confirms the quantitative results and the totality of the images can be accessed in the supplementary work folder[1].

Table 1: Average PSNR, SSIM and LPIPS score for four different training parameters

	PSNR [dB] ↑	SSIM ↑	PSNR_Y [dB] ↑	SSIM_Y ↑	LPIPS ↓
Sentinel-2A BATCH 8	**27.8789**	**0.7252**	29.7988	0.7582	0.1079
Sentinel-2A BATCH 16	27.8012	0.7239	**29.8063**	0.7577	0.1088
Sentinel-2A BATCH 32	27.6090	0.7208	29.6168	0.7547	0.1124
Sentinel-2A beta1= 0.99	27.4829	0.7251	29.4459	**0.7596**	**0.1055**

Figure 4 : Comparison of the performance of the four models. Image 36379. From left to right and top to bottom: bicubic, batch size =8, batch size = 16, batch size = 32 and batch size = 32/beta1 = 0.99

[1] /results/

3.2 Generically vs specifically trained

In this section, I will present the performance of the generically and specifically trained models on the Sentinel-2A validation dataset[1]. The specifically trained network has been trained using the same training parameters as the generically trained network. However, it has been trained from the beginning only on Sentinel-2A images. The generically trained network is the one provided with the ESRGAN code and has been trained on various datasets as explained in the related article (X. Wang et al., 2019). The two models have the exact same architecture and have been trained with the same hyperparameters. Consequently, the only variable between them is the training datasets.

All the results are found the folder of the supplementary material.[2]

Table 2 demonstrates that specific training outperforms generic training across each metrics used here. Interestingly, the generically trained network is outperformed by bicubic interpolation in PSNR and SSIM showing how crucial training data is. However, both models massively outperform bicubic interpolation when measured with LPIPS, which indicates that both produce more perceptually attractive results.

In the following lines, gSRi refers to images super-resolved by the generically trained model and sSRi to the images from the specifically trained model.

Table 2 : Performance comparison of bicubic interpolation, generic training, and specific training

	PSNR [dB] ↑	SSIM ↑	PSNR_Y [dB] ↑	SSIM_Y ↑	LPIPS ↓
Bicubic	27.5905	0.7193	NC	NC	0.4616
Sentinel-2A GENERIC	27.1807	0.7013	29.1101	0.7358	0.1425
Sentinel-2A SPECIFIC	**27.8012**	**0.7239**	**29.8063**	**0.7577**	**0.1088**

[1] /datasets/S2A_480LR_val
[2] /results/

On Figure 5, we can see that the generically trained model creates a lot of artifacts. The central road is reduced to a fine, disconnected, and to white line. The image looks sharper, but more fake. On the other hand, the image downscaled by the specifically trained network looks blurrier but a lot closer to the HR. Roads are better connected and the shape and size of buildings more accurate.

Figure 5 : Detail cropped from image 53, from left to right and top to bottom: bicubic, generic, ground truth, specific

On Figure 6, we can see that both networks failed to interpolate the circuit roads but the sSRi displays less noise and less high-frequency textures. The patches of forest are also more accurate in the sSRi.

Figure 6 : Detail cropped from image 219, from left to right and top to bottom: bicubic, generic, ground truth, specific

On Figure 7, we can see that the gSRi has over-sharpened the main road while smaller ones have disappeared. The city blocks have merged and gained a scale-like texture. The gSRi has better connections for the road network and therefore the city blocks are more distinct, however the saturated pixels in the HR image means even less information has been initially captured.

Figure 7 :Detail cropped from image 20977, from left to right and top to bottom: bicubic, generic, ground truth, specific

On Figure 8, we can again observe fake textures in the gSRi and one of the track field has almost disappeared. The rectangle neighborhood on the left side of the image are not even discernable.

Figure 8 : Detail cropped from image 36379, from left to right and top to bottom: bicubic, generic, ground truth, specific

On Figure 9, we can see that the fields in the sSRi are very well recreated. The borders between fields are close to the HR image and the major roads of the village have been well resolved and are well connected. The gSRi displayed hallucinated forms instead of a village and the roads have changed in color and have either been overly sharpened or disappeared.

Figure 9 : Detail cropped from image 17709, from left to right and top to bottom: bicubic, generic, ground truth, specific

On Figure 10, we can again see that the sSRi is well defined. The fields are well recreated and the rivers close to the HR image. The gSRi displays over sharpening and deformed riverbanks. Some field have merged and textures have been hallucinated in the crops.

Figure 10 : Detail cropped from image 40733, from left to right and top to bottom: bicubic, generic, ground truth, specific

3.3 Using the specifically trained model to downscale Landsat 8 images

As a last experiment, I used the specifically trained model to downscale Landsat 8 images. As the previous results were relatively satisfying, this was done to validate that the model is able to perform on real world data that have a similar spatial resolution to the LR images it has been trained on.

On figure 9, we can see the results of the downscaling of Landsat 8 images. The SR images display a lot of undesirable artefacts. It looks like the aliasing has been enhanced and weird grid-like pattern appeared. The images are less blurry at the cost of accuracy and realistic patterns.

Figure 11 : Landsat 8 image on the right, super-resolved image on the right. Images 2 and 12 in the result folder.

4 Discussion

4.1 Modifying the hyper-parameters

The first experiment showed us, with a limited sample, that the batch-size and the $beta_1$ hyperparameters did not drastically change the final model for this network architecture. This may indicate a good stability of the architecture, meaning that even with different training parameters the models still converge to the same neighborhood of the N-dimensional space where the answers reside. Therefore, the exploration of different training parameters was probably a waste of resources in this case. However, both parameters influence the speed of learning and the stability of the training process (Deng & Yu, 2014; I. Goodfellow et al., 2016). Consequently, it probably is a good idea to experiment with the hyper-parameters relevant to the architecture to find values suited for the particular use case.

4.2 The influence of the training dataset

The second experiment demonstrated how crucial the training dataset is to obtain good SR results, as shown by Takano & Alaghband (2019) in a similar experiment. This indicates that the network gains in specificity towards the data we feed it, and the fact that the state-of-the art generically trained network did so poorly suggests it does not learn to super-resolve in the general sense. This can be conceptualized by modeling the neural network as an abstract machine that takes LR images from the LR space and maps them in the HR space.[1] The training process corresponds to the adjustment we make to this machine in order to reduce the distance between the SR image and the HR image in the HR space. In our case, the generically trained network learned to map some regions of the LR space to their corresponding region in the HR space but the lack of specificity and the limited memory to store knowledge[2] about the mapping meant he could not map the specific data as well as the specifically trained network. This notion of specificity of a trained network also implies that this architecture does not learn low level features that would be underlying in any dataset. This means that at the current state of GAN based super-resolution, one should try to use a training dataset that closely represents the data he wishes to downscale.

[1] The LR images are described as a single point in a N-dimensional space where N is equal their number of pixels multiplied by the number of layers, the same is applicable for the HR M-dimensional space where the SR images also resides as they have the same dimensions as the HR images.
[2] This limited memory corresponds to the 16,697,987 parameters, the weights and biases of the network.

4.3 The poor performance with Landsat 8

For this last experiment we saw that the model did not perform well with the Landsat8 images, despite having good performance with the sentinel LR images. The downsampled images have probably not much use with the artifacts we observe. But this poor performance could teach us something about the setup of the initial experiment and about the framework that is standard in other deep learning-based SR methods. In this experiment, I used the standard imresize matlab function that uses bicubic interpolation and an anti-aliasing prefilter. The prefiltering is probably[1] a low-pass filter in order to sample above the Nyquist rate to eliminate aliasing (Landau, 1967). The bad quality of the results might suggest that the network learned to reverse the imresize function and that the bicubic interpolation is not a good representation of the LR Landsat images. This implies that if LR images are produced from HR counterpart to create the dataset, we should try to find a method that best represents the data we are training the neural network for. This might also mean to induce distortion and noise to best mimic the sensor the data is captured on. In the case of satellite imagery, we could also train the network on pair of images for two different satellite, one being lower resolution than the other. However, this solution would require such pair to exist, meaning both satellites should have taken a picture of the area at the same time. Another method tried by Lugmayr et al. (2019) is training the network with an unsupervised approach, which eliminates the need of LR/HR pairs.

4.4 Deep learning based SR

Deep learning based SR has seen a lot of development in the recent years and some commercial applications have already started to appear. For example, in February 2019, NVIDIA has launched DLSS, short for deep learning super sampling. It consists of neural networks trained by NVIDIA for some major games. They ship with the GPU drivers and allow to render a better image while reducing the load on the GPU (NVIDIA, 2018). Another, example is how modern TVs upscale the LR signal to the HR display with their proprietary trained network and neural processing unit in their SoCs (El-Khamy et al., 2019). However the research community and the industry have mostly emphasized on perceptual quality in order to satisfy the human vision and therefore most methods have not focused on the specific needs of remote sensing applications. This means that

[1] The function used in imresize is an internal function and therefore no detailed documentation about the implementation is published

researcher that need higher resolution images should find a suited architecture, train it specifically and most importantly to quantify the biases induced by it.

4.5 Limits and future works

The major limit of this project was the instabilities and bugs during the training process that caused most of the runs to fail. It greatly limited the sample size of the first experiment and the number of iterations of the specifically trained network as I would have liked to train it further. Secondly, now knowing that the network does not perform well with Landsat images, it would be interesting to see if generating the LR images with other methods would yield better results on real-world data and therefore confirm the hypothesis that model learned to reverse the imresize function. Thirdly, it would be interesting to explore the design of loss functions for the specific needs of remote sensing as they are the prime driver of learning for neural networks. Finally, the Volta node offers massive computing power that this project did not use fully. This means that bigger and more complex network could be trained, for example using larger kernel that might capture more complex features in the images.

5 Conclusion

In conclusion, I think that GAN based SR methods may offer great opportunities for satellite images and remote sensing research. However, the field still needs a lot of work in order to find suitable methods and their respective limits. Moreover, specific training datasets are a must to provide the best results by more closely mapping between the LR and HR space. GAN based SISR methods are still at their first year of development and the rapid increase in computing power will offer the possibility to train larger, more capable neural networks.

6 Acknowledgment

I would like to thank professor Grégoire Mariethoz for his confidence and the initial idea which goes far beyond the skills provided by this bachelor's program. He gave me valuable advice that guided this project. I also want to thank Dr. Mathieu Gravey for his endless support and his teaching, as well the review of this report. Finally, I want to thank my family and friends for their inputs and support.

7 References

Agustsson, E., & Timofte, R. (2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, *2017-July*, 1122–1131. https://doi.org/10.1109/CVPRW.2017.150

Arnold, L., Rebecchi, S., & Chevallier, S. (2011). An Introduction to Deep Learning. *European Symposium on Artificial Neural Networks (ESANN), January*.

Atkinson, P. M. (2013). Downscaling in remote sensing. *International Journal of Applied Earth Observation and Geoinformation*, *22*(1), 106–114. https://doi.org/10.1016/j.jag.2012.04.012

Becker, S., & LeCun, Y. (1988). Improving the Convergence of Back-Propagation Learning with Second Order Methods. In *Proceedings of the 1988 Connectionist Models Summer School* (pp. 29–37).

Benediktsson, J. A., Chanussot, J., & Moon, W. M. (2012). Very High-resolution remote sensing: Challenges and opportunities [point of view]. *Proceedings of the IEEE*, *100*(6), 1907–1910. https://doi.org/10.1109/JPROC.2012.2190811

Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., & Zelnik-Manor, L. (2018). *The 2018 PIRM challenge on perceptual image super-resolution*. https://doi.org/10.1007/978-3-030-11021-5_21

Bosch, M., Gifford, C. M., & Rodriguez, P. A. (2018). Super-Resolution for Overhead Imagery Using DenseNets and Adversarial Learning. *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, *2018-Janua*, 1414–1422. https://doi.org/10.1109/WACV.2018.00159

Chu, X., Zhang, B., Ma, H., Xu, R., & Li, Q. (2019). *Fast, Accurate and Lightweight Super-Resolution with Neural Architecture Search*. http://arxiv.org/abs/1901.07261

Deng, L., & Yu, D. (2014). Deep Learning Methods and Applications. *Foundations and Trends® in Signal Processing*, *7*(3–4), 197–387. https://doi.org/10.1007/978-981-13-3459-7_3

Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, *38*(2), 295–307. https://doi.org/10.1109/TPAMI.2015.2439281

Dong, C., Loy, C. C., He, K., & Tang, X. (2014). Learning a deep convolutional network for image super-resolution. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8692 LNCS*(PART 4), 184–199. https://doi.org/10.1007/978-3-319-10593-2_13

Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9906 LNCS*, 391–407. https://doi.org/10.1007/978-3-319-46475-6_25

El-Khamy, M., Lee, J., & Ren, H. (2019). *SYSTEM AND METHOD FOR DEEP LEARNING IMAGE SUPER RESOLUTION* (Patent No. US 10,489,887 B2). United States Patent.

European Space Agency. (2015). *SENTINEL-2 User Handbook* (Issue 1).

Goodfellow, I., Bengio, Y., & Courville, Aa. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *3*(January), 2672–2680. http://www.github.com/goodfeli/adversarial

Guimaraes, G. L., Sanchez-Lengeling, B., Outeiral, C., Farias, P. L. C., & Aspuru-Guzik, A. (2017). *Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models*. http://arxiv.org/abs/1705.10843

Haris, M., Shakhnarovich, G., & Ukita, N. (2018). Deep Back-Projection Networks For Super-Resolution. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1664–1673. http://arxiv.org/abs/1803.02735

Haut, J. M., Fernandez-Beltran, R., Paoletti, M. E., Plaza, J., Plaza, A., & Pla, F. (2018). A new deep generative network for unsupervised remote sensing single-image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, *56*(11), 6792–6810. https://doi.org/10.1109/TGRS.2018.2843525

Hayat, K. (2018). Multimedia super-resolution via deep learning: A survey. *Digital Signal Processing: A Review Journal*, *81*, 198–217. https://doi.org/10.1016/j.dsp.2018.07.005

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, *44*(13), 800–801. https://doi.org/10.1049/el:20080522

Jetchev, N., Bergmann, U., & Vollgraf, R. (2016). *Texture Synthesis with Spatial Generative Adversarial Networks*. http://arxiv.org/abs/1611.08207

Jiang, W., Liu, S., Gao, C., Cao, J., He, R., Feng, J., & Yan, S. (2020). *PSGAN: Pose and Expression Robust Spatial-Aware GAN for Customizable Makeup Transfer*. 5193–5201. https://doi.org/10.1109/cvpr42600.2020.00524

Jolicoeur-Martineau, A. (2019). The relativistic discriminator: A key element missing from standard GaN. *7th International Conference on Learning Representations, ICLR 2019*.

Kim, J., Lee, J. K., & Lee, K. M. (2016a). Accurate image super-resolution using very deep convolutional networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 1646–1654. https://doi.org/10.1109/CVPR.2016.182

Kim, J., Lee, J. K., & Lee, K. M. (2016b). Accurate image super-resolution using very deep convolutional networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 1646–1654. https://doi.org/10.1109/CVPR.2016.182

Kim, J., Lee, J. K., & Lee, K. M. (2016c). Deeply-recursive convolutional network for image super-resolution. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 1637–1645. https://doi.org/10.1109/CVPR.2016.181

Kingma, D. P., & Ba, J. L. (2015, December 22). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/abs/1412.6980v9

Laghrib, A., Hakim, A., & Raghay, S. (2017). An iterative image super-resolution approach based on Bregman distance. *Signal Processing: Image Communication*, *58*(June), 24–34. https://doi.org/10.1016/j.image.2017.06.006

Landau, H. J. (1967). Sampling, Data Transmission, and the Nyquist Rate. *Proceedings of the IEEE*, *55*(10), 1701–1706. https://doi.org/10.1109/PROC.1967.5962

Lecun, Y. (1989). Generalization and Network Design Strategies. In R. Pfeifer, Z. Schreter, F. Fogelman, & L. Steels (Eds.), *Connectionism in Perspective*. Elsevier.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 105–114. https://doi.org/10.1109/CVPR.2017.19

Lim, B., Son, S., Kim, H., Nah, S., & Lee, K. M. (2017). Enhanced Deep Residual Networks for Single Image Super-Resolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, *2017-July*, 1132–1140. http://arxiv.org/abs/1707.02921

Lugmayr, A., Danelljan, M., & Timofte, R. (2019). Unsupervised Learning for real-world super-resolution. *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, 3408–3416. https://doi.org/10.1109/ICCVW.2019.00423

NVIDIA. (2018). NVIDIA Turing GPU. *White Paper*. NVIDIA-Turing-Architecture-Whitepaper.pdf

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Takano, N., & Alaghband, G. (2019). *SRGAN: Training Dataset Matters*. 1–7. http://arxiv.org/abs/1903.09922

Thomson, A. M. (2010). Neocortical layer 6, a review. In *Frontiers in Neuroanatomy*

(Vol. 4, Issue MARCH, p. 13). Frontiers. https://doi.org/10.3389/fnana.2010.00013

Thornton, M. W., Atkinson, P. M., & Holland, D. A. (2006). Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping. *International Journal of Remote Sensing*, *27*(3), 473–491. https://doi.org/10.1080/01431160500207088

Timofte, R., Agustsson, E., Gool, L. Van, Yang, M. H., Zhang, L., Lim, B., Son, S., Kim, H., Nah, S., Lee, K. M., Wang, X., Tian, Y., Yu, K., Zhang, Y., Wu, S., Dong, C., Lin, L., Qiao, Y., Loy, C. C., … Guo, Q. (2018). NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, *2017-July*. https://doi.org/10.1109/CVPRW.2017.149

Tong, T., Li, G., Liu, X., & Gao, Q. (2017). Image Super-Resolution Using Dense Skip Connections. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-Octob*, 4809–4817. https://doi.org/10.1109/ICCV.2017.514

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Loy, C. C. (2019). ESRGAN: Enhanced super-resolution generative adversarial networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11133 LNCS*, 63–79. https://doi.org/10.1007/978-3-030-11021-5_5

Wang, Zhongyuan, Jiang, K., Yi, P., Han, Z., & He, Z. (2020). Ultra-dense GAN for satellite imagery super-resolution. *Neurocomputing*, *398*, 328–337. https://doi.org/10.1016/j.neucom.2019.03.106

Wang, Zhou, Wang, Z., Bovik, A. C., Sheikh, H. R., Member, S., Simoncelli, E. P., & Member, S. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, *13*, 600--612. https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.5689

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., & Berg-Kirkpatrick, T. (2018). Unsupervised text style transfer using language models as discriminators. *Advances in Neural Information Processing Systems*, *2018-Decem*, 7287–7298.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The Unreasonable

Effectiveness of Deep Features as a Perceptual Metric. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *1*, 586–595. https://doi.org/10.1109/CVPR.2018.00068