

# **Technische Universität Berlin**

Faculty of Electrical Engineering and Computer Science  
Dept. of Computer Engineering and Microelectronics  
**Remote Sensing Image Analysis Group**



---

## **Compression of Remote Sensing Images based on Generative Adversarial Networks**

---

Bachelor of Science in Computer Science

14th of February, 2022

**Alisa Korytova**

Matriculation Number: 386435

**Supervisor:** Prof. Dr. Begüm Demir, Prof. Dr. Olaf Hellwich


**Advisor:** Dr. Nimisha T M

## Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Sämtliche benutzten Informationsquellen sowie das Gedankengut Dritter wurden im Text als solche kenntlich gemacht und im Literaturverzeichnis angeführt. Die Arbeit wurde bisher nicht veröffentlicht und keiner Prüfungsbehörde vorgelegt.

I hereby declare that the thesis submitted is my own, unaided work, completed without any unpermitted external help. Only the sources and resources listed were used. The independent and unaided completion of the thesis is affirmed by affidavit:

Berlin, Date 14.02.2022



.....

*Alisa Korytova*

## Abstract

This thesis proposes novel architectures for spatio-spectral end-to-end compression of remote sensing (RS) archives. In details, we explore architecture improvements over the High-Fidelity Generative Image Compression (HiFiC) network for spatio-spectral compression using: i) Squeeze-and-Excitation (SE) blocks; and ii) 3D convolutions. HiFiC has proven accuracy on high-resolution RGB images, but fail to recognize dependencies between channels and only performs spatial compression that is very limited for multispectral RS data. Bands of multispectral images still may be redundant, and removing some information from those bands or downscaling them will improve the compression efficiency without impacting the reconstruction. We used SE blocks for channel attention to understand the spectral redundancy. Such block evaluates and weights each channel, based on its feature responses and interdependencies. 3D convolutions have also shown great results in video compression, and considering bands as the third dimension, they can be applied to multispectral data. The explored architecture improvements with SE block and 3D convolutions for spatio-spectral compression are thus capable to tap into this spectral redundancy and achieve slightly better compression on RS archives. Performance analysis on BigEarthNet dataset shows that the overall quality of decompressed images increased compared to spatial compression on all bit-rate ranges without dramatically affecting the compression-rate using the proposed architecture improvements over a naive HiFiC network. The fact that the proposed architecture improvements are easy to apply to any already existing compression network (especially SE blocks), makes them even more beneficial. Our success indicates that more complicated techniques can be employed in the future to achieve even better results.

## Zusammenfassung

In dieser Bachelorarbeit werden neuartige Architekturen für die räumlich-spektrale (spatio-spectral) End-to-End-Komprimierung von Remote Sensing (RS) Data vorgeschlagen. Im Detail untersuchen wir Architekturverbesserungen gegenüber dem High-Fidelity Generative Image Compression (HiFiC)-Netzwerk für räumlich-spektrale Komprimierung unter Verwendung von: i) Squeeze-and-Excitation (SE)-Blöcken; und ii) 3D-convolutions. HiFiC hat seine Genauigkeit bei hochauflösenden RGB-Bildern unter Beweis gestellt, erkennt jedoch keine Abhängigkeiten zwischen den Bändern und führt nur eine räumliche Komprimierung durch, die für multispektrale RS-Daten sehr begrenzt ist. Die Bänder (Kanäle) von Multispektralbildern können immer noch redundant sein, und das Entfernen einiger Informationen aus diesen Bändern oder deren Herunterskalierung verbessert die Komprimierungseffizienz, ohne die Rekonstruktion zu beeinträchtigen. Wir haben SE-Blöcke für die Kanalaufmerksamkeit verwendet, um die spektrale Redundanz zu verstehen. Diese Blöcke bewerten und gewichten die einzelnen Kanäle auf der Grundlage ihrer Merkmalsausprägungen und Interdependenzen. 3D-convolutions haben sich auch bei der Videokomprimierung bewährt, und mit Bändern als dritter Dimension können sie auf multispektrale Daten angewendet werden. Die untersuchten Architekturverbesserungen mit SE-Blöcken und 3D-convolutions für die räumlich-spektrale Kompression sind daher in der Lage, diese spektrale Redundanz zu nutzen und eine etwas bessere Komprimierung bei RS-Archiven zu erreichen. Die Leistungsanalyse des BigEarthNet-Datensets zeigt, dass die Gesamtqualität der dekomprimierten Bilder im Vergleich zur räumlichen Komprimierung in allen Bitratenbereichen gestiegen ist, ohne dass die Komprimierungsrate durch die vorgeschlagenen Architekturverbesserungen im Vergleich zu einem naiven HiFiC-Netzwerk dramatisch beeinflusst wurde. Die Tatsache, dass die vorgeschlagenen Architekturverbesserungen (insbesondere SE-Blöcke) einfach auf jedes bereits bestehende Komprimierungsnetzwerk angewendet werden können, macht sie noch vorteilhafter. Unser Erfolg zeigt, dass kompliziertere Techniken in Zukunft eingesetzt werden können, um noch bessere Ergebnisse zu erzielen.

# Contents

<b>List of Acronyms</b>	<b>VI</b>
<b>List of Figures</b>	<b>VII</b>
<b>List of Tables</b>	<b>IX</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Work</b>	<b>4</b>
2.1 Rate-Distortion Theory . . . . .	4
2.2 End-to-End Lossy Image Compression . . . . .	4
2.3 Generative Adversarial Networks (GANs) . . . . .	5
2.4 Squeeze-and-Excitation Block (SE) . . . . .	6
2.5 2D and 3D Convolutional Layers . . . . .	6
<b>3 Proposed Compression Architectures</b>	<b>8</b>
3.1 Spatial Compression with HiFiC . . . . .	8
3.2 Spatio-Spectral Compression . . . . .	10
3.2.1 Squeeze-and-Excitation Generative Adversarial Network (SE GAN) . . . . .	10
3.2.2 3D Convolutional Generative Adversarial Network (3D CGAN) . . . . .	11
<b>4 Dataset and Experiments</b>	<b>12</b>
4.1 Dataset . . . . .	12
4.2 Experimental Setup . . . . .	13
4.3 Ablation Studies . . . . .	14
<b>5 Experimental Results and Discussions</b>	<b>17</b>
5.1 Ablation Studies . . . . .	17
5.1.1 Normalization . . . . .	17
5.1.2 Distortion Loss . . . . .	17
5.1.3 Target Rate and Lambdas . . . . .	18
5.1.4 Placement of SE Blocks and 3D Convolutional Layers . . . . .	19
5.2 Spatial Compression Results with HiFiC Network . . . . .	20
5.3 Spatio-Spectral Compression Results . . . . .	22
5.3.1 SE GAN . . . . .	22
5.3.2 3D CGAN . . . . .	25
5.4 Comparison between Spatial and Spatio-Spectral Compression . . . . .	27

<b>6 Conclusion and Discussion</b>	<b>28</b>
6.1 Further Studies . . . . .	28
<b>Bibliography</b>	<b>30</b>
<b>Appendix</b>	<b>33</b>

## List of Acronyms

AE	Autoencoder
BPP	Bits Per Pixel
CGAN	Convolutional Generative Adversarial Network
CNN	Convolutional Neural Network
d	Distortion
FC-MLP	Fully Connected Multi-Layer Perceptron
GAN	Generative Adversarial Network
GDN	Generalized Divisive Normalization
HiFiC	High-Fidelity Generative Image Compression
LPIPS	Learned Perceptual Image Patch Similarity
LSTM	Long Short-Term Memory
MSE	Mean Squared Error
MS-SSIM	Multi-Scale Structural Similarity Index Measure
PSNR	Peak Signal-to-Noise Ratio
r	bit-Rate
RD	Rate-Distortion (trade-off)
RNN	Recurrent Neural Networks
RS	Remote Sensing
SE	Squeeze-and-Excitation (block)
SSIM	Structural Similarity Index Measure
VAE	Variational Autoencoder

# List of Figures

1.1	RS data volume per year of Sentinel 1, 2 and 3 compared to MODIS and Landsat8 (originally published in [21]) . . . . .	1
2.1	A Squeeze-and-Excitation block (originally published in [7]) . . . . .	6
2.2	SE block integration designs (originally published in [7]) . . . . .	6
2.3	(a) 2D convolution, (b) 3D convolution (originally published in [27]) . . . . .	7
3.1	Architecture of the spatial HiFiC . . . . .	8
3.2	Frequency of lambda switching, (a) original HiFiC with MSE on RGB images, (b) spatial architecture with MSE on RS images, (c) spatial architecture with SSIM+MSE of RS images . . . . .	10
3.3	Architecture of the SE GAN . . . . .	10
3.4	SE block's structure . . . . .	11
3.5	Architecture of 3D CGAN . . . . .	11
4.1	Patch sample . . . . .	13
4.2	Effect of normalization: (a) per sample normalization, (b) global normalization . . . . .	15
5.1	Information loss through normalization: (a) per sample normalization, (b) global normalization, (c) image in (b) compressed with autoencoder . . . . .	18
5.2	Impact of loss functions on the final decompressed results. Results obtained with models trained with (a) SSIM, (b) MS-SSIM . . . . .	18
5.3	Results obtained with 3D GAN models with different placements of 3D convolutions (a) 3D Convolution in the initial layer of the Encoder and the final layer of the Generator (b) 3D convolution in all layers. Visually the results look very similar. . . . .	20
5.4	Reconstructed sample 1 with spatial compression architecture: autoencoder and GAN at $r_t = 0.8$ . . . . .	21
5.5	Reconstructed sample 2 with spatial architecture: autoencoder and GAN at $r_t = 0.8$ . . . . .	22
5.6	Reconstructed sample with spatial compression HiFiC for different $r_t$ values. . . . .	23
5.7	Reconstructed sample of SE GAN architecture: autoencoder and GAN, $r_t = 0.8$ . . . . .	24
5.8	Close up comparison of spatial compression with HiFiC and spatio-spectral compression with SE GAN . . . . .	25
5.9	Reconstructed sample of 3D CGAN architecture: autoencoder and GAN at $r_t = 0.8$ . . . . .	26
5.10	Close up comparison of spatial compression with HiFiC and spatio-spectral compression with and 3D CGAN. . . . .	26



5.11 Average PSNR and bpp of all models . . . . .	27
A.1 Comparison on different bands and $r_t$ values of spatial GAN . . . . .	33
A.2 Comparison on different bands and $r_t$ values of SE GAN . . . . .	34
A.3 Comparison on different bands and $r_t$ values of 3D CGANs . . . . .	35
A.4 Spatial autoencoder and GAN results at different bit rates (large samples) . . . . .	37
A.5 Spatial GAN, SE GAN and 3D CGAN on different bit rates . . . . .	40

# List of Tables

4.1 List of Sentinel-2 bands . . . . .	12
4.2 List of target rate $r_t$ and $\lambda^{(a)}$ pairs for creating different models at each of the bit rates. . . . .	14
5.1 Metrics obtained with different placements of SE blocks in SE GAN. . . . .	19
5.2 Metrics obtained with different placement of 3D convolutions in 3D GAN (Note that the models are trained on smaller amount of iterations and results may differ from the main experiments). . . . .	20
5.3 List of average PSNR and bpp values of spatial compression with HiFiC at different target bit rates. . . . .	21
5.4 Bands' weights calculated by SE Block. Max and min values are marked with color . . . . .	23
5.5 List of average PSNR and bpp values of SE GANs . . . . .	24
5.6 List of average PSNR and bpp values of 3D CGANs at different target rates. . .	25

# 1 Introduction

Data compression is the process of encoding information in order to use fewer bits than the original representation, to help reduce storage and transmission costs. It is especially relevant in the remote sensing (RS) domain where large volumes of high-resolution images are encountered. Data volume increases exponentially on a daily basis. The biggest orbital expeditions so far are Sentinels and once all series reached full operational capacity, data is transmitted at a rate of 10 TB per day [21]. Figure 1.1 shows the amount of data transmitted per year between 2013-2018. Subsequently, even more data will be received, and more orbital expeditions will be launched. Considering the large volume of data, a higher compression rate might be needed.

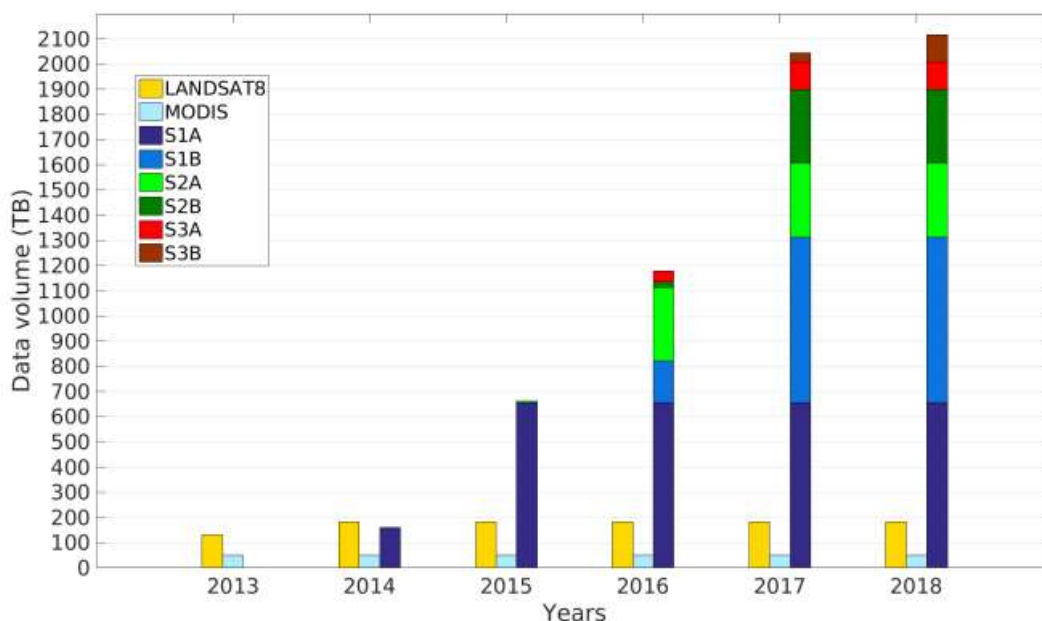


Figure 1.1: RS data volume per year of Sentinel 1, 2 and 3 compared to MODIS and Landsat8 (originally published in [21])

Traditional multispectral image compression codecs consist of several modules, such as prediction, vector quantization, and transform coding. In prediction coding, which is used for lossless compression, pixels can be predicted by using their relation with the spatial and spectral neighbors. The errors of predicted pixels are relatively small and easy to compress, but compression rate is rather low. In vector quantization coding, data is divided into a set of vectors,

## 1 Introduction

which is quantized and encoded. Compression rate of this approach is bigger than in prediction coding, but due to the calculation complexity, the computation time is longer. Transform coding does not have those disadvantages: it can achieve higher compression without long computation times. However, it is not perfect. Transform coding divides the image on small blocks and, after their analysis, builds a frequency domain for more targeted quantization. Such division can introduce blocking effects and visible boundaries, which affect the overall quality of the compressed image [12].

Despite the versatility and popularity of traditional codecs, all of them share the same limitation, which is optimization. Codec modules are interdependent on each other, making it difficult to update each individual block separately. Moreover, the partial change of one module is often insufficient for noticeable improvement. While it is difficult to overcome these limitations using only traditional programming, as machine learning grows in popularity, the number of new image compression methods available also increases.

The first working model of RGB image compression was published in 2016 by George Toderici [26]. It is based on recurrent neural network (RNN) and uses end-to-end optimization to optimize the entire framework concurrently. Performance improvement in a module naturally leads to improvement of the final objective, and joint optimization causes all modules to work more adaptively with each other. Moreover, RNN showed better results when reconstructing images when compared to traditional methods such as JPEG2000 [22] and WEBP codecs. Follow-up studies further increased the gap between traditional and learning based compression. Most prior studies have focused on RGB images, but since the majority of geospatial data is multispectral, RGB compression is not fully applicable to RS archives. However, RGB images can be considered a special type of multispectral images with only three channels. With this in mind, it should be possible to adapt RGB-only compression models to multispectral RS data with sub-optimal compression rates. Most of the works in RS compression directly use the existing RGB compression architectures without considering the spectral redundancies, such as the Reduced-Complexity End-to-End Variational Autoencoder [20] that was made for onboard satellite image compression. This model was adapted for multispectral images from Generalized Divisive Normalizations (GDN) [1] with the assumption that all channels are independent. Only a few works exist that exploit spatial and spectral redundancies for multispectral compression. Model ResConv [12] is one of such example, which can do both spectral and spatial compression and outperforms JPEG2000 and 3D-SPIHT [14] (transform-based codec). Despite the success of ResConv, there are still few end-to-end optimized compression models for RS data compared to RGB-based ones.

The main goal of this thesis is to develop better models for spatio-spectral compression of an RS archive. We explore the existing Generative Adversarial Network (GAN) based compression architectures [17] [30] that have improved compression performance in RGB images compared to Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) (which are the most basic architectures in learning compression) and adapt it for multispectral images by exploiting the spectral and spatial redundancies. To this end, we explore new architectures using: 1) Squeeze-and-Excitation (SE) blocks; and 2) 3D convolutions. SE blocks [7] reduce feature redundancies using channel attention. Similarly, 3D convolutions [15] play a major role in video compression methods, making it a potential candidate for multispectral data where the spectral

axis can be viewed as similar to the time axis in videos. We explored two new architectural changes to the GAN based spatial compression to exploit the spectral redundancies and provide better compression rates of the RS archives.

GANs have improved performance in compression and significant potential for spatial and spectral compression. GANs have a very flexible architecture and can use different network types as its parts (e.g., CNN as generator/discriminator). This gives freedom of choice and the ability to use improvements specific to particular network types. A good example of a spatial compression model is High-Fidelity Generative Image Compression (HiFiC) [17]. HiFiC can work with high-resolution RGB images, and its main focus is to create natural looking reconstructions. Also, HiFiC has open-source code, making it a good candidate for further experiments.

As a baseline study, we first adapted the HiFiC architecture to multispectral images to perform spatial compression only. All bands are compressed independently, and only the RGB channels are weighted by Learned Perceptual Image Patch Similarity (LPIPS) [31], similar to the original HiFiC. The resulting architecture is trained and evaluated for different rate-distortion trade-offs to test as many scenarios as possible.

The second stage is implementation of spatio-spectral compression. We propose architecture changes to the spatial GAN by using: i) SE blocks; and ii) 3D convolutions. Addition of channel attention in the form of a SE block into spatial GAN can increase overall performance. We also test the replacement of 2D convolutions with 3D convolutions. Both architectures are trained under the same conditions as spatial GAN for a more straightforward comparison and to determine which is more successful with the RS data.

This thesis is organized in the following structure: **Chapter 2** gives a more detailed overview of the existing compression methods and the background theory required for understanding the architectural changes introduced in the thesis. **Chapter 3** contains the proposed architectures: spatial and spatio-spectral ones. **Chapter 4** has information about the dataset and experimental setup. All experimental results will be shown and discussed in **chapter 5**. **Chapter 6** concludes the thesis and gives an outlook of further research in the direction of spatio-spectral compression.

## 2 Related Work

Some important concepts related to image compression and a few theories regarding GANs, SE blocks, and 3D convolutions will be discussed in this chapter for better understanding.

### 2.1 Rate-Distortion Theory

The theoretical foundations of lossy compression are rooted in Shannon’s formative work [24] on rate-distortion theory. This theory describes how much compression can be achieved by using compression methods, and how much distortion is incurred when reconstructing the data from its compressed representation. Even though both high compression rate and quality reconstruction are desirable, in reality they are inversely proportional and can not be achieved together. That is why finding the rate-distortion (RD) trade-off is one of the main challenges of lossy compression. It may be calculated as:

$$RD = r + \lambda d \quad (2.1)$$

where  $\lambda$  is varying hyperparameter,  $r$  is rate and  $d$  is distortion loss.

### 2.2 End-to-End Lossy Image Compression

For traditional compression methods, like JPEG2000 [22] or BPG [4], improved performance mainly comes from designing more complex tools for each component in the coding loop. Deeper analysis can be conducted on the input image, and more adaptive operations can be applied, resulting in more compact codes. However, in some cases, although the performance of the single module is improved, the final performance of the codec, i.e., the superimposed performance of different modules, might not increase much, making further improvement difficult. Since traditional codecs cannot be optimized as a whole, researchers have turned to machine learning to achieve better reconstruction performance and quality [8]. Consequently, end-to-end learned image compression was developed.

Two key aspects must be considered when designing an end-to-end learned image compression method, latent representation coefficients and probability distribution. If the latent representation coefficients after the transform network are less correlated, a greater bit-rate can be saved in the entropy coding. Meanwhile, if the probability distribution of the coefficients can be accurately estimated by an entropy model, the bit-stream can be more efficiently utilized and the bit-rate required to encode the latent representations can be better controlled. Thereby achieving a better trade-off between the bit-rate and distortion [8].

There are several major differences between traditional and learned based methods. While traditional methods need manual-tuning, learned methods use metrics like Mean Squared Error (MSE), Structural Similarity Index Measure (SSIM) [29], or Multi-Scale Structural Similarity Index Measure (MS-SSIM) [28] that are automatically tuned. The same applies to all trainable parts and parameters: they can be optimized concurrently. However, learned methods require a significant amount of testing and fine-tuning. Also, traditional codecs need an additional rate control component, while learned models can directly target the rate-distortion constraint. The main disadvantage is the need to train several models for different rate-distortion trade-offs, which may be time-consuming. Most traditional codecs divide the image into blocks and work with each block separately, which often introduces the blocking effect on the recreated image. Neural networks support the processing of the whole image, even with high-resolution, however it dramatically increases model complexity and computational speed [8].

George Toderici first introduced working end-to-end optimized networks [26]. His model reconstructs the image by applying a RNN and a convolutional Long Short-Term Memory (LSTM), better than JPEG2000. Toderici's work motivated other researchers to try different architectures. Right now, most methods can be roughly divided into three groups:

- **RNN.** The first model introduced by Toderici [26], then in 2018 Johnston [11] suggested adding the priming method. In general RNNs perform well, however, their major disadvantage is slow computation time as a result of its recurrent nature. While RNNs are still used in image compression, they are mostly used in combination with other architectures, for example the model of Islam et al. [10].
- **CNN.** CNNs are fast and flexible. They can successfully capture spatial and temporal dependencies within an image using filters and are built upon the variational autoencoder (VAE) concept. To increase overall performance, CNN was incorporated with generalized divisive normalization (GDN) in 2017 (by Ballé [1]). GDN is spatially adaptive and non-linear. Next, Hyperprior was added in 2018 [3], to capture spatial dependencies. Then 2D convolutions (by Minnen [18]) and 3D convolutions (by Mentzer [16]) were introduced to improve context understanding. Finally, the Gaussian mixture model (GMM) by Cheng in 2020 [5], was employed for more accurate estimation of likelihood.
- **GAN.** Mentzer's [17] recent work is competitive with CNN-based approaches, since it can reconstruct high-resolution images with low bit-rates by inducing natural looking textures. GANs generate data that looks more natural to the human eye, but they are much harder to train than RNNs or CNNs.

## 2.3 Generative Adversarial Networks (GANs)

GANs were designed by Ian Goodfellow [6] in 2014. It is based on minimax game theory [19] and consists of two models ("players"), that are simultaneously trained. One is a generative model (generator) that captures the data distribution, and the other is a discriminative model (discriminator) that estimates the probability that a sample came from the training data rather than generator. The training procedure for the generator is to maximize the probability of the

## 2 Related Work

discriminator making a mistake. At the same time, the discriminator learns how to distinguish fake data sent by the generator and real samples taken from dataset and is trying to minimize the error. Since the generator and the discriminator are separate neural networks, they can have different architectures, for example convolutional or contain residual blocks. Other blocks or frameworks can also be added, which makes GAN very flexible.

## 2.4 Squeeze-and-Excitation Block (SE)

The SE block [7], depicted in Figure 2.1 investigates the relationship between channels by explicitly modelling the inter-dependencies between them. The SE block evaluates all channels and puts their weights in a  $(1 \times 1 \times C)$  array. These weights are applied to the feature maps to generate the output of the SE block, which can be fed directly into subsequent layers of the network.

The SE block is an add-on module and can be added to any baseline architecture to improve performance, with negligible computational overhead. It should be placed close to the residual block. Figure 2.2 shows different options for integration.

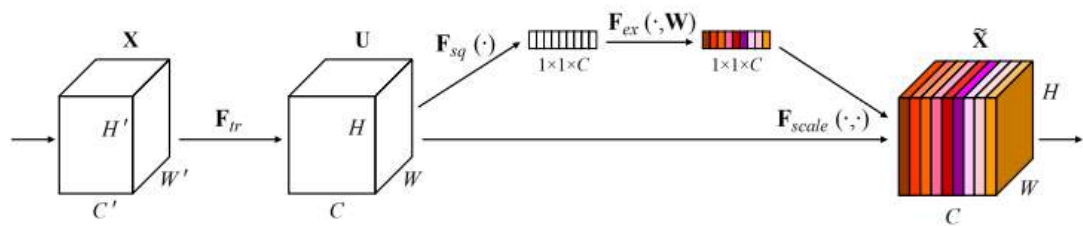


Figure 2.1: A Squeeze-and-Excitation block (originally published in [7])

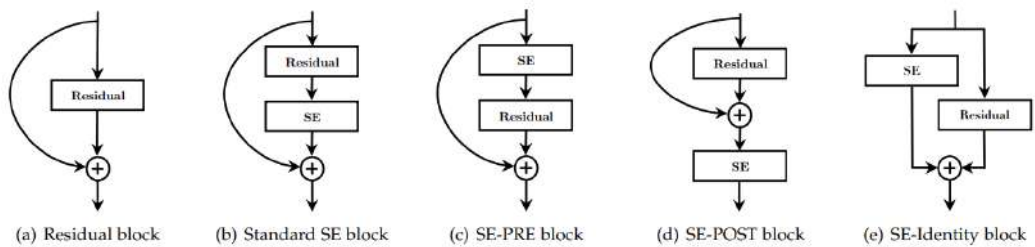


Figure 2.2: SE block integration designs (originally published in [7])

## 2.5 2D and 3D Convolutional Layers

The convolutional layer is a filter applied to the input to assign importance through weights and biases. Repeated application of the same filter to an input results in a map of activations called



a feature map.

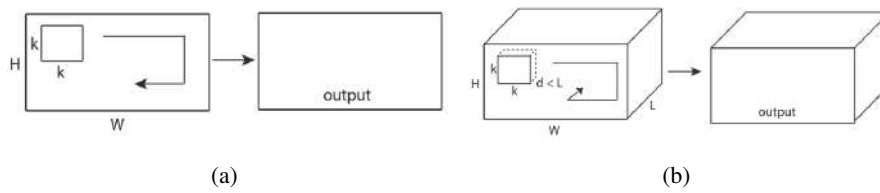


Figure 2.3: (a) 2D convolution, (b) 3D convolution (originally published in [27])

Since the filter moves across the data (see Figure 2.3), there are different types of convolutions depending on the dimensionality of the input. 2D convolutions are used for two or more dimensional data, like images. 3D convolutions require data with at least three dimensions, like videos or 3D images. While 3D convolutions may also be applied to images, with the RGB channel as the third dimension, it is unnecessary since RGB channels are independent and equal. Conversely, multispectral RS images may have dependencies that 2D convolutions are unable to detect.

## 3 Proposed Compression Architectures

This chapter covers the proposed method for spatio-spectral compression of RS archives. We built off of the existing HiFiC network [17] proposed for the RGB data compression. Then we extended it to multispectral compression by considering the channels as independent and applying HiFiC directly on the RS archive. Next we introduced the architecture improvements using a SE block and 3D convolutions for spatio-spectral compression.

### 3.1 Spatial Compression with HiFiC

Given a set of RS images  $x$  the aim of spatial compression is to produce its representation  $y$  which has the least possible spatial correlations and can be entropy coded and stored with minimal bit-rates. The representation learned should be such that it encodes all the necessary information for further decoding ( $x'$ ) and visualizing the images with the least distortion possible. Spatial compression can be achieved by using existing RGB compression models where the channels are treated independently and only the spatial correlations are exploited. We explored the use of GAN based compression model HiFiC [17] for spatial compression of RS archives.

Spatial compression using GAN based architecture has four parts: the encoder  $E$ , probability model  $P$ , the generator  $G$ , and the discriminator  $D$  (see Figure 3.5). Both the encoder  $E$  and the generator  $G$  use ChannelNorm as the normalization layer to avoid darkening artifacts. With an encoder  $E$  and quantizer  $Q$ , the image  $x$  is encoded to a compressed representation  $y$ .  $P$  denotes a scale hyperprior [3] - a block that consists of an arithmetic encoder and an arithmetic decoder. Hyperprior helps to extract the side information ( $z$ ) in order to model the distribution of  $y$  and simulate quantization with uniform noise in the hyper-encoder and when estimating  $p(y|z)$ . Next, the representation  $y$  is decoded with generator  $G$  as  $x'$ . Finally, a single-scale discriminator  $D$ , also with access to the side information ( $z$ ) and the conditional information ( $y$ ), decides if the reconstructed image is acceptable.

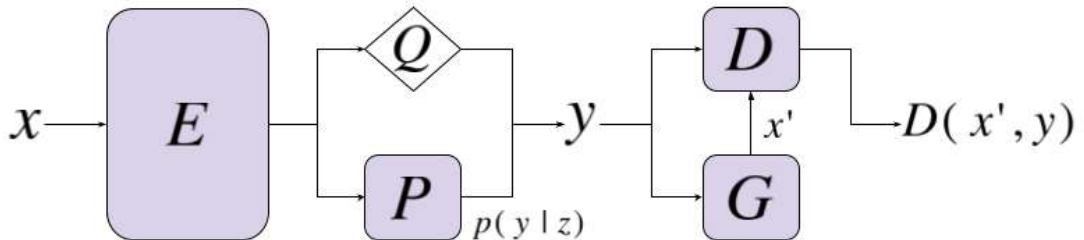


Figure 3.1: Architecture of the spatial HiFiC

### 3.1 Spatial Compression with HiFiC

Considering  $E$ ,  $G$ ,  $D$ , and  $P$  as CNNs (readers are referred to HiFiC [17] for detailed network description of each of these parts) they can be trained jointly by minimizing the rate-distortion trade-off. The optimization of  $E, G$  and  $P$  are done using the overall loss given as:

$$L_{EGP} = \mathbb{E}_{x \sim p_X} [\lambda r(y) + d(x, x') - \beta \log(D(x', y))] \quad (3.1)$$

where  $r(y)$  is rate,  $d(x, x')$  is distortion loss,  $\lambda$  is the hyperparameter controlling the trade-off, and  $\log(D(x', y))$  is the discriminator loss with  $\beta$  controlling the discriminator loss effect. Here the distortion loss  $d(x, x')$  is at odds with the rate term  $r(y)$ . The distortion loss is modelled as

$$d = k_M * MSE + k_P * LPIPS \quad (3.2)$$

where  $k_M$  and  $k_P$  are hyperparameters and MSE and Learned Perceptual Image Patch Similarity (LPIPS) are the loss terms.

The final (average) bit-rate of the model is thus controlled by varying only  $\lambda$  in (3.1). For a fixed  $\lambda$ , different  $k_M$ ,  $k_P$  and  $\beta$  would thus result in models with different bit-rates, making comparison difficult. To alleviate this, target bit-rate  $r_t$  is used.  $\lambda$  is also exchanged by two hyperparameters  $\lambda^{(a)}$ ,  $\lambda^{(b)}$ , where  $\lambda = \lambda^{(a)}$  if  $r(y) > r_t$ , otherwise  $\lambda = \lambda^{(b)}$ . Setting  $\lambda^{(a)} \gg \lambda^{(b)}$  allows model learning with an average bit-rate close to  $r_t$ . By making  $\lambda^{(b)}$  set and changing only  $r_t$  and  $\lambda^{(a)}$ , it is possible to achieve different bit-rate ranges without changing the whole architecture.

LPIPS in (3.2) is the metric that mimics the human visual system and measures the distance in the feature space. It uses trained architectures like AlexNet [13], VGG Net [23], or Squeezenet [9] to calculate the difference between the generated and the reference image features to understand the perceptual differences between the data points.

For the spatial compression model with HiFiC we use the variant based on AlexNet [13] as in [17] for LPIPS calculation. The weights of AlexNet are predefined and must be downloaded before training. Since they are made for RGB images specifically and are placed in binary files, there is no way to adapt them for multispectral data. Therefore, we apply LPIPS on RGB channels of the multispectral image, avoiding other channels.

We observed that MSE is also not suitable for multispectral data because it reduces the switching between  $\lambda^{(a)}$  and  $\lambda^{(b)}$ , which is required to keep the current rate close to the desired target rate  $r_t$ . For RGB images, HiFiC maintains this change with high frequency but for multispectral data this switching is rarer. In order to correct this, we used SSIM loss along with MSE in the distortion loss calculation. Combining SSIM with MSE helped achieve more frequent lambdas switches (see Figure 3.2) with no color change.

So, we updated the distortion loss for spatial GAN architecture as:

$$d = k_M * (\theta_1 MSE + \theta_2 (1 - SSIM)) + k_P * LPIPS \quad (3.3)$$

where  $\theta_1$  and  $\theta_2$  control the effect of MSE and SSIM in the total loss calculation and  $k_P * LPIPS$  is only applied to the RGB channels of the multispectral data.

### 3 Proposed Compression Architectures

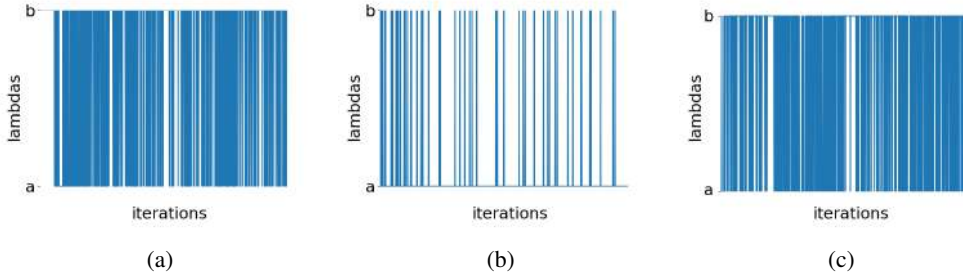


Figure 3.2: Frequency of lambda switching, (a) original HiFiC with MSE on RGB images, (b) spatial architecture with MSE on RS images, (c) spatial architecture with SSIM+MSE of RS images

## 3.2 Spatio-Spectral Compression

Multispectral data exhibits spectral redundancy which can also be used along with spatial redundancies to achieve better compression. In order to understand the spectral information relevant for reconstructing the RS images, we used a channel attention module (SE block) in the spatial compression module. We also tried replacing the 2D convolutions in the spatial compression model with 3D convolutions to get better spectral compression. Both suggested improvements for spatio-spectral architecture were separately applied to the already adapted spectral architecture from section . The details of the proposed architectures are explained in detail in the below sections.

### 3.2.1 Squeeze-and-Excitation Generative Adversarial Network (SE GAN)

For the first spatio-spectral architecture, we added Squeeze-and-Excitation blocks on the initial layer of the encoder and after each ChannelNorm block in both the encoder and the generator (Figure 3.3), followed after a group of residual blocks (SE-POST architecture, see Figure 2.2, (d)).

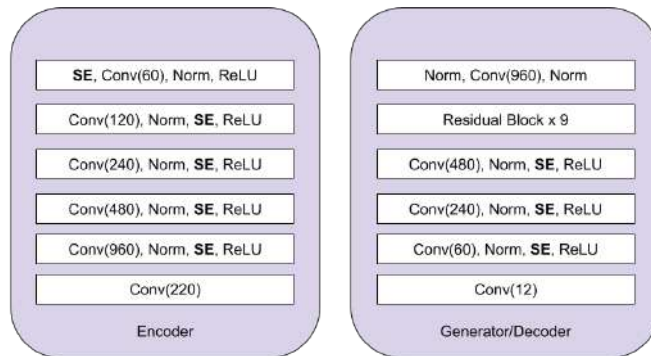


Figure 3.3: Architecture of the SE GAN

The SE block (Figure 3.4) itself consists of several layers: Global Pooling, Fully Connected Multi-Layer Perceptron (FC-MLP) with ReLU activation, where neuron number will be decreased by reduction ratio  $r_r$ , and FC-MLP with Sigmoid activation. Both activations have  $l_1$  regularization, for more sparse weights distribution. After the input goes through all layers, it will be scaled: multiplied with the initial input.

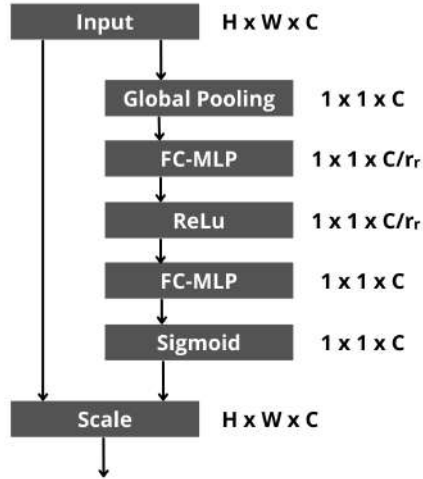


Figure 3.4: SE block's structure

### 3.2.2 3D Convolutional Generative Adversarial Network (3D CGAN)

For the second spatio-spectral compression, we replaced some of the 2D convolutional layers with 3D ones. Since the last 2D layer of the encoder is needed for entropy coding, we only replaced the initial layers of the encoder  $E$  and the final layer of the generator  $G$  (Figure 3.5).

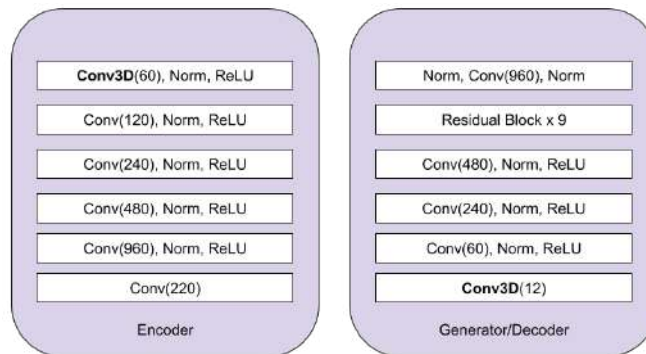


Figure 3.5: Architecture of 3D CGAN

## 4 Dataset and Experiments

### 4.1 Dataset

The suggested architecture was trained and evaluated on multispectral Sentinel-2 images, which consist of 13 bands with different light spectrum and resolutions (Table 4.1).

Band	Light Spectrum, nm	Resolution, m
Band 1: Coastal aerosol	443	60
Band 2: Blue	490	10
Band 3: Green	560	10
Band 4: Red	665	10
Band 5: Vegetation red edge	705	20
Band 6: Vegetation red edge	740	20
Band 7: Vegetation red edge	783	20
Band 8: NIR	842	10
Band 8A: Narrow NIR	865	20
Band 9: Water vapor	940	60
Band 10: SWIR - Cirrus	1375	60
Band 11: SWIR	1610	20
Band 12: SWIR	2190	20

Table 4.1: List of Sentinel-2 bands

We used BigEarthNet-S2 [25] as a source of Sentinel-2 images. This is a large RS archive of 590,326 image samples from 10 European countries (Austria, Belgium, Finland, Ireland, Kosovo, Lithuania, Luxembourg, Portugal, Serbia, Switzerland). After removing 71,042 patches which were covered by seasonal snow and clouds, 519,284 patches remained. In order to reduce training time, the following experiments were conducted on 14,832 patches selected from BigEarthNet Serbia Summer area. These selected patches were then split between the training set (7,761), the validation set (3,508), and the test set (3,563).

BigEarthNet-S2 is often used for image recognition and classification. It has only 12 bands: B10, which contains information about clouds, was removed since it was not useful for model testing. This removal did not affect image compression, and only reduced the number of channels from 13 to 12. Figure 4.1 shows a patch sample.

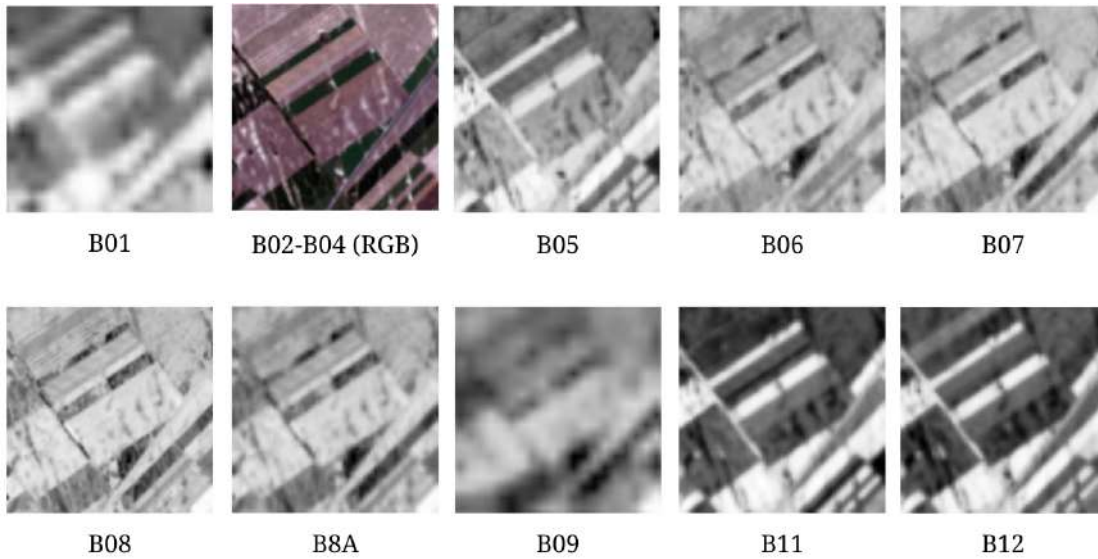


Figure 4.1: Patch sample

For data preprocessing, cubic interpolation was applied to all bands, in order to increase their size to 128x128. Then each sample was converted into int32 format and normalized between [0, 255]. This step guaranteed proper functioning of the HiFiC architecture.

## 4.2 Experimental Setup

All the experimental results were obtained by training on a NVIDIA Tesla V100 GPU with 32 GBs of memory. Code for experiments was implemented with TensorFlow v1.15.2 and TensorFlow Compression v1.3 [2].

According to the definition of target rate  $r_t$  in [17], the maximum bit-rate that can be achieved is approximately 0.45 bpp, but we decided to extend this range till  $r_t = 1$  to see how HiFiC works with higher bit-rates. As mentioned in section 3.1, different bit-rates can be achieved by changing two hyperparameters: target rate  $r_t$  and  $\lambda^{(a)}$ . The associated  $\lambda^{(a)}$  values are set with the similar step between models as in the original HiFiC. In total we configured five models with different settings of  $r_t$  and  $\lambda^{(a)}$  as listed in Table 4.2.

Since the dataset was changed from high-resolution RGB to multispectral images, the data load had to be completely rewritten. We prepared tfrecord files for the training, test, and validation sets, and loaded them as the TensorFlow dataset. Due to the small sample size, the cropping function was removed completely. We also changed all the data shapes from batch size, height, width, and 3 to batch size, height, width, and channels. The original HiFiC was trained on 1,000,000 iterations with setting smaller  $r_t$  and  $\lambda$  after the 50,000th iteration and 500,000th iteration for learning rate  $lr$ . Training first on higher values and then switching to smaller values helps alleviate rate loss dominating training. After testing, it was established that training on 200,000 iterations was sufficient for a smaller dataset with lower resolution. All hyperparam-

## 4 Dataset and Experiments

$r_t$	$\lambda^{(a)}$
0.2	$2^1$
0.4	$2^0$
0.6	$2^{-1}$
0.8	$2^{-2}$
1	$2^{-3}$

Table 4.2: List of target rate  $r_t$  and  $\lambda^{(a)}$  pairs for creating different models at each of the bit rates.

ters (including  $lr$ ) should be decreased in the middle of the training, in our case on the 100,000th iteration. For the distortion loss Eq. (3.3) we used  $\theta_1 = 0.2$  and  $\theta_2 = 1000$ . These values were determined through multiple tests. Other hyperparameters (e.g.  $k_M, k_P, \beta$ ) were left unchanged with a batch size of 8.

Reconstruction quality of the end-to-end compression models are measured using PSNR given as:

$$PSNR(x, x') = 20 * \log_{10} * \frac{255}{\sqrt{MSE(x, x')}} \quad (4.1)$$

where  $x$  and  $x'$  are the original and decompressed images respectively. Bigger PSNR means the reconstruction has more similarities with the original image. PSNR is dependent on the maximum value of the dataset (255), which means data range and normalization type can affect the metric.

And the compression quality is measured in terms of bit-rate in bpp (bits per pixel), with smaller bpp indicating a bigger compression ratio.

The setup and metrics described above apply to all proposed architectures. We started the experiments with the spatial HiFiC model which is adapted for multispectral data. We trained and evaluated the spatial HiFiC for all bit-rate ranges. Then we moved to spatio-spectral models. We evaluated the SE GAN first. The hyperparameter of the SE blocks are set as follows for the training: sparsity regularization with  $l_1$  was weighted by a value 0.1 and the reduction ratio was set as  $r_r = 2$ . Next we experimented with the 3D CGAN architecture. It does not have specific hyperparameters, but the input shapes must be adjusted from batch, height, width, and channels to batch, 1, height, width, and channels before each 3D convolutional layer and reshaped to the original size afterwards. Finally, we compared the results obtained from each of these architectures for a specified bit-rate and evaluated the performance for multispectral RS compression.

### 4.3 Ablation Studies

Since there are several possible solutions for some parts of the framework, we conducted ablation studies to better understand the performance changes. The studies apply to 1) data normalization; 2) target rate and lambda hyperparameters; 3) distortion loss function; and 4) the impact



of the SE blocks' and 3D convolutions' placement. The choice of the final architecture will be discussed in Chapter 5.1 with comparison results to justify the choice.

### Normalization

There are several possible ways to normalize the dataset. We selected two of them for testing: 1) per sample normalization where normalization is applied to each sample between its own max and min values; and 2) global normalization where normalization is performed through dividing the whole dataset by its maximum pixel value (20566). The first method is used in the original HiFiC with RGB images, and the second is a common way to normalize the BigEarthNet archive. There is a noticeable change in color (see Figure 4.2): the first normalization tends to produce cooler tones like red, blue and white, which also makes the images brighter. The second normalization has warmer colors dominating (yellow and green) but it makes the overall appearance darker. So far, it is unclear if choice of normalization will affect the training process, or if it is relevant for visual representation only. It may influence the loss function, since some of them tend to work better on brighter/darker images.

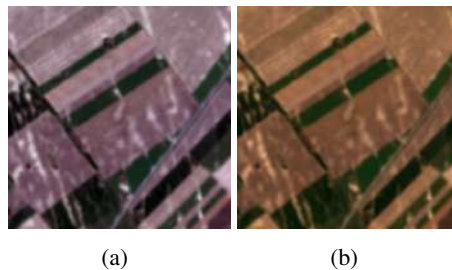


Figure 4.2: Effect of normalization: (a) per sample normalization, (b) global normalization

### Distortion Loss

The distortion loss function of the original HiFiC was not suitable for multispectral data (see explanation in Chapter 3.1). We had to replace it with an alternative (Eq. 3.3) using a combination of MSE, SSIM and LPIPS with  $\theta_1 = 0.2$  and  $\theta_2 = 1000$ . We studied the impact of using other loss functions (like MS-SSIM) on the final output. We also analysed the impact of changing the weightage of MSE and SSIM loss in Eq. 3.3 by changing  $\theta_1$  and  $\theta_2$  and visualizing the results of the so trained networks.

### Target Rate and Lambdas

The pairs of  $r_t$  and  $\lambda^{(a)}$  listed in Table 4.2 were set close to the original HiFiC with the same step between the values. The accuracy of this division must still be tested.

Going with bit-rate further than 1 is difficult because the value of  $\lambda^{(a)}$  is connected with  $\lambda^{(b)}$ , which is fixed to  $2^{-4}$ . The two lambdas must have a large gap between each other to let the

#### 4 Dataset and Experiments

model learn with an average bit-rate close to  $r_t$ . For  $r_t = 1$  this gap is already rather small, and we expect this model to have low precision. If this actually happens, we could try changing  $\lambda^{(b)}$ .

##### **Placement of SE Blocks and Convolutional Layers**

The architecture designed and discussed in Chapters [3.2.1](#) and [3.2.2](#) were made after several experimentation. The impact of placing these blocks at different locations in the architecture will be studied in detail.

## 5 Experimental Results and Discussions

Following the settings mentioned in the previous chapter, five models were trained for each architecture type (spatial GAN, SE GAN and 3D CGAN) for the different bit-rate ranges, so in general there are 15 models. We first trained the autoencoder and then initialized the GAN models from the autoencoder’s checkpoints. Both models, autoencoder and GAN, were evaluated.

### 5.1 Ablation Studies

Before moving to the main experiments, we provide the results of ablation studies.

#### 5.1.1 Normalization

For the test, we prepared two identical datasets. First, we interpolated the bands to the same size, 128x128 pixels. Then applied per sample and global normalizations as discussed in Section 4.3. Next, we brought values to the uint8 format and range of [0,255]. However, rounding the values to int causes loss of information. For per sample normalization this loss was not drastic and not even visually noticeable. The BigEarthNet values are originally saved in int64 format, and division by difference of local extremes did not cause many characters after the decimal to be discarded after rounding. However, for global normalization, we divide the full dataset by the biggest pixel value possible, which resulted very small float values. Rounding of those values caused significant loss of information on some samples, which was even more apparent after compression. Figure 5.1 shows one of those cases: Fig 5.1(b) has visible blocking effects, especially on the green part, which are not present on Fig 5.1(a), that was created using the per sample normalization. Fig 5.1(c) was reconstructed by the autoencoder and the blocking effect is even more noticeable.

Normalizing between [0,1] instead of [0,255] could solve the problem. We tried to adapt HiFiC to the new data format (float32 instead of uint8), but it was unsuccessful. The whole architecture was built for RGB image format, and most of the hyperparameters had to be recalculated. This would be very time-consuming and would make comparison difficult. Consequently, we decided to use per sample normalization, as in the original HiFiC.

#### 5.1.2 Distortion Loss

After choosing the normalization type, we completed the distortion loss study. We tried to establish which metric, SSIM or MS-SSIM, performed better. For that, we trained two models on  $r_t = 0.2$  under the same conditions with 60,000 iterations only. It is enough to see if there is any difference, but does not require long computation time. Figure 5.2 shows the reconstructed images. It is easy to see, that SSIM delivers sharper image with slightly better coloration (both

## 5 Experimental Results and Discussions

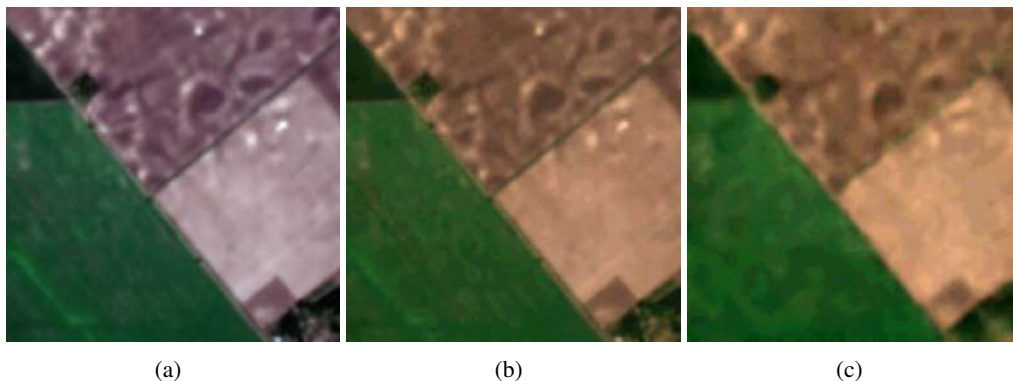


Figure 5.1: Information loss through normalization: (a) per sample normalization, (b) global normalization, (c) image in (b) compressed with autoencoder

images are quite far from the original in terms of color and details because of the small number of iterations).

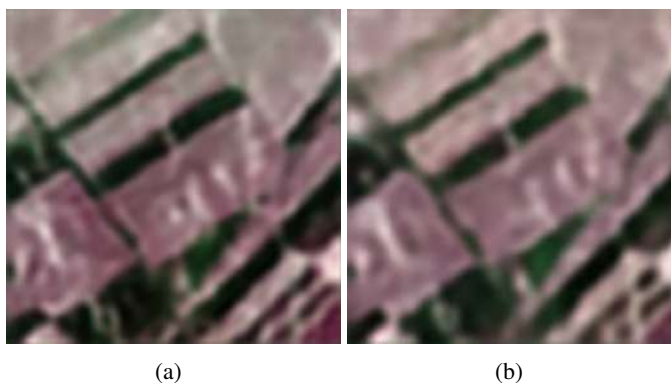


Figure 5.2: Impact of loss functions on the final decompressed results. Results obtained with models trained with (a) SSIM, (b) MS-SSIM

### 5.1.3 Target Rate and Lambdas

$\lambda^{(a)}$  values for the desired target rate were picked based on the original HiFiC. Since we increased the range of rates, we had to pick lambdas for the new rate values. For each target rate  $r_t$  we trained three models: with  $\lambda^{(a)}$  from Table 4.2, smaller and bigger value than one from Table 4.2. The resulting models were evaluated on PSNR and bpp. It was established that the values assigned in Table 4.2 deliver the best result. We also noticed that each lambda tends to bring the final bpp to its "comfortable" rate, which can differ from  $r_t$ . However, this does not apply to higher rates  $r_t = \{0.8, 1\}$ . In that case, models tends to go to higher bpp regardless of the chosen  $\lambda^{(a)}$ . This behavior can be explained with the small gap between  $\lambda^{(a)}$  and  $\lambda^{(b)}$ . Changing  $\lambda^{(b)}$

so that it differentiate with  $\lambda^{(a)}$  enough caused strong discoloration for all models, so we decided to leave  $\lambda^{(b)}$  to be its default value. It is possible to adapt HiFiC for higher bit-rates, but then tuning based on  $\lambda^{(a)}$  and  $\lambda^{(b)}$  balance must be revised or even removed.

#### 5.1.4 Placement of SE Blocks and 3D Convolutional Layers

It is important to place the SE block in the initial part of the encoder, before the convolutional layer, so it can calculate the weights of the spectral channels of the original input. Placing the SE blocks in subsequent layers (downsampling) may differ. We tried placing it after convolution, ChannelNorm and ReLU. The best performance (based on PSNR and visual quality) was the model with the SE block after ChannelNorm.

We also tested the need of SE blocks in the generator. They were placed accordingly after ChannelNorms, but avoiding the initial part before the group of residual blocks and the final convolutional layer. This model showed a larger average PSNR, but computational time also increased. We also tested putting SE blocks after the convolutional layer in both the encoder and the generator, because that placement got the second-best result in the encoder-only placement, but it performed slightly worse. The full list of PSNR values and training times are reported in Table 5.1.

Placement	Avg PSNR	Avg iterations per second (higher is better)
(a) Encoder: initial layer	28.18	7.25
(b) Encoder: initial + after convolution	29.20	7.14
(c) Encoder: initial + after ChannelNorm	29.26	7.10
(d) Encoder: initial + after ReLU	29.12	7.15
(e) Encoder + generator: initial + after convolution	29.27	6.88
<b>(f) Encoder + generator: initial + after ChannelNorm</b>	29.30	6.80

Table 5.1: Metrics obtained with different placements of SE blocks in SE GAN.

For 3D CGAN we started by replacing only the initial layer of the encoder and the final layer of the generator with 3D convolution layers. Another option could be to replace all 2D convolutions in the spatial GAN with 3D convolutions (except for the last layer of the encoder and the first layer of the generator, since they are required for entropy coding). However, we found that replacing only the initial layer of the encoder and the final layer of the generator works just as well as replacing all the layers in terms of recreation quality (see Fig 5.3), but required less training time. Full list of PSNR values and training time are given in Table 5.2.

## 5 Experimental Results and Discussions

Placement of 3D Conv	Avg PSNR	Avg iterations per second
<b>(a) Encoder +generator: initial/final</b>	29.24	5.25
(b) Encoder + generator: initial/final + middle	29.27	3.8

Table 5.2: Metrics obtained with different placement of 3D convolutions in 3D GAN (Note that the models are trained on smaller amount of iterations and results may differ from the main experiments).

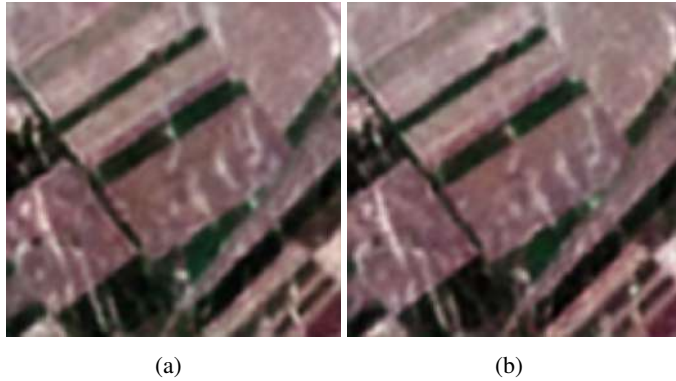


Figure 5.3: Results obtained with 3D GAN models with different placements of 3D convolutions (a) 3D Convolution in the initial layer of the Encoder and the final layer of the Generator (b) 3D convolution in all layers. Visually the results look very similar.

### 5.2 Spatial Compression Results with HiFiC Network

After all ablation studies are done and the hyperparameters and loss functions are finalized, we move to the experiments. For each model we trained the autoencoder and then GAN on top of it. Both of them are evaluated separately. In terms of recreation quality, GAN depends on autoencoder, so if autoencoder has low PSNR and bpp, GAN has the same correlation. GAN is supposed to make autoencoder’s recreation more sharp and improve its texture. In order to achieve that, it has to sacrifice both the bit-rate (new information will be added on top) and PSNR (because texture is not fully identical to the original, but should look similar and be more natural to the human eye). Table 5.3 presents the averaged PSNR and bit-rate on the validation set for spatial GANs. As you can see, pure autoencoder has better PSNR values than GAN and the gap between them increases with increase of bpp.

A closer look at the reconstructed images shows us some strengths and weaknesses of GANs. Figure 5.4 shows a reconstructed sample for  $r_t = 0.8$  with PSNR above the average reported in Table 5.3. There was noticeable improvement with the GAN on the forest’s texture compared to a smooth output with the autoencoder. Most of the samples with higher PSNR values depict either forests or mountains. So, GAN performs best on images with large textured surfaces.

## 5.2 Spatial Compression Results with HiFiC Network

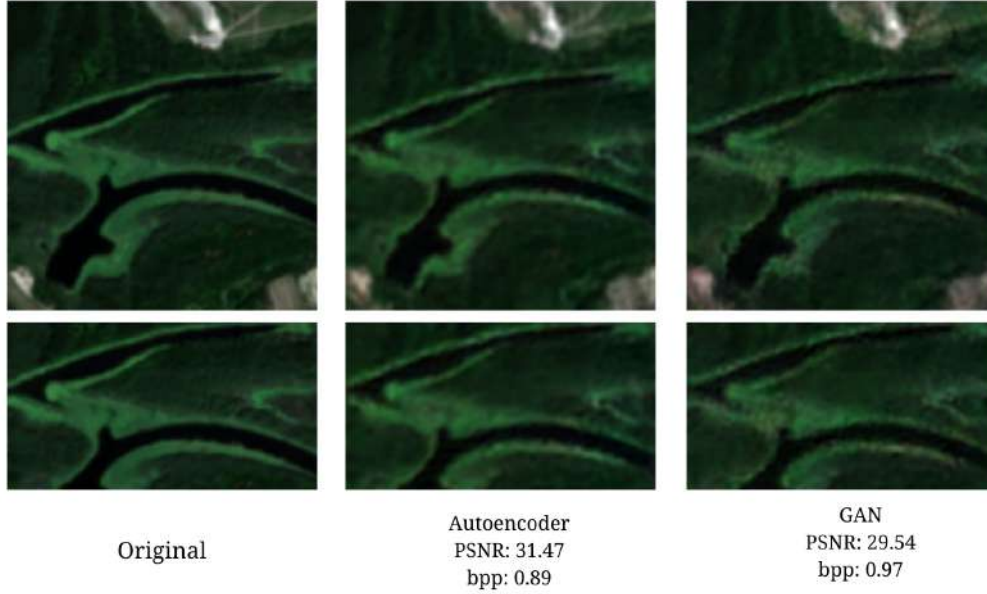


Figure 5.4: Reconstructed sample 1 with spatial compression architecture: autoencoder and GAN at  $r_t = 0.8$

$r_t$	AE		GAN	
	PSNR	bpp	PSNR	bpp
0.2	24.23	0.24	23.78	0.23
0.4	26.49	0.43	25.52	0.44
0.6	27.77	0.64	26.62	0.64
0.8	29.02	0.88	27.71	0.95
1	29.94	1.15	28.56	1.32

Table 5.3: List of average PSNR and bpp values of spatial compression with HiFiC at different target bit rates.

Another interesting case is shown in Figure 5.5 with a sample PSNR lower than the average value. GAN was unable to recreate small objects, especially if there are close to each other, like city layouts. However, we can not determine for sure if the problem is caused due to the autoencoder or the GAN. Since GAN takes autoencoder’s recreation into consideration, poor quality may be just inherited. The original HiFiC also has the same issue, and is likely the reason HiFiC is recommended for high-resolution images.

In Figure 5.6 you can see how bit-rate affects the recreation quality of the compressed images. Starting with bpp greater than 0.6, the image looks more sharp (especially object’s borders and lines), and the color scheme is closer to the original. Models with  $r_t = \{0.6, 0.8\}$  maintain the

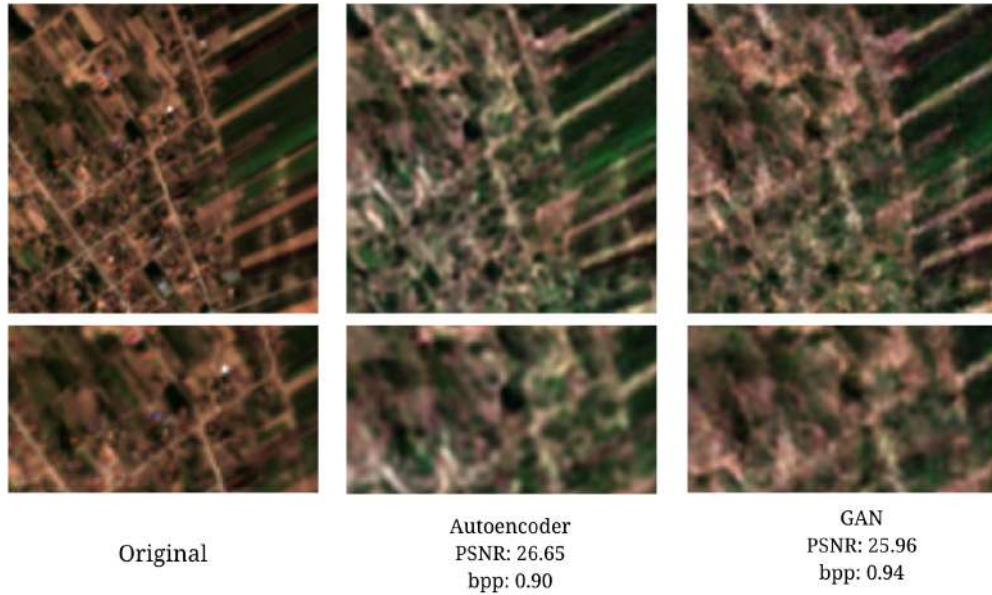


Figure 5.5: Reconstructed sample 2 with spatial architecture: autoencoder and GAN at  $r_t = 0.8$

best balance between quality and compression. Smaller bit-rates have strong discoloration and "phantom" shapes, however, objects are still recognizable and overall quality is decent for that amount of compression.

### 5.3 Spatio-Spectral Compression Results

After evaluation of the spatial compression models we move to spatio-spectral architectures. These architectures are designed to exploit the spectral redundancy and provide better compression rates without impacting the reconstruction quality of images.

#### 5.3.1 SE GAN

The SE GAN architecture is based on channel attention. Each band is weighted depending on its interdependencies with the others. Bands with smaller weights will be compressed more and vice-versa. This should lead to more deliberated image compression, but it is only possible if all weights are sparse. To achieve that, we used  $l_1$  regularization with a higher value as a sparsity constraint.

The band's weights computed by the SE blocks are presented in Table 5.4. They have the range between 0 and 1, and tend to the middle value (0.5). Band 3 (Green) has the highest value, possibly because green color is dominating in the whole dataset. The same applies to band 4: red color is less present, that is why the channel has the smallest impact on reconstruction.



### 5.3 Spatio-Spectral Compression Results

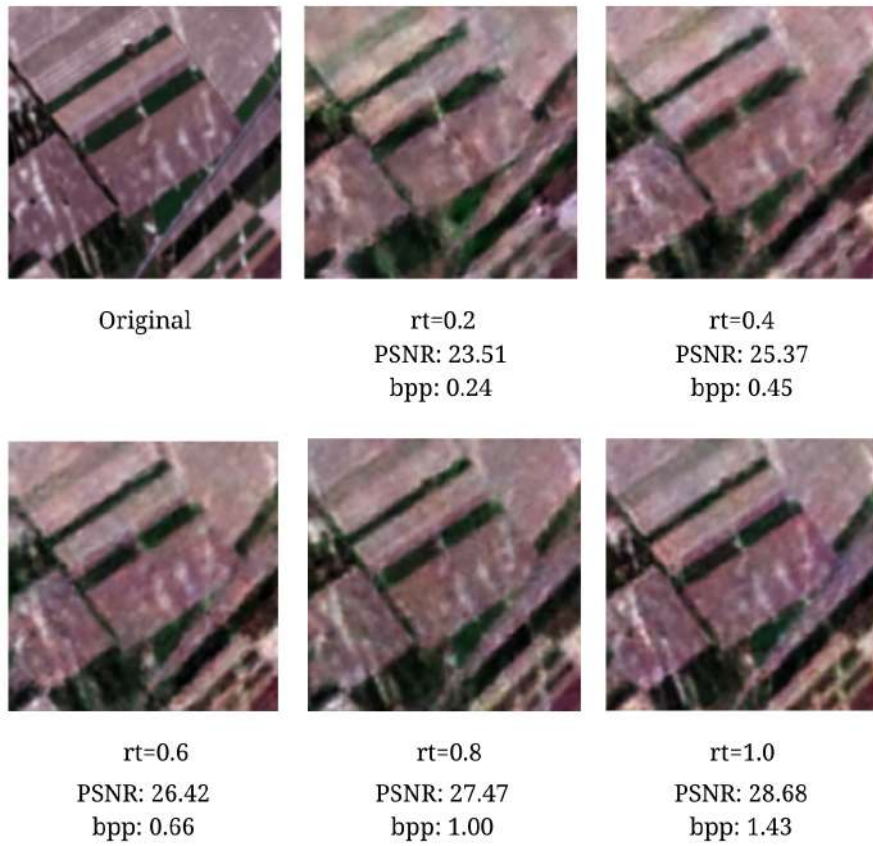


Figure 5.6: Reconstructed sample with spatial compression HiFiC for different  $r_t$  values.

Band	Weight
Band 1: Coastal aerosol	0.46864307
Band 2: Blue	0.48156163
Band 3: Green	0.60466766
Band 4: Red	0.39222118
Band 5: Vegetation red edge	0.39984882
Band 6: Vegetation red edge	0.5554283
Band 7: Vegetation red edge	0.55418533
Band 8: NIR	0.41679692
Band 8A: Narrow NIR	0.51749414
Band 9: Water vapor	0.4960584
Band 11: SWIR	0.4355944
Band 12: SWIR	0.47052222

Table 5.4: Bands' weights calculated by SE Block. Max and min values are marked with color

## 5 Experimental Results and Discussions

$r_t$	AE		GAN	
	PSNR	bpp	PSNR	bpp
0.2	24.75	0.24	23.77	0.24
0.4	26.64	0.44	25.69	0.44
0.6	28.39	0.65	27.03	0.66
0.8	29.3	0.89	28.09	0.95
1	30.07	1.15	28.75	1.29

Table 5.5: List of average PSNR and bpp values of SE GANs

The results from training the SE GAN at different target rates are presented in Table 5.5. PSNR values indicate slight performance increase for both autoencoder and GAN compared to spatial architecture. Bpp in the same time did not change (except for GAN with  $r_t = 1$ ), which means better recreation quality for the same compression rate.

Figure 5.7 shows the reconstructed sample of the autoencoder and the GAN of the SE GAN architectures. Since PSNR values are quite similar to the spatial compression, it may be hard to see visual improvements with the naked eye, because they hide in details. Figure 5.8 shows the areas that were the most affected. The differences were divided into three groups. The first difference is that small objects were more defined (area No 1 on Fig. 5.7). Also, the objects' borders had a sharper contour and texture from other surfaces and tended to be less overlapping (areas No 2 and 3). Finally, the lines were more precise and had higher contrast, making them seem less blurry (area No 4).

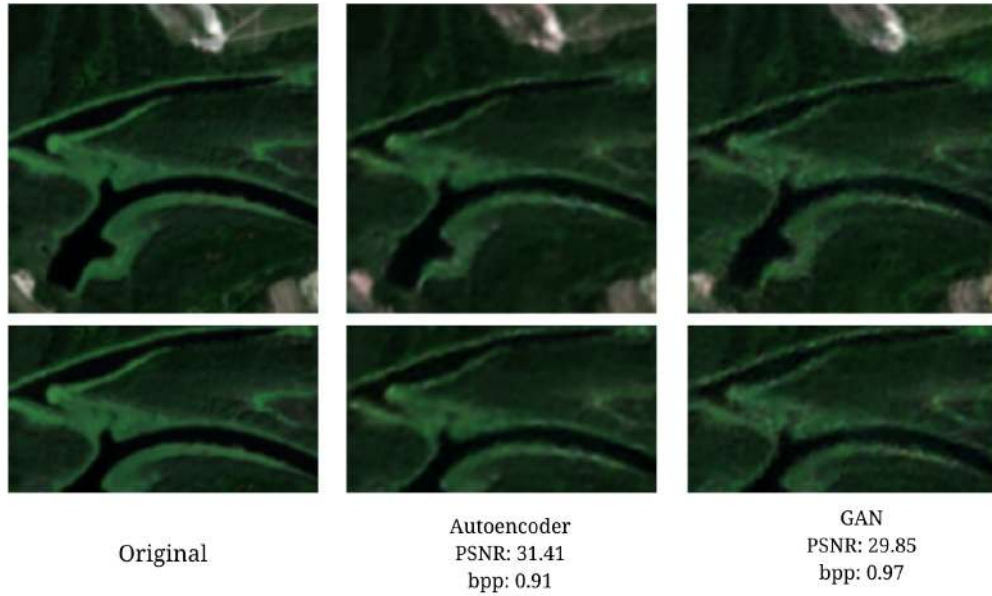


Figure 5.7: Reconstructed sample of SE GAN architecture: autoencoder and GAN,  $r_t = 0.8$

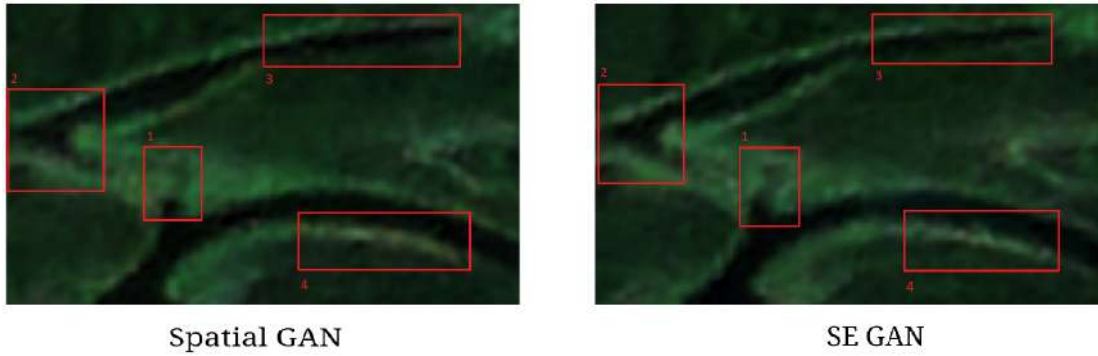


Figure 5.8: Close up comparison of spatial compression with HiFiC and spatio-spectral compression with SE GAN

Such small details might not be noticeable at low resolution but are quite appealing on bigger images. They also indicate some limitations of the spatial architecture, such as blurriness in lines and borders, and ignoring small objects in the reconstruction.

### 5.3.2 3D CGAN

Usage of channels as a third dimension may let us use 3D convolutions, just like in video compression. However, this method is more complicated in its nature.

Table 5.6 contains the PSNR and bpp values of the 3D CGAN models. You can see a clear improvement over spatial compression at larger bit-rates  $r_t = \{0.6, 0.8, 1\}$ , but there is a performance drop at lower target rates ( $r_t = \{0.2, 0.4\}$ ).

$r_t$	AE		GAN	
	PSNR	bpp	PSNR	bpp
0.2	24.04	0.24	23.5	0.24
0.4	26.39	0.44	25.38	0.44
0.6	28.0	0.65	26.79	0.66
0.8	29.24	0.88	27.84	0.96
1	30.48	1.17	28.8	1.3

Table 5.6: List of average PSNR and bpp values of 3D CGANs at different target rates.

Similar to SE GAN, 3D CGAN does not have obvious improvements on the visual appearance of the images (Fig. 5.9), but it does have its own changes in the details. Figure 5.10 marks up where 3D CGANs have their strengths. Areas No 1 and 2 highlight the details and shape recognition. It is not as good as SE GAN, but better than spatial compression. Areas 3 and 4 show how lines are more defined with the contrasted color. Even though their borders are

## 5 Experimental Results and Discussions

still blurry, contrast gives better representation and make them seem sharper to the human eye. Areas 5 and 6 show color changes. 3D CGAN made those parts darker, than they should be. This might be the fault of the autoencoder, because on Figure 5.9 they are more accentuated than on the original image.

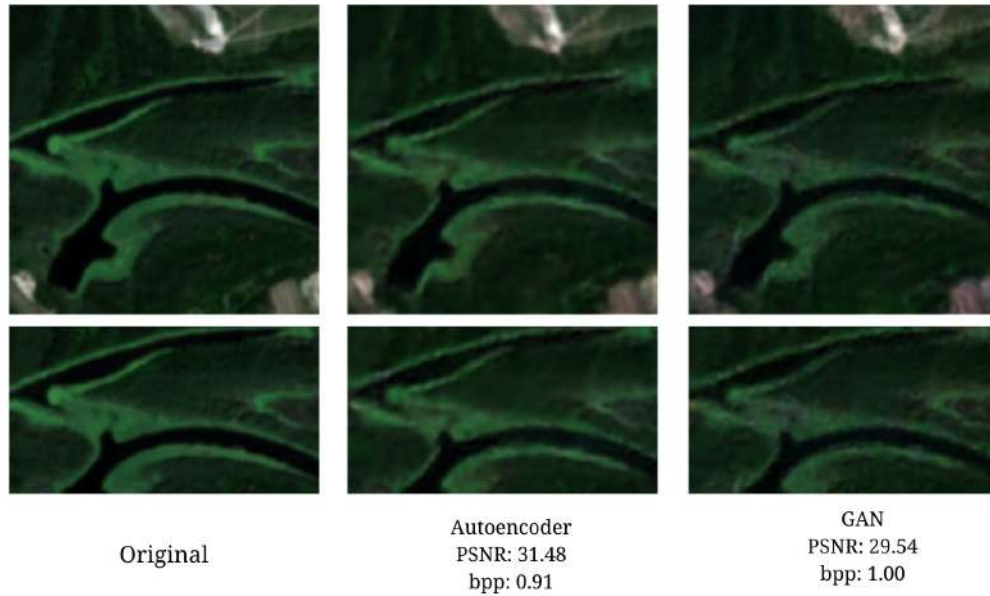


Figure 5.9: Reconstructed sample of 3D CGAN architecture: autoencoder and GAN at  $r_t = 0.8$

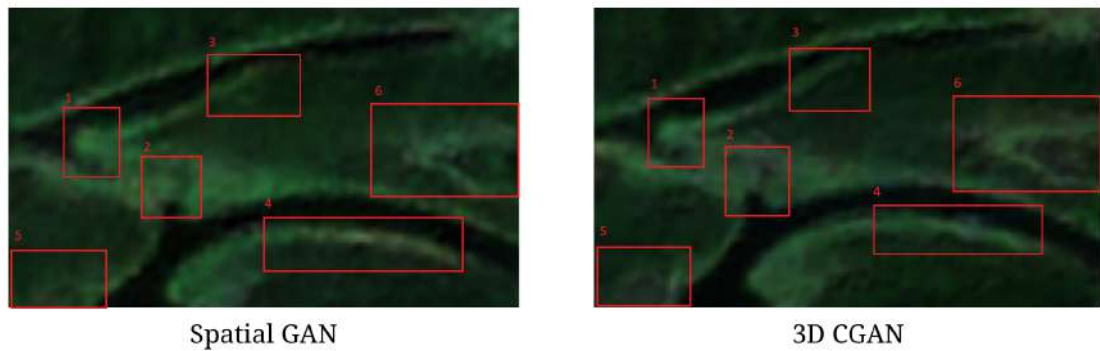


Figure 5.10: Close up comparison of spatial compression with HiFiC and spatio-spectral compression with and 3D CGAN.

## 5.4 Comparison between Spatial and Spatio-Spectral Compression

After receiving all the results, we compared the three architectures amongst each other. Figure 5.11 shows all gathered metrics combined. There are intersections on some bit-rates, but it is clear that the SE block has better results overall, spatial GAN is better on smaller bit-rates ( $< 0.6$  bpp), and 3D CGAN is better on higher ones ( $\geq 0.6$  bpp).

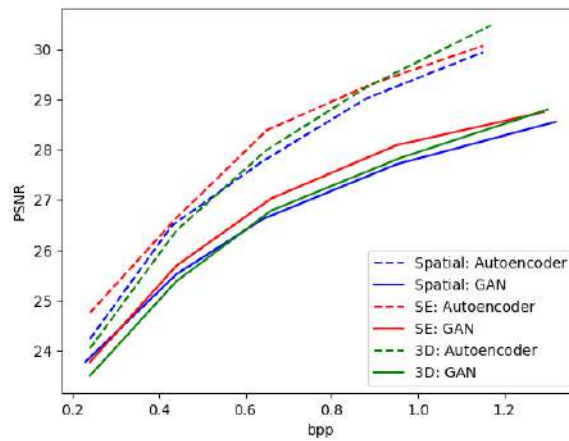


Figure 5.11: Average PSNR and bpp of all models

Spatio-spectral compression gain appears mostly in the details of the recovered images as shown in Figures 5.8 and 5.10. It is also worth noting that the SE GAN showed better performance than the 3D CGAN in almost half the training time. Also, the SE block is easier to plug-in into already existing architectures.

In the Appendix you can find even more visual comparisons on different bitrates and architectures.

## 6 Conclusion and Discussion

With the amount of RS data growing on a daily basis, so does the need for more efficient storage and transmission. Since the traditional methods are limited, exploring the new ways of big data compression are key to solving this problem.

The goal of this thesis was to adapt an existing spatial compression end-to-end optimized model for multispectral RS images and come up with spatio-spectral compression architectures that perform well on multispectral data. To this end, we proposed two architecture modifications to the existing RGB based compression work of HiFiC by using 1) SE blocks and 2) 3D convolutions. As a first step, we adapted the HiFiC model to multispectral data and tested it on the BigEarthNet archive for getting a baseline spatial compression model. The main changes to the architecture were: 1) new calculation of distortion loss adapted specifically for BigEarthNet; 2) a new data load process; and 3) a new set of hyperparameters values. Those adaptations made it possible to spatially compress a multispectral image. However, spatial compression does not consider the spectral redundancies present in multispectral data. So, two improvements were proposed: SE GAN and 3D CGAN.

SE GAN consists of SE blocks that can be easily plugged in between layers of an existing network. It does channel attention and weights each band of multispectral image, depending on their inter-dependencies. Similarly, 3D CGANs made up of 3D convolutions, can be applied to the generator (and the encoder, if present), based on convolutional networks. Replacing 2D convolutions with 3D ones can help to catch the spectral redundancies and compress the images more effectively. The proposed architectural changes for spatio-spectral compression were evaluated on the BigEathNet dataset and have shown slight improvement over the spatial compression with HiFiC. The results were better than spatial compression: higher PSNR values for the same bpp and nicer looking reconstructions achieved though details and better coloration.

### 6.1 Further Studies

SE blocks and 3D convolutions are the baselines of many more advanced frameworks for image compression. Since experiments showed a slight quality increase compared to pure spatial compression with using SE blocks and 3D convolutions only, more spectral compression techniques should be tested in future studies.

It can also be helpful to test GAN-based approaches on high-resolutional RS images to see what limitation are still present and what were caused by the small image size. This can help better elucidate the problem and reduce disadvantages like blurriness, poor details, and discoloration with the help of the new architecture improvements.

Even with a proper ablation study, there is still a lot of space for testing, such as increasing the bit-rate range. We decided to stay closer to the original HiFiC to make it easier to compare

the resultant architectures, this is why higher bit-rate ranges did not perform as expected (actual bit-rate is far from the targeted one). Replacing more hyperparameters or overwriting the rate-calculation part could help break through that limit.

Also, the loss functions can be improved. Right now LPIPS, which is one of the reasons for the original HiFiC's performance, is applied only on the RGB bands. Recalculating this metric for all bands could be an alternative.

This thesis paves the way for future research in spatio-spectral compression. Though the proposed architectures improved the results, we observed only a small margin of improvement in the performance. Ideally with spectral redundancy we expect a huge gain in performance. Hence, much can be explored surrounding these architectures to further improve the results.

## Bibliography

- [1] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. *End-to-end optimization of non-linear transform codes for perceptual quality*. 2016. arXiv: [1607.05006](https://arxiv.org/abs/1607.05006) [cs.IT].
- [2] Johannes Ballé et al. *TensorFlow Compression*. URL: <https://github.com/tensorflow/compression> (visited on 01/20/2022).
- [3] Johannes Ballé et al. *Variational image compression with a scale hyperprior*. 2018. arXiv: [1802.01436](https://arxiv.org/abs/1802.01436) [eess.IV].
- [4] Fabrice Bellard. *BPG Image format*. URL: <http://bellard.org/bpg/> (visited on 01/28/2022).
- [5] Zhengxue Cheng et al. *Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules*. 2020. arXiv: [2001.01568](https://arxiv.org/abs/2001.01568) [eess.IV].
- [6] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014.
- [7] Jie Hu, Li Shen, and Gang Sun. “Squeeze-and-Excitation Networks”. In: *CoRR* abs/1709.01507 (2017). arXiv: [1709.01507](https://arxiv.org/abs/1709.01507). URL: <http://arxiv.org/abs/1709.01507>.
- [8] Yueyu Hu et al. *Learning End-to-End Lossy Image Compression: A Benchmark*. 2021. arXiv: [2002.03711](https://arxiv.org/abs/2002.03711) [eess.IV].
- [9] Forrest N. Iandola et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size”. In: *CoRR* abs/1602.07360 (2016). arXiv: [1602.07360](https://arxiv.org/abs/1602.07360). URL: <http://arxiv.org/abs/1602.07360>.
- [10] Khawar Islam et al. *Image Compression with Recurrent Neural Network and Generalized Divisive Normalization*. 2021. arXiv: [2109.01999](https://arxiv.org/abs/2109.01999) [eess.IV].
- [11] Nick Johnston et al. *Improved Lossy Image Compression with Priming and Spatially Adaptive Bit Rates for Recurrent Networks*. 2017. arXiv: [1703.10114](https://arxiv.org/abs/1703.10114) [cs.CV].
- [12] Fanqiang Kong et al. “Spectral&Spatial Feature Partitioned Extraction Based on CNN for Multispectral Image Compression”. In: *Remote Sensing* 13.1 (2021). ISSN: 2072-4292. DOI: [10.3390/rs13010009](https://doi.org/10.3390/rs13010009). URL: <https://www.mdpi.com/2072-4292/13/1/9>.
- [13] Alex Krizhevsky. “One weird trick for parallelizing convolutional neural networks”. In: *CoRR* abs/1404.5997 (2014). arXiv: [1404.5997](https://arxiv.org/abs/1404.5997). URL: <http://arxiv.org/abs/1404.5997>.
- [14] P. Luigi Dragotti, G. Poggi, and A.R.P. Ragozini. “Compression of multispectral images by three-dimensional SPIHT algorithm”. In: *IEEE Transactions on Geoscience and Remote Sensing* 38.1 (2000), pp. 416–428. DOI: [10.1109/36.823937](https://doi.org/10.1109/36.823937).



- [15] Fabian Mentzer et al. “Conditional Probability Models for Deep Image Compression”. In: *CoRR* abs/1801.04260 (2018). arXiv: [1801.04260](https://arxiv.org/abs/1801.04260), URL: <http://arxiv.org/abs/1801.04260>.
- [16] Fabian Mentzer et al. *Conditional Probability Models for Deep Image Compression*. 2019. arXiv: [1801.04260 \[cs.CV\]](https://arxiv.org/abs/1801.04260).
- [17] Fabian Mentzer et al. *High-Fidelity Generative Image Compression*. 2020. arXiv: [2006.09965 \[eess.IV\]](https://arxiv.org/abs/2006.09965).
- [18] David Minnen, Johannes Ballé, and George Toderici. *Joint Autoregressive and Hierarchical Priors for Learned Image Compression*. 2018. arXiv: [1809.02736 \[cs.CV\]](https://arxiv.org/abs/1809.02736).
- [19] J. von Neumann. “Zur Theorie der Gesellschaftsspiele”. In: *Mathematische Annalen* 100 (1928), pp. 295–320. URL: <http://eudml.org/doc/159291>.
- [20] Vinicius Alves de Oliveira et al. “Reduced-Complexity End-to-End Variational Autoencoder for on Board Satellite Image Compression”. In: *Remote Sensing* 13.3 (2021). ISSN: 2072-4292. DOI: [10.3390/rs13030447](https://doi.org/10.3390/rs13030447), URL: <https://www.mdpi.com/2072-4292/13/3/447>.
- [21] Soille P et al. “Towards a JRC Earth Observation Data and Processing Platform”. In: Proceedings of the 2016 conference on Big Data from Space. LB-NA-27775-EN-N. European Commission. Luxembourg (Luxembourg): Publications Office of the European Union, 2016. ISBN: 978-92-79-56980-7. DOI: [10.2788/854791](https://doi.org/10.2788/854791), URL: <http://dx.doi.org/10.2788/854791>.
- [22] Majid Rabbani and Rajan Joshi. “An overview of the JPEG 2000 still image compression standard”. In: *Signal Processing: Image Communication* 17.1 (2002). JPEG 2000, pp. 3–48. ISSN: 0923-5965. DOI: [https://doi.org/10.1016/S0923-5965\(01\)00024-8](https://doi.org/10.1016/S0923-5965(01)00024-8), URL: <https://www.sciencedirect.com/science/article/pii/S0923596501000248>.
- [23] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: <http://arxiv.org/abs/1409.1556>.
- [24] N. J. A. Sloane and Aaron D. Wyner. “Coding Theorems for a Discrete Source With a Fidelity Criterion”. In: *Institute of Radio Engineers, International Convention Record*, vol. 7, 1959.” In: *Claude E. Shannon: Collected Papers*. 1993, pp. 325–350. DOI: [10.1109/9780470544242.ch21](https://doi.org/10.1109/9780470544242.ch21).
- [25] Gencer Sumbul et al. “BigEarthNet: A Large-Scale Benchmark Archive For Remote Sensing Image Understanding”. In: *CoRR* abs/1902.06148 (2019). arXiv: [1902.06148](https://arxiv.org/abs/1902.06148), URL: <http://arxiv.org/abs/1902.06148>.
- [26] George Toderici et al. *Variable Rate Image Compression with Recurrent Neural Networks*. 2015. arXiv: [1511.06085 \[cs.CV\]](https://arxiv.org/abs/1511.06085).

## Bibliography

- [27] Du Tran et al. “C3D: Generic Features for Video Analysis”. In: *CoRR* abs/1412.0767 (2014). arXiv: [1412.0767](https://arxiv.org/abs/1412.0767). URL: <http://arxiv.org/abs/1412.0767>.
- [28] Z. Wang, E.P. Simoncelli, and A.C. Bovik. “Multiscale structural similarity for image quality assessment”. In: *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*. Vol. 2. 2003, 1398–1402 Vol.2. DOI: [10.1109/ACSSC.2003.1292216](https://doi.org/10.1109/ACSSC.2003.1292216).
- [29] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: [10.1109/TIP.2003.819861](https://doi.org/10.1109/TIP.2003.819861).
- [30] Lirong Wu, Kejie Huang, and Haibin Shen. *A GAN-based Tunable Image Compression System*. 2020. arXiv: [2001.06580](https://arxiv.org/abs/2001.06580) [eess.IV].
- [31] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *CoRR* abs/1801.03924 (2018). arXiv: [1801.03924](https://arxiv.org/abs/1801.03924). URL: <http://arxiv.org/abs/1801.03924>.

# Appendix

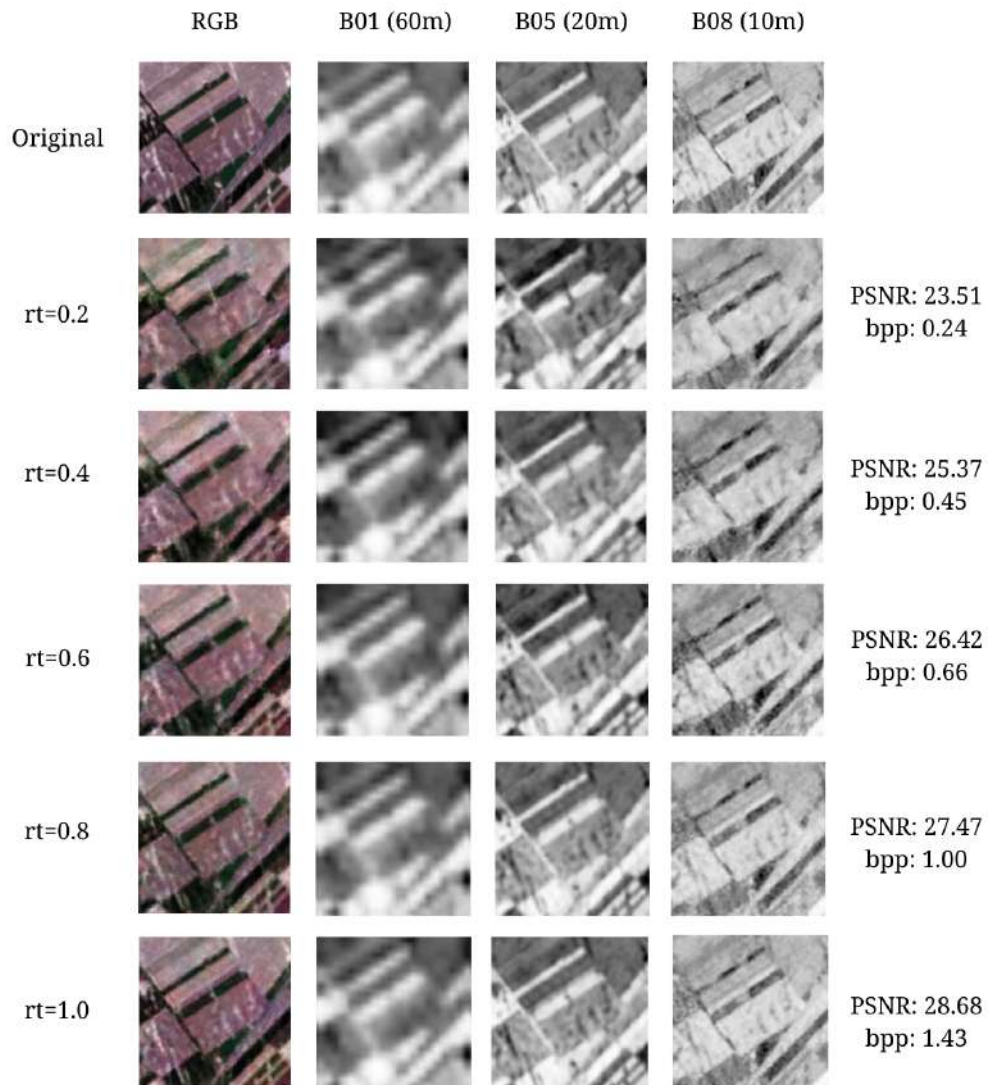


Figure A.1: Comparison on different bands and  $r_t$  values of spatial GAN

Appendix

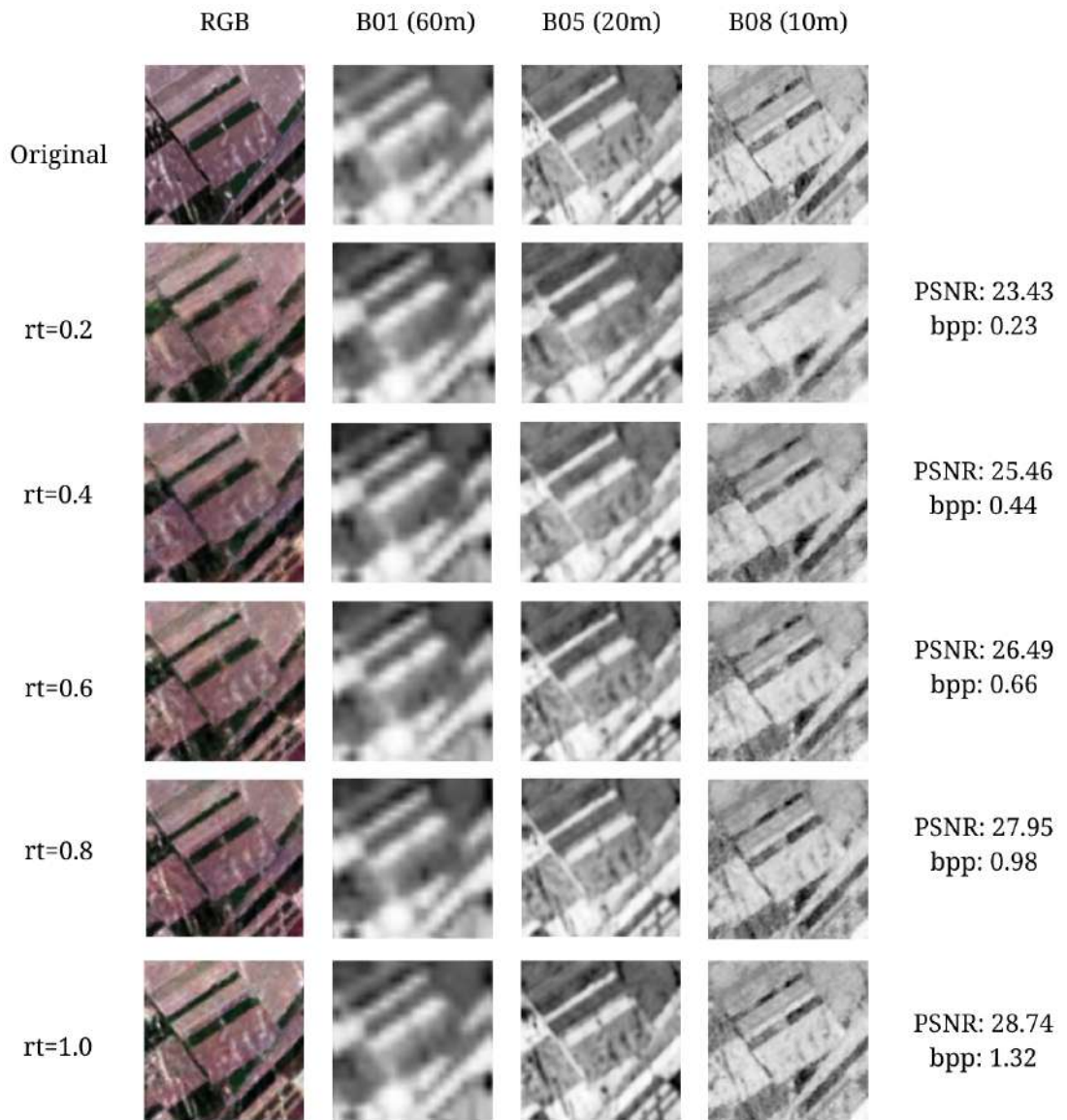


Figure A.2: Comparison on different bands and  $r_t$  values of SE GAN

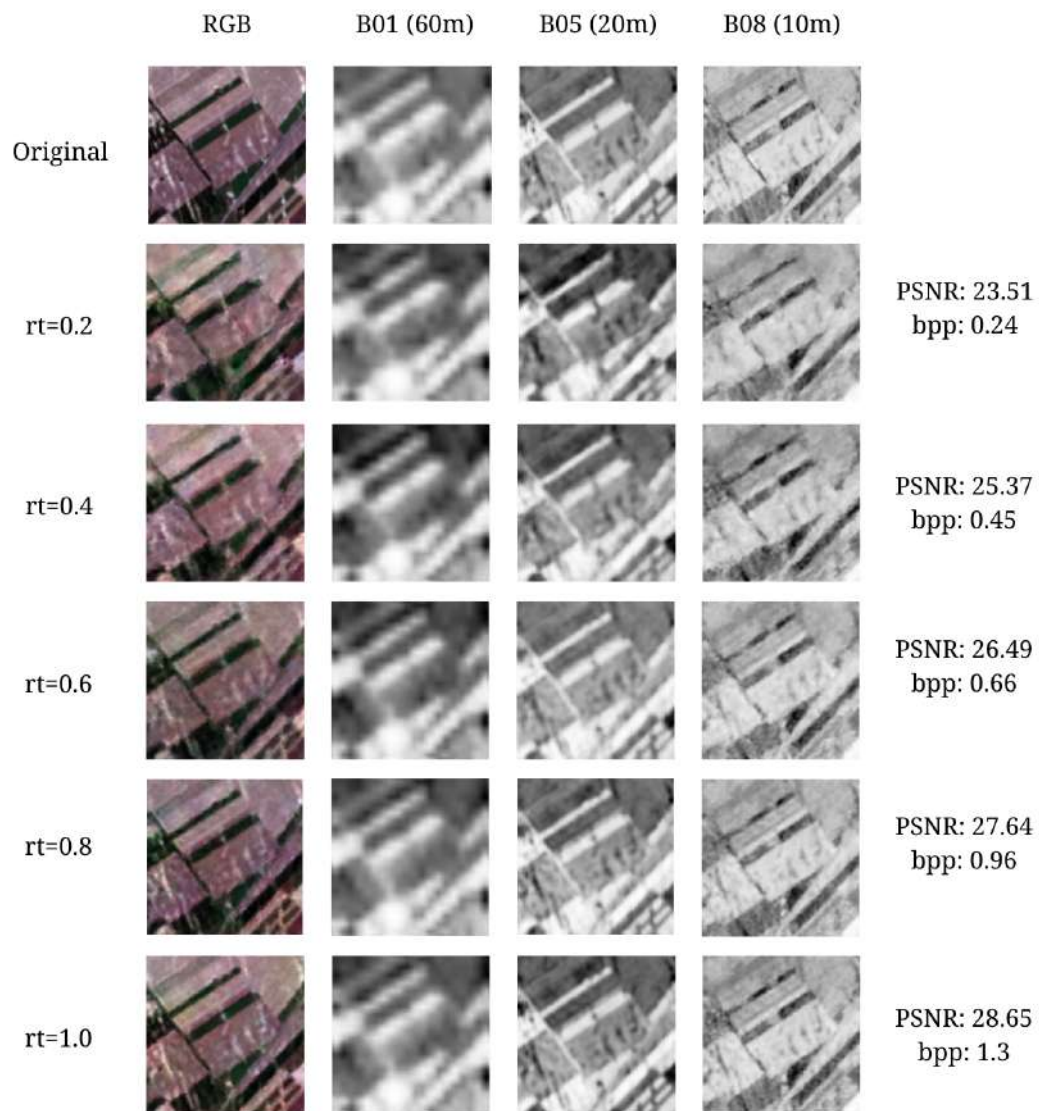


Figure A.3: Comparison on different bands and  $r_t$  values of 3D CGANs

Appendix

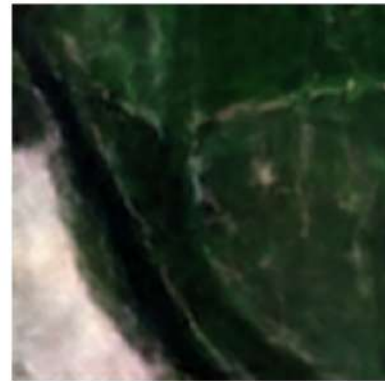


Original

rt=0.2



Autoencoder  
PSNR: 26.61  
bpp: 0.24



GAN  
PSNR: 25.90  
bpp: 0.24

rt=0.4



Autoencoder  
PSNR: 29.01  
bpp: 0.45



GAN  
PSNR: 27.42  
bpp: 0.45

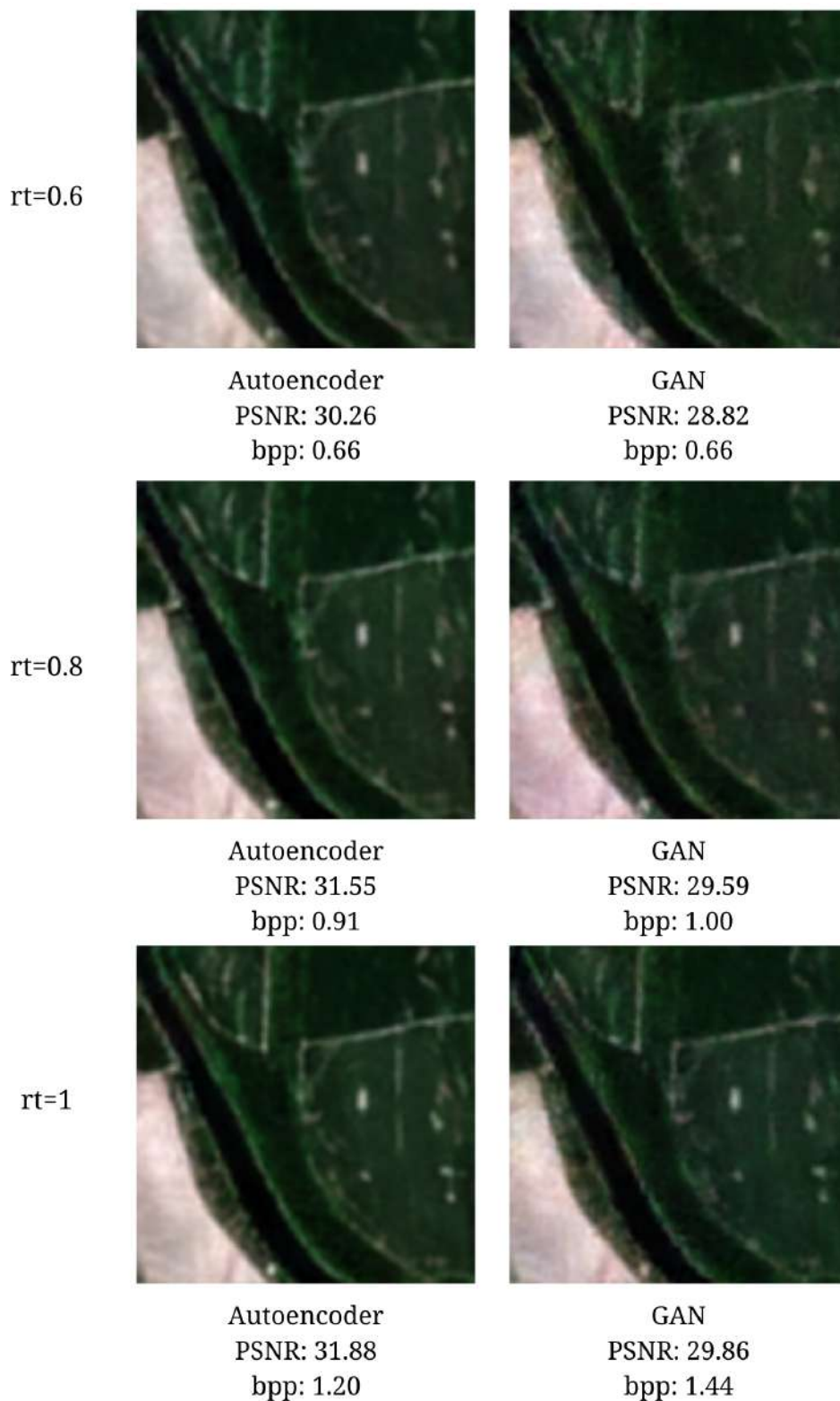


Figure A.4: Spatial autoencoder and GAN results at different bit rates (large samples)

*Appendix*



Original



Spatial GAN  
PSNR: 25.66  
bpp: 0.24



SE GAN  
PSNR: 25.28  
bpp: 0.24



3D CGAN  
PSNR: 24.86  
bpp: 0.24

rt = 0.2





Spatial GAN  
PSNR: 27.20  
bpp: 0.44



SE GAN  
PSNR: 27.44  
bpp: 0.44



3D CGAN  
PSNR: 27.25  
bpp: 0.43

rt = 0.4



Spatial GAN  
PSNR: 28.34  
bpp: 0.64



SE GAN  
PSNR: 28.55  
bpp: 0.66



3D CGAN  
PSNR: 28.66  
bpp: 0.65

rt = 0.6

Appendix

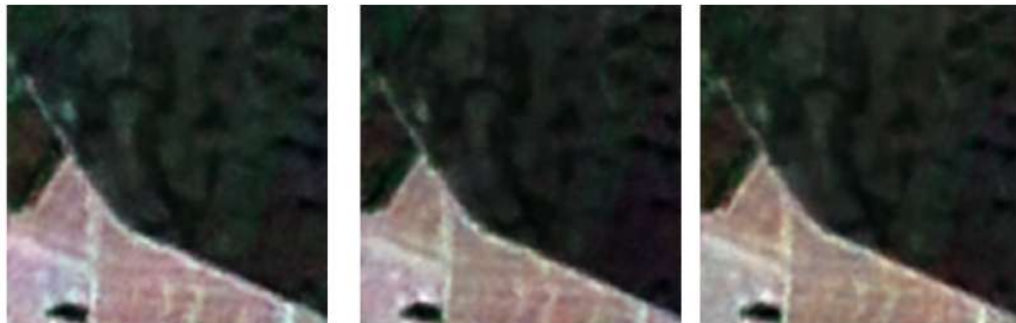


Spatial GAN  
PSNR: 29.53  
bpp: 0.91

SE GAN  
PSNR: 29.53  
bpp: 0.93

3D CGAN  
PSNR: 29.84  
bpp: 0.92

rt = 0.8



Spatial GAN  
PSNR: 29.87  
bpp: 1.27

SE GAN  
PSNR: 29.72  
bpp: 1.23

3D CGAN  
PSNR: 30.46  
bpp: 1.22

rt = 1

Figure A.5: Spatial GAN, SE GAN and 3D CGAN on different bit rates