SAR image synthesis using text conditioned pre-trained generative AI models

Nicolas Trouvé, Nathan Letheule, Olivier Lévêque, Ilias Rami, Elise Colin

SAR image synthesis using text conditioned pre-trained generative AI models

Nicolas Trouvé[a], Nathan Letheule[a], Olivier Lévêque [a], Ilias Rami [b], and Elise Colin[c]
[a]DEMR-ONERA, University Paris-Saclay (France)
[b]IOGS, Institut d'Optique Graduate School (France)
[c]DTIS-ONERA, University Paris-Saclay (France)

Abstract

We explore the utilization of artificial intelligence (AI) generative models for creating high-resolution airborne Synthetic Aperture Radar (SAR) images. Our methodology involves the use of a text-conditioned latent diffusion architecture to train a generative model. We use a database of high-resolution SAR images obtained from the SETHI sensor at ONERA for training purposes. This model is capable of generating synthetic images based on textual prompts provided by users. Additionally, we illustrate the model's versatility for various applications, such as generating SAR images from hand-drawn sketches.

1 Introduction

Synthetic Aperture Radar (SAR) image synthesis and simulation have diverse applications in sensor design and signal processing algorithm evaluation. Traditionally, this field relied on physics-based simulations using electromagnetic modeling for vehicles and scenes. However, the emergence of deep neural network techniques has led to efforts to apply these methods to SAR image generation.

Early network architectures predominantly utilized convolutional networks and the Generative Adversarial Network (GAN) framework. These networks, comprising a generator and discriminator, were constrained in scale, often working with small image sizes. They typically operated on pairs of images, such as optical and SAR images or different-frequency SAR images of the same area, aiming to convert one image type into another, akin to style transfer. This approach required training from scratch, posing challenges related to model depth and dataset size.

Recent research has introduced foundation models, exemplified by Meta's Segment Anything Model (SAM), Llama, and Runway's Stable Diffusion. These transformer-based models, trained for extensive hours on large open datasets, boast billions of parameters and exceptional generalization capabilities, despite being trained on internet-sourced data.

Compared to previous models, foundation models offer the advantage of minimal fine-tuning, leveraging their inherent capabilities. However, they require powerful GPUs and necessitate careful fine-tuning on smaller datasets to prevent overfitting.

In this paper, we present the results of fine-tuning the Stable Diffusion foundation model using real ONERA SETHI X band SAR images. We start in section 2 with a brief overview of the model's architecture, components, and initial training data. Next, we discuss various fine-tuning methods relevant to our domain. We detail our image dataset and automated image captioning process in section 3, concluding with examples and use cases, such as text-to-image and sketch-to-image generation in section 4, before concluding.

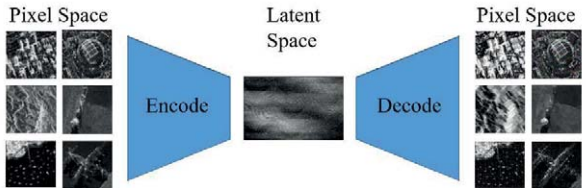

Figure 1 Autoencoder principle, images in *pixel space* are encoded into their *latent space* equivalent, and can be decoded back in *pixel space*

2 Text conditioned latent diffusion

In this section, we describe the architecture of the latent diffusion model as initially outlined in [1] and further elaborated in [2]. This description is tailored for the conference audience; readers acquainted with the principles may proceed to the subsequent section. As summarized in Figure 4, a latent diffusion model comprises three primary components:

- Variational Auto Encoder
- Text Encoder
- U-net : Noise predictor

Their purpose is detailed in the following sections.

2.1 Variational Auto Encoder

An autoencoder, a fundamental component in many deep learning architectures, consists of two parts: an encoder

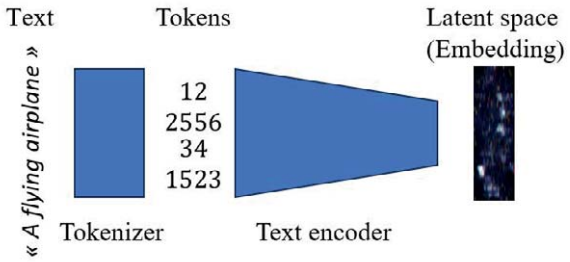

Figure 2 Text encoder, words are convered to a vector through the tokenizer which is further process by a text encoder into an embedding

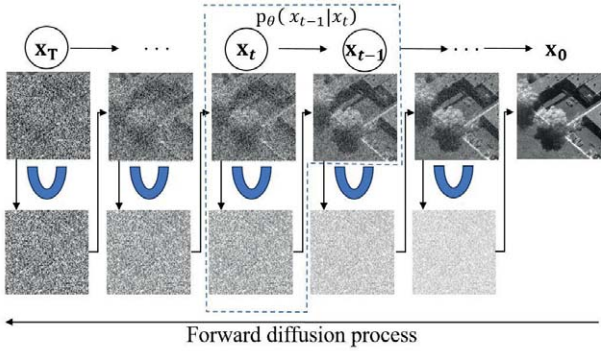

Figure 3 Iterative use of an U-net model as a noise estimator to substract noise at each step (exemple in pixel space)

and a decoder. The encoder compresses a natural image from *pixel space* into a more compact vector in *latent space*. This compression, as used in [3] and this paper, reduces a $512 \times 512 \times 3$ image into a $64 \times 64 \times 4$ sized latent representation, a reduction factor of 48. Despite this significant compression, the decoder can efficiently revert the process as shown Figure 1, restoring the image to its original space and size, leveraging the manifold hypothesis [4]. However, this compression is not lossless and can impact image quality. The effect on the compressed image depends on its similarity to the images used during the training stage. Variational Auto Encoders (VAE), an advanced form of autoencoders, differ as the encoder output is a distribution of vectors (mean and variance) rather than a single deterministic vector. During learning, a vector is sampled from this distribution before being processed back through the decoder. This alteration enforces latent normalization and induces latent continuity, granting the VAE generative properties. As an example most interpolated latents can be decoded into a coherent image, enhancing the model's generative capabilities.

2.2 Text Encoder

The role of a text encoder is to process text or prompts for use as conditioning in the generative process. Initially, a tokenizer converts the word sequence into an embedding vector, where each word or sub-word, named *token*, is mapped to a number via a dictionary. The text encoder fixes the maximum number of tokens. This raw embedding undergoes further processing through a text transformer, which outputs a conditioning vector representing the text sequence's meaning rather than the individual words. For instance, words with similar meanings (e.g., plane and aircraft) will produce similar vectors as illustrated Figure 2. The attention processing within the model also allows the relative position of words, sentence structure, and grammar to influence the output vector.

In the model utilized in this paper (ViT-L/14 Clip [5]), the text encoder has a modest number of parameters (123 Millions compared to the 150 Billions of GPT3 [6]), limiting its understanding of text descriptions. Consequently, the model may often misinterpret detailed prompts involving relative positions or specific item colors.

2.3 Noise predictor U-net

The noise predictor, central to the diffusion process, aims to estimate noise in a given input image. The analogy of diffusion arises from the mathematical similarity between the chosen noise modeling and the physical modeling of particles dispersing in a fluid over time, progressively obscuring the initial state (i.e., the image without noise). The mathematical model, involving differential equations, is inverted by the network in an effort to recover the initial state from a subsequent diffuse state.

Employing the U-net architecture [7], the noise predictor is trained on pairs of images with varying levels of added noise, estimating the artificially added noise. As the largest network in the latent diffusion architecture, the noise-predicting U-net can process all frequencies and scales of noise simultaneously due to its convolutional structure. Although theoretical denoising could occur in a single step, practical implementation benefits from iterative denoising for enhanced output results as shown in Figure 3. The model estimates the noise amount and removes only a portion according to a predetermined noise schedule, which define the noise removal at each step (linear or exponential). Utilizing more steps reduces estimation errors at the expense of increased computational time.

In latent diffusion models, the U-net receives the text encoder's output as a conditioning vector, applied at every U-net layer and each iterative denoising step using the cross-attention mechanism [8]. During training, this conditioning serves as additional prior information about the image content, aiding the U-net in improving its noise estimation. It is notable that while diffusion could occur directly in pixel space, operating in a sparsely compressed space enhances the model's efficiency.

2.4 Assembled workflow

2.4.1 Text to image generation

In text-to-image synthesis, as illustrated in Figure 4, the aforementioned models operate as follows. An initial latent noise image is generated based on a seed. The text input, or prompt, is processed through the text encoder into a conditioning vector. Optionally, a second text can be processed similarly to serve as negative conditioning (named Nega-

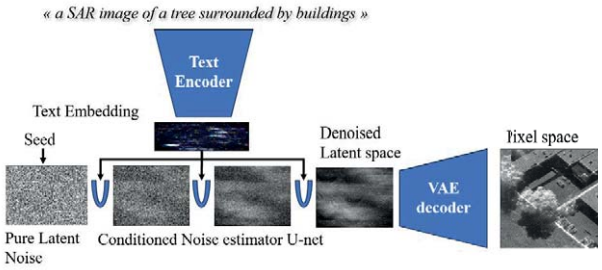

Figure 4 Text to image workflow involving a text encoder, a variational auto encoder (VAE) and a noise estimating U-net operating in latent space

tive Prompt), preventing certain concepts from appearing in the resultant image. Over a predetermined number of steps, and following a noise schedule, the U-net, conditioned by the embeddings, estimates and removes noise from the latent image at each step. Upon complete noise removal, the latent is decoded using the VAE decoder into a pixel space image, rendering it interpretable by humans.

2.4.2 Image to image

An alternative application of the models begins with an existing image. Instead of using pure latent noise, the input image is encoded through the VAE, and a variable amount of noise is added before proceeding with the same text-to-image process using a noise schedule adapted to the amount of latent noise added. The applications of this process are further elaborated in the final section of this paper.

2.5 Training

2.5.1 Initial Training

The first published Stable Diffusion model [9], with both architecture and weights available and a reasonable model size (860 million parameters for the U-net), is executable on consumer hardware, making it a prime foundation model for text-to-image generation.

The initial training utilized the LAION5B database [10], comprising 5.85 billion images from open sources and random internet scraping. Automatic captioning was performed using CLIP [5], and the database was filtered for content, size, and quality. The training consumed approximately 150,000 hours on A100 GPUs for version 1.0 and nearly 600,000 hours for version 1.5. Initially, the VAE part of the model is trained independently and then frozen, ensuring a converged latent space representation as the process foundation. The pre-trained and frozen text encoder from CLIP was used during the U-net training, which constituted the majority of the training process.

A examination of the LAION5B database reveals few representation of SAR images, primarily downscaled large-scale images from general news articles by space agencies or regular newspapers covering havoc. The base model's text-to-image generation further underscores its inability to produce credible SAR images, necessitating additional training.

2.6 Fine Tuning

Since Stable Diffusion was made open source in [3], the community has undertaken extensive work to adapt the base model to various needs. A primary challenge in fine-tuning a foundation model lies in managing the learning process with datasets that are several orders of magnitude smaller than those used in initial training. Maintaining a critical ratio between the model's number of parameters and the amount of data is vital to prevent overfitting, which could not only diminish the base model's ability to generate content within its initial learning spectrum but also impair the generalization capacity for new content introduced through fine-tuning. Another significant consideration is the computing power and time required for the fine-tuning process, which, in many cases, has been designed to be accessible to high-end consumer hardware.

Two main strategy can be developed and are explained in the following section.

2.6.1 Full or partial fine tuning

The second approach involves retraining components of the base model, often involving the U-net, optionally the text encoder, and in rare instances, the VAE. Fine-tuning the network entails updating all weights involved in the original network, adhering to the same process utilized in the initial training. This method is technically as memory and computationally intensive as the initial training, although the smaller database allows it to be executed within a timescale of hours on a single computer. This process is susceptible to all potential pitfalls arising from overlearning. The outcome is a completely new model, branching from the original, with new or modified capacities. While this method is ideally suited for training new concepts, it may require special attention when dealing with small database inputs. One method, known as DreamBooth [11], develop the concept of utilizing the model itself to generate regularization images, which are then fed back into the model during the fine-tuning process, as a strategy to prevent overlearning when using a very limited number of images (typically fewer than 10). These additional generated images, usually numbering in the thousands, are typically chosen to describe a broader concept than the subject of the fine-tuning, often referred to as a class (in our application, the class might be aerial photography). The result is that by augmenting the training set with images generated by the model, one engages in 'prior preservation'. This means that the fine-tuning process is enforced in a manner that does not impair the existing capacities of the model.

2.6.2 Low rank methods

The final approach involves modifying only a subsection of the model or adding an extra layer that intersects two existing layers. Typically, these modified layers are situated at critical points in the workflow, such as the cross-attention section of the model, which interfaces between the text encoder and the U-net. The remaining parts of the model stay frozen, making the process less resource-intensive than training the entire model in terms of both time and memory usage. Further optimizations leverage

the mathematical properties of these layers or employ so-called low-rank (LoRa [12]) models. The underlying concept is that a layer connecting M inputs to N outputs can be represented as a matrix, analogous to a linear algebra matrix. Assuming this matrix has a limited rank, the number of parameters requiring estimation can be significantly reduced. Experience indicates that this assumption holds, and ranks as low as 4 to 256 can sometimes suffice to enhance the model's capacities, depending on the complexity of the new concept. Various LoRa variants have been proposed in the literature, including LoHa with Hadamard product, LoKR with Kronecker product, LoCon modifying the convolution part of the network, and DyLoRa for adaptive rank estimation. One advantage is that enforcing a limited number of parameters through a restricted rank can reduce the risk of overlearning while maintaining the overall capacity of a very large model. Another is that, since the additional inclusions are relatively small, they can easily be toggled on and off at various stages of generation.

2.6.3 Model choice, data preparation and captioning

After evaluating various options and conducting numerous attempts, we concluded that, given the Stable Diffusion 1.5 base model and training on a concept significantly divergent from its initial training type, a full fine-tuning was necessary. Even a very high-rank LoRa was incapable of capturing all the complexity needed to accurately reproduce speckle, layover, and image composition. Our preliminary tests with Stable Diffusion XL [13], a substantially larger model, indicate that full fine-tuning could be superfluous for this model.
Training with or without prior preservation, utilizing aerial photography optical images as regularization pictures, yielded no significant changes in the output. Considering we do not require the model to maintain its ability to generate optical images, the final model presented was trained without them.

3 Dataset

3.1 ONERA's airborn systems and database

The SETHI remote sensing system is onboard a Falcon 20 [14], an aircraft developed by Dassault Aviation in the 1970s. This aircraft, equipped with its payload, is certified by European Aviation Safety Agency (EASA), allowing it to fly worldwide. Since its deployment 15 years ago, SETHI has acquired numerous SAR images (in Ku, X, L, and UHF frequency bands) covering various regions of the globe, including French Guiana, Gabon, Tunisia, Sweden, Norway, Greenland, and notably metropolitan France. All raw data, processed images, and associated metadata (trajectories, georeferencing, calibration, etc.) are organized and stored in an internal ONERA database. These archives can be queried via a Python API, facilitating the automation of post-processing tasks on a large volume of data and the construction of datasets for training AI methods. In this article, we used X-band SAR images from this database, covering regions in the south of mainland France, to build the training dataset. All images were normalized, resampled at a fixed ground square resolution of 0.35 cm then split into 512×512 sized images resulting in a total of more than 10000 images. Images were kept into their native SAR orientation, the range axis being in the same direction for all images.

3.2 Automated data captionning

Captioning a large number of SAR images is a challenging task, no known nor published model have been trained for this purpose. In this perspective, the creation of a text invitation, or "prompt", could be achieved by captioning another image source, such as an optical image sharing the same geographical coverage and acquired within a comparable timeframe.
This approach is made possible because networks trained on very large databases of conventional images for captioning have been published in the past years. The state of the art in captioning includes the following:

- CLIP [5], whose main advantage is generality. However, it is less effective for abstract tasks such as counting objects, describing, or interpreting spatial connections.
- GIT, which has a single image encoder and a single text decoder, is interesting for its versatility, with the ability to address three distinct problems: image-to-text transformation, video-to-text translation, and also solving Visual Question Answering (VQA) tasks.
- BLIP, an acronym for "Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation," relies on a multimodal pre-training architecture integrating encoder-decoders (MED) [15].

In our work, we chose to use the BLIP algorithm due to its ability to create comprehensive descriptions that incorporate spatial connectivity concepts with typical descriptions such as "a few trees gathered around a clearing" or to note the presence of a specific element like "a path that crosses the forest." The drawbacks of all the methods, is that their training set involve a major proportion or photograph taken at ground heights by hand held devices. Even if airborn or spaceborn optical images are much more common than SAR images in the databases used they are still underrepresented resulting in below average captions quality. Just like Stable Diffusion those models architecture and weights have been published and can also be further trained to improve their performance in our narrower domain. To this purpose two datasets were used together (RSICD [16] and UCM [17]).
The metrics we used were employed to compare BLIP's performance on remote sensing databases specific to our field of study. These evaluations led to a comparison of BLIP's performance before and after applying the fine-tuning method on the RSICD and UCM databases.
In the results shown in Table 1, using a wide variety of metrics commonly used to measure the similarity between two text sequences, it is noticed that for the RSICD database,

Methods \ Metrics	Blue-1	Blue-2	Blue-3	Blue-4	CIDEr	METEOR	ROUGE-L
CCSMLF	0.5759	0.3859	0.2832	0.2217	0.5297	0.2128	0.4455
SM Attention + LSTM	0.7571	0.6336	0.5385	0.4611	2.3563	0.3513	0.6458
Cross-Hierarchy attention	0.770	0.649	0.532	0.471	2.363	0.4238	0.651
ML Attention + semantic	0.7597	0.6421	0.5517	0.4623	2.3614	0.3543	0.6563
Multi-level attention	0.7905	0.6782	0.5742	0.5030	2.6309	**0.4640**	**0.7246**
Full Transformer	0.560			0.309	1.964	0.298	0.581
(Ours) BLIP-RSICD+UCM	**0.8029**	**0.6941**	**0.6061**	**0.5336**	**2.7239**	0.4208	0.7178

Table 1 Table of results for the BLUE, ROUGE, CIDEr, and METEOR metrics on the RSICD database using different captioning methods.

which has lower resolution, most metrics exhibit better results than the state-of-the-art. Consequently, we now have a refined BLIP network, adapted to optical aerial photography, thanks to training on two captioning databases for remote sensing, namely UCM and RSICD. This network was then used to automatically caption optical images sampled from the footprint of our SAR image database.

4 Applications and results

4.1 Image to text generation

Utilizing the SETHI's X-band database and automated captioning from the associated optical images via the fine-tuned BLIP model, training was conducted based on the Stable Diffusion 1.5 base model without prior preservation. Final training encompassed both the text encoder and U-net, with learning rates of 5e-6 and 5e-5 respectively, using batches of 16 for up to twelve epochs. 'An aerial view' was adopted as the activation token, but we will reference 'A SAR image of' in the example prompts provided in the illustrations. The learning progress can be visually seen in 5. As it can be seen on the sample images 6 The network is capable of generating different images in response to different prompts. This clearly demonstrates that text-guided SAR image simulation is feasible. The network successfully established a connection between the concept of "city" and building generation, as well as distinguishing fields, forests, and urban areas.

4.2 Combining with other type of conditioning

Currently, the main limitation lies in the constraint of prompts to capture the truly important concepts in a radar image. In particular, the images are generated without the ability to prioritize the spatial organization of specific content. Therefore, we can explore approaches such as the ControlNet [18] network, to guide generation not only based on text but also with spatial considerations using optical imagery, sketches, segmentation maps, or SAR imagery acquired at other resolution of frequencies. An example using an extra sketch input as an extra conditioning is shown Figure 7. Finally, a promising perspective would be to use the diffusion network to add details to a simulated SAR image generated using a conventional physical simulator like EMPRISE. This approach aims to transform our simulation, which can sometimes be too idealistic, into a SAR image that appears more realistic.

5 Conclusion

In summary, this article presents an innovative approach to generate SAR images using text-to-image techniques, harnessing the power of diffusion networks and large language models. Within the existing landscape of networks, radar images are notably scarce. To overcome this limitation, we chose to leverage descriptions rooted in the optical modality, which is far more prevalent, to characterize these rare and intricate radar images. In doing so, we fully harness the potential of these extensive pre-trained networks originally trained on large datasets, to benefit radar image simulation.

We employed a two-tier fine-tuning strategy: at the diffusion model level, we customized the model to high-resolution aerial SAR images from SETHI (ONERA), and at the captioning model level (BLIP), we tailored the generation of descriptions to suit remote sensing images.

This novel approach has yielded promising results. Leveraging recent remarkable advancements in large language models, this work represents a significant step toward achieving realism in generated images.

As we continue to explore the boundaries of AI-driven image synthesis, the synergy between diffusion models and language-based guidance presents substantial potential for a wide range of applications in remote sensing.

6 Literature

[1] Sohl-Dickstein et al., "Deep unsupervised learning using nonequilibrium thermodynamics," in *ICML*. PMLR, 2015.

[2] Ho et al., "Denoising diffusion probabilistic models," *Advances in NeurIPS*, 2020.

[3] Rombach et al., "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

[4] Brown et al., "Verifying the union of manifolds hypothesis for image data," 2023.

[5] Radford et al., "Learning transferable visual models from natural language supervision," 2021.

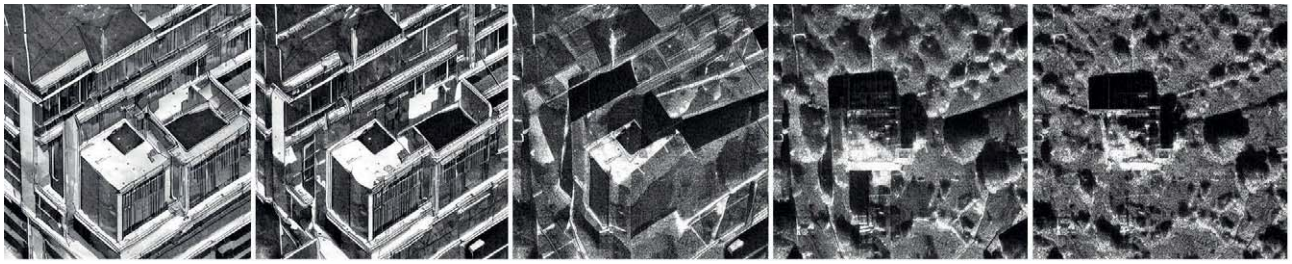

Figure 5 Training progression illustrated through a sample image generated with the same seed and prompt "a SAR image of a building". From left to right Epochs 1, 3, 6, 9 and 12. The coherence given by the seed shows the image evolving from an artwork into a SAR image

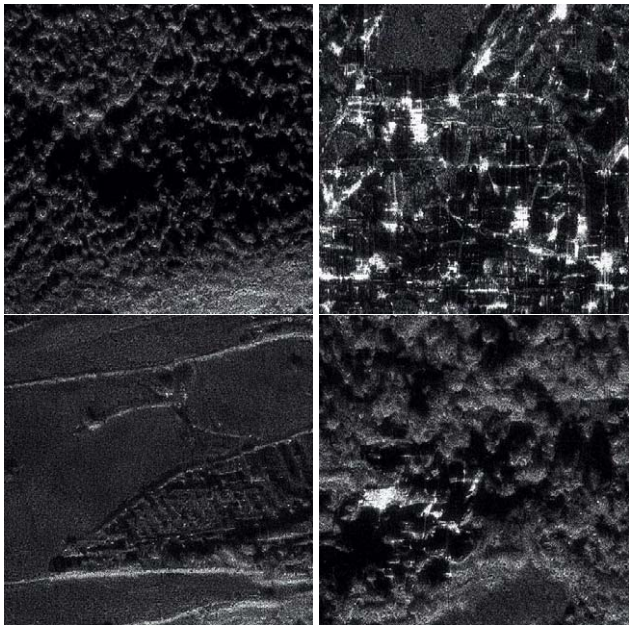

Figure 6 Examples of text to image generation using "A sar image of a forest" for the top left image, "A sar image of a city" for the top right image, "A sar image of a field" for the bottom left image, "A sar image of a forest around a city" for the bottom right image

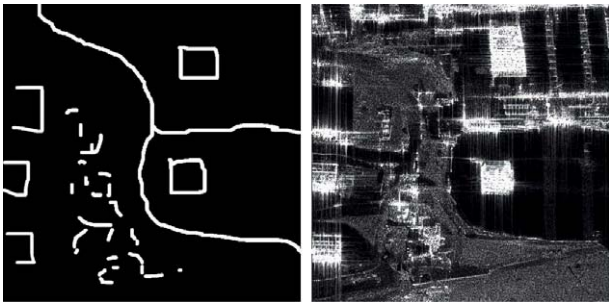

Figure 7 On the left a hand drawn scribble used as an input in a ControlNet model added as a extra conditionning to the image generation. On the right the resulting generated image.

[6] Brown et al., "Language models are few-shot learners," in *Advances in NeurIPS*, 2020.

[7] Ronneberger et al., "U-net: Convolutional networks for biomedical image segmentation," in *ICMICCAI*, 2015.

[8] Gheini et al., "Cross-attention is all you need: Adapting pretrained Transformers for machine translation," in *Proceedings on EMNLP*, Nov. 2021.

[9] Rombach et al., "High-resolution image synthesis with latent diffusion models," in *CVPR*, June 2022.

[10] Schuhmann et al., "Laion-5b: An open large-scale dataset for training next generation image-text models," 2022.

[11] Ruiz et al., "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *CVPR*, 2023.

[12] Hu et al., "Lora: Low-rank adaptation of large language models," 2021.

[13] Podell et al., "Sdxl: Improving latent diffusion models for high-resolution image synthesis," *arXiv preprint arXiv:2307.01952*, 2023.

[14] Baqué et al., "Sethi : Review Of 10 Years Of Development And Experimentation Of The Remote Sensing Platform," in *International Radar Conference (RADAR)*, 2019.

[15] Li et al., "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *ICML*. PMLR, 2022.

[16] Lu et al., "Exploring models and data for remote sensing image caption generation," *TGRS*, 2017.

[17] Ali and , "UCM image dataset," Aug 2018.

[18] Zhang et al., "Adding conditional control to text-to-image diffusion models," 2023.