Analysis and experiments of deepfake creation methods in the geospatial domain and design of an adapted and original detection method

Mémoire

Valentin Meo

Sous la direction de:

Thierry Badard, Research Director

Abstract

Due to their inherent threat, deepfakes have emerged as a societal concern. However, no substantial study has been conducted on deepfakes applied to the geospatial domain. This work begins by providing a comprehensive review of generative technologies enabling the partial modification of images. Leveraging these techniques, high-quality deepfakes of aerial imagery have been created. Conventional falsification detection techniques were employed to assess their robustness. While these methods proved useful, they were not sufficient. Consequently, an original detection method was proposed tailored to the unique characteristics of geospatial images. The highly satisfactory results obtained with this method demonstrate that information control is not the sole solution to the issue of misinformation. This work could be valuable for a broad audience, ranging from intelligence agencies and journalists to concerned citizens aiming to detect falsifications from various entities.

Contents

List of Figures

Introduction

0.1 Contextualization

Misinformation, commonly known as "fake news," has existed throughout history. However, in recent years, with the growing influence of social media and easy access to mass information, the issue of fake news has become a societal concern. Its significant impact is evident during elections, health crises, and international conflicts. Governments, including Russia, China, and France, have implemented laws to ban fake information and punish those responsible, raising questions about freedom of expression and the state's monopoly on truth.

The detrimental effects of fake news are undeniable, particularly regarding conspiracy theories, opinion manipulation by foreign countries, and opinion polarization. However, information verification, historically, has not been a significant problem. For instance, digitally manipulated photos using software like Photoshop can be easily detected. Moreover, logistical challenges arise when dealing with large quantities of data.

Additionally, the emergence of artificial intelligence has given rise to new forms of information manipulation, such as deepfakes. These involve creating or modifying images or videos using specifically trained algorithms. The most common form of deepfake involves substituting one person's likeness with another, such as a U.S. president. Although this field is relatively new, its convincing results suggest that distinguishing truth from falsehood may become increasingly difficult. Detecting these deepfakes is crucial to prevent what some call the "post-truth era" [20]. Detection models exist but are imperfect, and deepfakes continue to improve. Therefore, ongoing development of new tools is essential to combat these information manipulations.

On another front, maps have always played a practical and critical role. Falsification of maps has also existed for as long as they have. In contemporary times, maps and aerial images have never been more accessible and ubiquitous. Their roles, ranging from sovereign and strategic to practical, political, and even ideological, make them prime targets for falsification. For instance, BuzzFeed's investigation, which received a Pulitzer Prize, used aerial images to expose the existence of concentration camps in China despite the government's attempt to conceal them [21]. Cases in Israel and South Korea involve restrictions on detailed aerial images to ensure security. Propagandistic use of satellite images, particularly in communist countries, serves to mask genocides or undermine trust in institutions through fake news. More recently, the war in Ukraine has highlighted the significance of satellite images for communication, investigation, and strategic intelligence.

The production of fake news using artificial intelligence techniques in the field of geographic imagery will be referred to as "deepfake geography." Producing and detecting this type of content poses significant challenges due to the intrinsic characteristics of maps and

aerial images. However, geographic deepfakes, limited in their contextual diversity, require specific methods and models.

0.2 State of the Art

0.2.1 Current Research Context

There are existing studies on false geographical data, but they are not explicitly linked to artificial intelligence techniques or aerial images. For Example, research has been conducted on detecting false geolocations, using data processing methods to identify suspicious cases [22].

These approaches differ from deep fake techniques, which use automated and deep approaches specifically to generate false information from visual media such as photographs or videos. Deepfake methods primarily rely on Generative Adversarial Networks (GANs) to alter or create images [23].

It is not new for GANs to be applied in aerial imagery or cartography. For instance, researchers have proposed training a GAN to create new maps, resulting in intentionally more artistic than practical outcomes [24]. Since 2017, GANs have been employed for road detection in aerial imagery [25] and subsequently used to segment various objects from clouds to buildings, yielding promising results [26][27][28]. GANs are also common in super-resolution problems, suggesting the feasibility of generating fake objects.

"Inpainting" is a process where damaged, deteriorated, missing, or camouflaged parts are filled in to present a complete image. Specialized inpainting methods exist in the geospatial domain but are designed to address image damage. These methods utilize temporal data to aid image reconstruction and are tailored to handle limited damage regarding missing data quantity and various causes. These highly specialized treatments and minimally rely on machine learning techniques [29].

In recent years, some studies have raised concerns even with conventional methods. For Example, an author in a recent article [30] questioned the impact of publishing aerial images of Central Park featuring fake flames. These inquiries have called into question the reliability of geographic information. Debates in recent years have focused on the potential impact of deepfakes on societies. While GANs contribute to advancing knowledge, they can also be used maliciously, requiring epistemological reflection in various fields [31].

These concerns are not limited to researchers. Even high-ranking defense officials have started discussing a "battle for truth" in the digital realm. In 2019, Andrew Hallman, who leads the CIA's digital department, stated, "We are engaged in an existential battle for truth in the digital domain" [32]. Indeed, the production of fake news using artificial intelligence in the context of geographic imagery raises significant issues. Not only can this false information

negatively impact the reliability of geographic information, but it can also be used for malicious activities, especially in defense and security. Unfortunately, military research on this subject is not publicly disclosed for obvious security reasons.

That why it is crucial to establish appropriate methods and models to detect geographic deepfakes and protect the integrity of geographic information. Moreover, raising awareness of this phenomenon among researchers and defense officials is a crucial first step in addressing this threat.

0.2.2 Geographic Deepfakes

Despite the growing interest in GANs and the context of modern science, only one public research study has been conducted to date on the exploration of deep learning approaches for deceptive modification and detection of fake geospatial images.

This research, conducted by Bo Zhao and colleagues at the University of Washington in Seattle in April 2021 [1], aimed to apply deepfake methods to geospatial imagery data. The results demonstrated the potential of these technologies to generate realistic images (Figure 1). The team used a Cycle GAN to generate aerial images from urban structure maps (Figure 2).

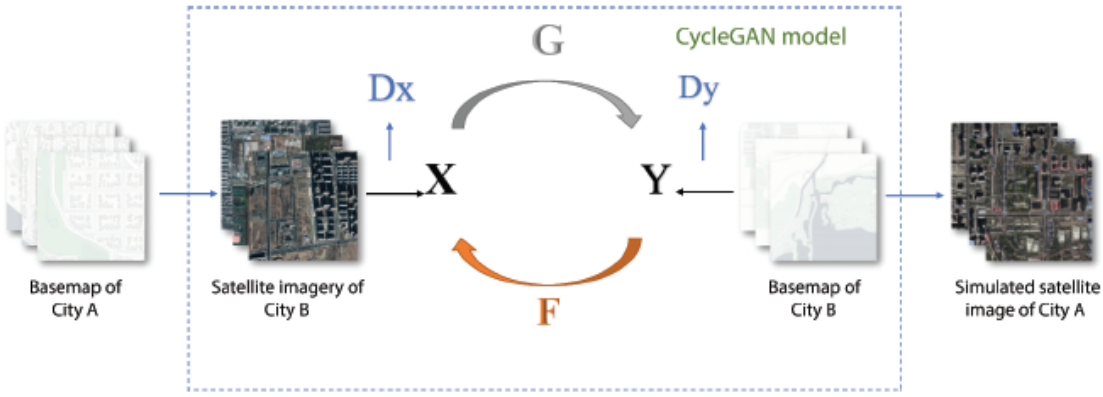
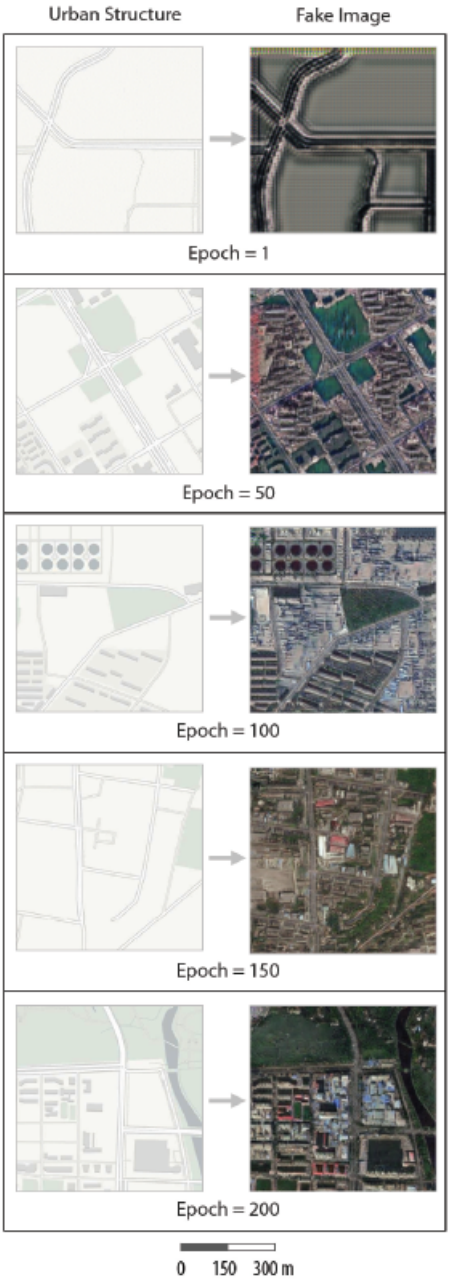


Figure 2: Cycle GAN Schema[1]

After training the algorithm on various cities, fictitious aerial images of Tacoma were generated. The team then implemented a method to distinguish real images from fake ones using features such as spectrum and contrast. These features were processed by a Support Vector Machine (SVM) to classify the images. The model achieved an F1 score and a Recall of 95% for fake images under pressure.

However, it is essential to qualify this result. Firstly, the generated images do not consider the surrounding context of the modified area. Furthermore, the article tests only

Figure 1: Generated fake image[1]



one GAN (CycleGAN), while numerous others exist and can provide better results. The algorithmic concept can be found in the founding article of CycleGAN, which compares several existing algorithms (Figure 3)[2] and Pix2Pix seeks to deliver a better qualitative result.

As for the detection method, it also presents possibility of improvement. Firstly, it compares images created by an algorithm trained on aerial images taken on the other side of the world with local data. The algorithm is trained on aerial images of Beijing to generate

fake images of Tacoma, which are then compared to real images of Tacoma. Consequently, the satellite used may differ, the season may be different, the time of day may vary, and the architecture and urbanization may differ, leading to different color ranges. Therefore, obtaining a hyperplane with excellent results is not surprising. The detection algorithm is also elementary, testing more recent algorithms could yield better results. Finally, the detection result is binary, which is impractical, as the model cannot determine if a specific object is fake, delineate a potentially fake area, or assign a probability.

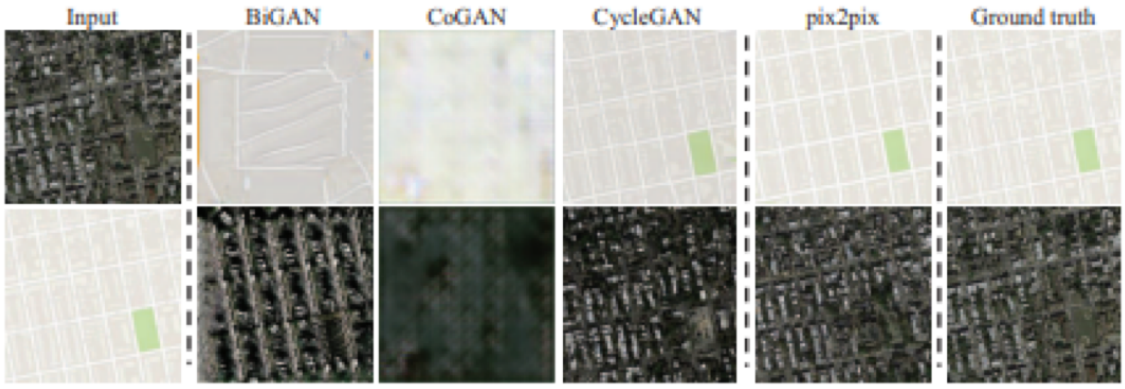


Figure 3: Quick test of other algorithms[2]

While this article is informative and innovative, it has some shortcomings. The author encourages readers to delve deeper into the subject and conduct more comprehensive and less sensational tests. For instance, it would be interesting to explore other types of GANs, attempt to integrate or conceal buildings in an unfalsified environment and use a more rigorous detection method that corrects its shortcomings. However, credit must be given to Zhao for his pioneering approach, laying the foundation for a new public research domain. His work has sparked numerous questions and stimulated curiosity in exploration.

0.2.3 Adapted Creation Methods

GAN

To grasp the concept of deepfakes geography fully, it is imperative to understand the most popular and advanced algorithm employed in deepfake approaches, namely the Generative Adversarial Network (GAN). This section will present the class of GAN algorithms and the nuances that characterize them.

But before delving into that, it is necessary to begin by comprehending in detail the structure and functioning of artificial neural networks, which are a crucial element of this algorithmic class. Neural networks are structures consisting of multiple layers of interconnected neurons. Each neuron is a computing unit performing a linear combination of input values,

weighted by weights, with the addition of a bias, followed by applying a non-linear activation function to produce an output. Learning occurs by iteratively adjusting the network's weights and biases using optimization algorithms, such as gradient descent, to minimize a cost function.

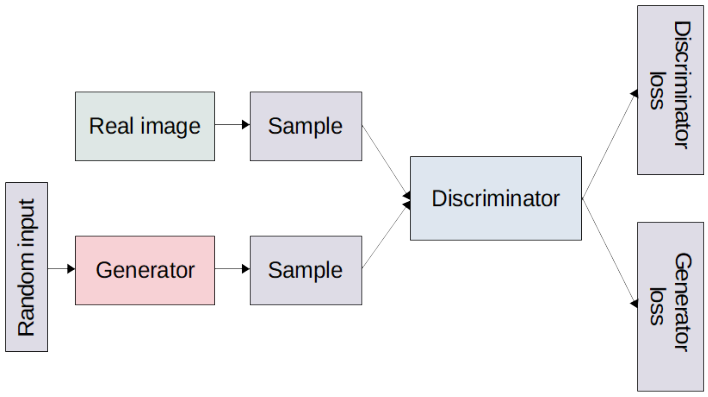


Figure 4: GAN Schema

GAN is a deep learning algorithm composed of two neural networks: a generator network and a discriminator network (Figure 4). During multi-epoch training, these two networks engage in antagonistic competition. The generator aims to deceive the discriminator by producing increasingly realistic images, while the discriminator seeks to distinguish real images from fake ones.

Throughout training, the generator improves by generating images that increasingly deceive the discriminator, which, in turn, improves to better distinguish real images from generated ones[33]. As the discriminator is a classification network, any image classification network can be employed for this task. The deepfake is generated from a random image with 100% noise to obtain a generative result. Although the generator is evaluated based on supervised criteria, the training can be considered unsupervised as it involves antagonistic training.

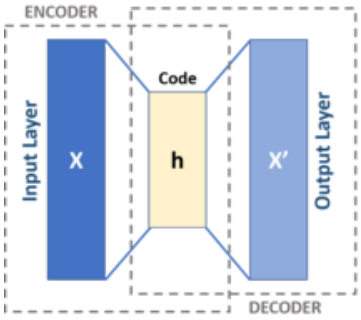
Thus, the challenge of training arises from the often delicate balance between the two actors in this training game. Due to the nature of training, GANs often struggle to converge, which is an advantage for generative tasks but complicates training. Two main learning failure situations must be distinguished: "Vanishing Gradient" and "Mode Collapse."

- "Vanishing Gradient" may occur if the generator fails to create coherent deepfakes because the discriminator is too good. An optimal discriminator does not provide information for the generator to improve, potentially leading to a gradient approaching zero, rendering training ineffective.

- "Mode Collapse" may also occur if the generator learns to generate a single particularly plausible output. Normally, the discriminator would start rejecting everything. However, if it is stuck in a local minimum, the generator will be stuck on a small set of outputs, resulting in a loss of diversity in the generated outputs.

Additionally, GANs pose another problem. While classical GANs effectively create images from one or more references, they do not consider the environment in which the falsified data must be integrated. This is the case with the CycleGAN used in Zhao's article, for Example, and most GAN models such as Pix2pix or StyleGAN. For a deepfake to be credible and applicable in a practical context, seamlessly blending it into a real image is essential. To address this issue, improvements can be made to the classical algorithm using an autoencoder combined with convolutional layers.

Figure 5: Autoencoder Schema[3]



An autoencoder is a type of neural network initially used for encoding but is now employed in various applications such as facial recognition, data generation, classification, anomaly detection, and more. They are traditionally unsupervised and have an architecture that can be simplified as follows (Figure 5): one network encodes the data by reducing it to a lower-dimensional space, and another network is trained from this reduced data to reconstruct the original image. Thus, the network is encouraged to learn the most essential feature representations in the latent space. Then, a loss function is used to measure the error and minimize the difference by optimizing the network. The choice of the loss function depends on the context.

However, autoencoders do not consider the environment in which the data must be inserted and are not well-suited for generating images, which is problematic for creating credible deepfakes. Certain specificities can be added to the classical algorithm to remedy this issue by associating the autoencoder with convolutional layers or CNN (Convolutional Neural Networks). Unlike autoencoders and their fully connected layer, CNNs redefine the image with local kernel matrix operations. The significant reduction in the number of weights to be calculated allows the autoencoder architecture to be applied to images much more efficiently.

Thus, by changing the reconstruction criteria and thanks to these particular neural layers, an autoencoder can learn to reconstruct a sharp, noise-free, and undamaged image. Multiple encoders/decoders can be stacked to enable the network to learn abstract features more easily.

Combined with a GAN, an autoencoder with convolutions layers will complete the image by creating a fake area in the missing spaces of the image, and the evaluator will define whether the modified area is true or false.

Therefore, this GAN approach will be used. Although many variants and approaches exist, a list of the most common and relevant GAN approaches for inpainting tasks will be presented. This will take into account the context of an image.

In the rest of the document, inpainting deepfakes geography are defined as inpainting deepfakes of visible geospatial images. In parallel, classic deepfakes or whole-image deepfakes are derived from models that produce a single image without context.

Context Encoder

This method, called Context Encoder, is the first to use a GAN structure to generate images considering their context. It was invented in 2016 at the University of California, Berkeley[4]. The generator used in this method is an autoencoder trained to predict missing parts of an image. Conversely, the discriminator is trained in competition with the generator (Figure 6).

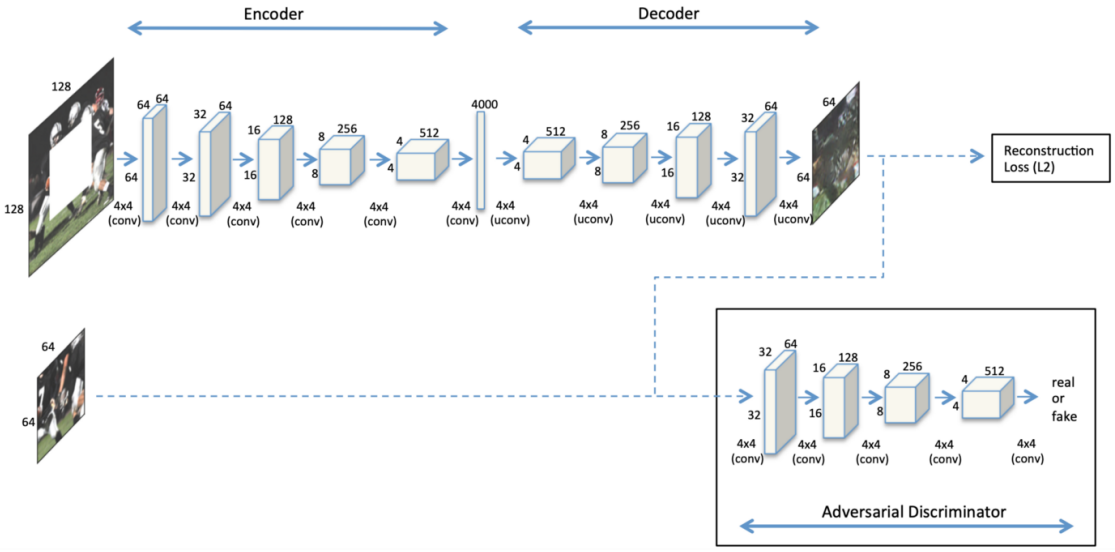


Figure 6: Context Encoder[4]

As initially proposed, the autoencoder follows an AlexNet structure[34], but transforms the convolution layers into fully connected layers like a classical autoencoder. One of the

drawbacks of context encoders is that the output is often blurred, meaning the model predicts uniform colors. Further details on loss functions can be found later in this document.

Globally and Locally Consistent Image Completion (GL-GAN)

The GL-GAN network was first introduced in 2017[5]. It is a convolutional network followed by dilated convolution, capable of taking input images of various sizes and resolutions. One distinctive feature of this network is the use of two discriminators: a global discriminator examining the entire image and a local discriminator focusing on the false parts of the image (Figure 7). A simple improvement involves putting two GL-GAN networks in series [18], which has shown superior results. Additionally, "contextual attention" in the discriminators has also enhanced results.

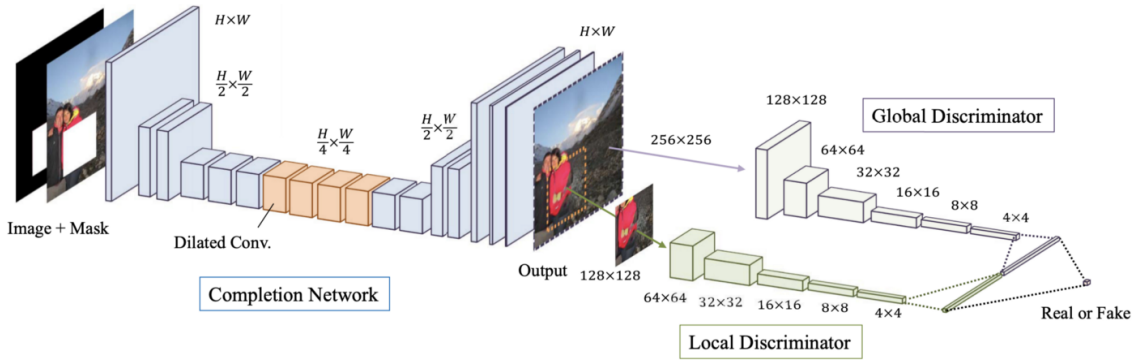


Figure 7: GL-GAN[5]

GL-GAN is the only algorithm tested for deep inpainting of aerial images[6]. However, the conducted tests were basic, using very low-resolution images, yielding mediocre results (Figure 8). These results are significantly lower than those typically observed in the literature for GL-GAN applied to facial datasets. The lack of details highlights a genuine problem in the methodology of this experiment.

Figure 8: Results obtained in the literature[6]

Patch-Based Methods

The idea behind these methods is to detect similar patterns in the image to those present on the edge of the area to be completed[35], using the nearest-neighbor algorithm. These patterns are then used to fill in the edges of the image. While these approaches may be practical for relatively homogeneous aerial images, such as a forest, they are ineffective in complex environments without repetitive patterns, such as cities. It is worth noting that these methods are not based on a GAN architecture, and their implementation is outdated. Moreover, they are highly resource-efficient, making them potentially useful as a baseline to beat.

Another commonly used baseline is the image completion implementation in Photoshop, which can also be explored. Although this software is proprietary, the approach used is likely to be of the "patch" type. However, before these methods, inpainting techniques were limited. Some pioneering approaches are described in different works[36][37][19], mainly based on diffusion.

Patch-Based Image Inpainting with Generative Adversarial Networks (PG-GAN)

This is a revision[7] of the previous method: GL-GAN. The idea is to replace the dilated convolution of the generator network with a network using only residual connections, namely the ResNet algorithm[38]. As for the discriminator network, it is suggested to use a patch-based discriminator instead of a local discriminator, along with a G-GAN discriminator for the global discriminator (Figure 9). Both the convolutional and residual layers can be tuned to refine the results.

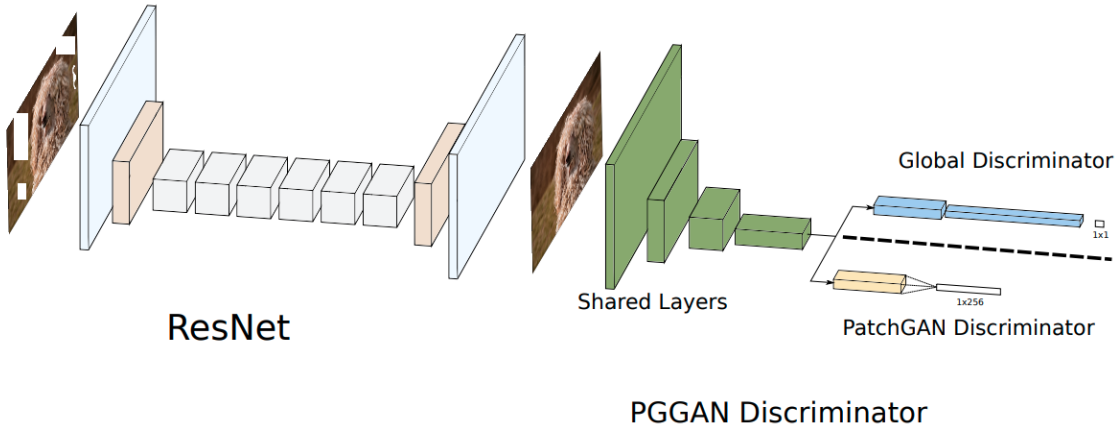


Figure 9: PG-GAN Schema[7]

EdgeConnect

There is a hybridization between GL-GAN and PG-GAN, named EdgeConnect[8]. This implementation consists of two networks in series. The first network is trained to generate a simplified image version, allowing conditioning by adding a sketch. The second network takes the result of the first network and the image to be completed as input and completes the work. The generator network comprises dilated convolution and residual layers (Figure 10). The discriminator is of the SN-PatchGAN type, meaning it is a PatchGAN discriminator combined with spectral normalization (SN) (Figure 11). According to the author, conditioning degrades the results by adding more blur.

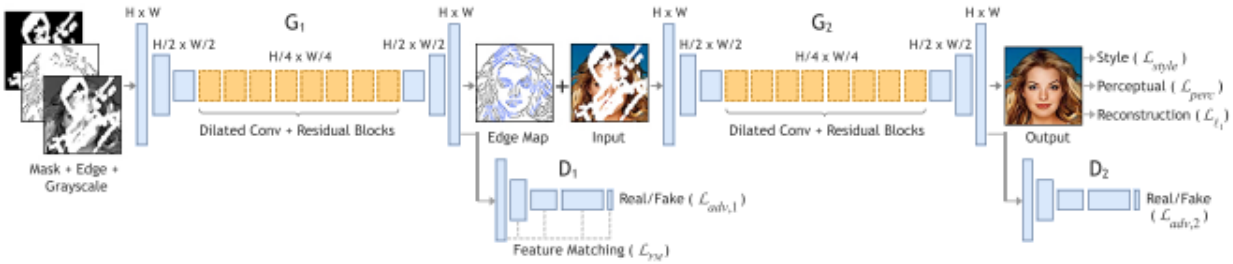


Figure 10: EdgeConnect Algorithm[8]

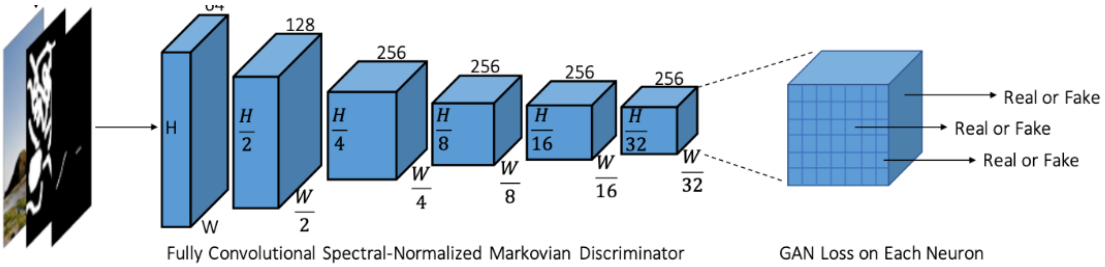


Figure 11: SN-PatchGAN Discriminator

U-Net

U-Net is a widely used network that performs well in computer vision tasks. It combines classic convolution and residual elements while reducing dimensions before reconstructing the image. For Example, this network has been implemented as shown in Figure 12[9]. In the article, a shift module was added to refine the results. A different implementation of the same concept can also be found under the name FPN for Fast Pyramid Network[39].

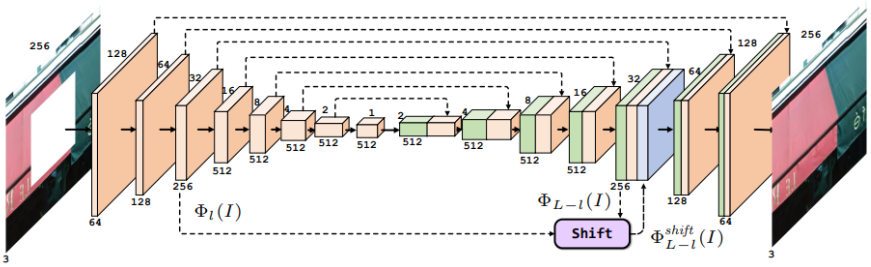


Figure 12: Inpainting Algorithm based on U-Net[9]

Partial Convolution: Addressing Missing Parts

One of the issues with traditional convolutional networks is that they perform convolution on the missing parts of the image as well. To address this problem, Nvidia's research team proposed Partial Convolution[10] (PConv). This method relies on the idea that the initial convolution layers should only use actual input data, making it a suitable convolution method for inpainting. The results obtained with this method are significantly better than those achieved with conventional convolution methods (Figure 13). Additionally, this method can be combined with any convolution algorithm, making it highly flexible.

However, this comprehensive and exemplary research work explores the limits of the proposed method. The results indicate that when the missing area is too substantial on the object to be completed, this area is not considered, limiting the method's effectiveness (Figure 14).

Figure 13: Comparison of different convolution algorithms[10]

Figure 14: Comparison of inpainting based on the relative space occupied by the area to be completed on the total image[10]

An improvement to partial convolution, called Gated Convolution, was proposed for better performance[19]. This method uses a simple convolution followed by a sigmoid function to generate the new partial convolution mask rather than updating it with arbitrary rules. In a classic partial convolution, the mask updates quickly when the convolution matrix result applied to the mask exceeds 1, indicating a suitable pixel. In contrast, the Gated Convolution approach has a smoother transition as the sigmoid produces a value in the [0-1] range.

Common Loss Functions

To assess the quality of deepfakes created during training, several loss functions are commonly used. Those that have garnered particular interest in the literature will be detailed below. It should be noted that the loss function can be composed of many other weighted loss functions.

Let us start by recalling the commonly used L1 and L2 loss functions in computer vision and in a hybrid way:

- The L1 loss function is defined as the mean of absolute differences between each pair of pixels, as shown in Equation (1).

$$loss(x, y) = \frac{1}{n} \sum_{i=1}^{n} |x(i) - y(i)| \tag{1}$$

- The L2 loss function is similar, but it uses the squared difference and then calculates the mean of these differences. To account for multiple color channels (e.g., RGB), an adaptation of these functions is called "pixel-wise loss."

A combination of the advantages of the last two functions is to use a loss function that computes the L1 score when values are distant and the L2 score when values are close. This approach prevents neglecting weak errors and being less sensitive to extreme values.

GANs use an adversarial loss function that depends on the adversarial game, meaning a penalty for the adversary's success. Thus, a commonly used loss function in inpainting consists of an L2 loss applied to the area to be completed, combined with a classic normalized adversarial loss, weighted at 0.9 and 0.1, respectively. A standalone L2 loss leads to the creation of blurry areas.

An interesting idea for inpainting-specific loss functions is to apply a weight to the pixel near the area to be completed. However, few loss functions are genuinely adapted to inpainting, with the previously mentioned functions often preferred.

The most commonly used loss functions during discriminator training are classification losses. However, several slightly more complex loss functions can be used when the discriminator solves a segmentation task. For Example, the DICE loss measures the overlap between the predicted area and the area to be predicted. Weighted cross-entropy loss can also be employed. The Jaccard index, also known as the intersection over union (IoU) score, is the most commonly used function in segmentation tasks (Formula 1.3). Indeed, the total predicted area is considered in the score. This score ranges from 0 to 1, with 1 being the best possible score and 0 being the worst. To make this function differentiable, the classical L2 method is often used.

$$Jaccard\ Index = \frac{Overlap\ Area}{Union\ Area} \tag{2}$$

Beyond GANs

Another particularly effective method has emerged with similar results for deepfake tasks and surpasses them in cases of parametrization with text. This is the diffusion method[40]. However, it should not be confused with the previously discussed diffusion methods. This approach does not rely on adversarial learning but learns to reconstruct a noisy image during training. Then, for the application, a 100% noisy image is given as input. This approach has recently been adapted to inpainting. Empirical results are inferior to what GANs pro-

duce in unguided textless deepfake creation. Adversarial training methods with a diffusion architecture are not yet refined.

0.2.4 Fake Image Detection

GANs are potent models for generating fake images. Therefore, detecting "manipulated" images is imperative to avoid falling into a dystopia where it is impossible to believe what one sees. This section will present the main current approaches to detect modified images.

FotoForensics

FotoForensics Methods

FotoForensics methods have often inspired the development of more advanced techniques. FotoForensics is a non-deep analysis technique for digital images to detect and identify potential manipulations, alterations, or forgeries. Most of these methods were developed before deepfakes emerged and remain among the most commonly used. Standard methods include ELA, PCA, and JPEG Ghost. They aim to identify areas from different sources, making them suitable for inpainting detection. ELA and JPEG Ghost achieve this by assessing the difference from the same compressed image, while PCA uses dimension reduction to highlight features in areas from different sources.

Feature Standard (Feature Engineering)

The specific method presented here is the oldest, founded on the hypothesis that each GAN is unique. Depending on the model's training, it leaves a specific footprint that can be detected[41]. This footprint can take various forms, such as a unique color range. The approach involves studying the behavior of a specific GAN to generate features adapted to its detection or manually creating multiple features to attempt detection, considering the common flaws of GANs. In 2014, for instance, 14 features were defined as elementary to determine the authenticity of a fake[42]. With the help of these features, supervised algorithms can be trained.

However, this approach is highly specific and dependent on the type of GAN used. It yields mediocre results when applied to an image modified by a GAN not used to generate the training set or on newer models. It is preferable to detect deepfakes without knowing the generating algorithm. Determining features may vary for each case, making an approach effective for one algorithm but not for another. Additionally, since the flaws of a GAN are known, it is possible to create a tailored loss function to control better the weaknesses that facilitated detection.

Figure 15: Features in Zhao's article[1]

Table 1. Features of authentic and fake satellite images.

Code	Feature description
Spatial	
CFI	Image Colorfulness Index: A larger value indicates a more colorful image
BIQ	Brenne Image Quality Index: A larger value indicates a clearer image
TIQ	Tenengrad Image Quality Index: A larger value indicates a clearer image
LIQ	Laplacian Image Quality Index: A larger value indicates a clearer image
ASM	Angular Second Moment of GLCM: A larger value indicates a more uniform and regularly changing texture pattern
CON	Contrast of GLCM: The greater the CON, the deeper the grooves of the texture, and the clearer the visual effect
ENT	Entropy of GLCM: The more complex and uneven the texture in the image, the greater the ENT value
IDM	Inverse Different Moment of GLCM: The larger the IDM, the smaller the change between areas of the image texture, or the local pattern is more uniform
Frequency	
FASM	ASM at Frequency Domain: Similar to ASM
FCON	CON at Frequency Domain: Similar to CON
FENT	ENT at Frequency Domain: Similar to ENT
FIDM	IDM at Frequency Domain: Similar to IDM
Histogram	
MEAN	Mean of GLH, The larger the value the brighter the image
STD	Standard Deviation of GLH, the larger the value less concentrated the GLH
SKEW	Skewness of GLH, the larger the value more skewed the GLH
KURT	Kurtosis of GLH, the larger the value the steeper the GLH
GET	Entropy of GLH, the larger the value, the less even the GLH
CM1_R/G/B	First Order Color Moment of Red (Green/Blue): mean of color histogram
CM2_R/G/B	Second Order Color Moment of Red (Green/Blue): variance of color histogram
CM3_R/G/B	Third Order Color Moment of Red (Green/Blue): skewness of color histogram

GLCM means gray level concurrence matrix; GLH refers to gray level histogram.

Zhao experimented with this approach, creating 20 features (Figure 15). He then eliminated non-discriminatory features and trained a supervised model with the remaining ones. Although this improves the model's performance on specific data, it makes the model less suitable for other data from different models, reducing its generalization ability by overfitting specific data.

Zhao used a simple model, but deeper implementations are presented below.

Fourier Spectrum

This method is based on the observation that current GANs cannot generate a perfect Fourier spectrum (Figure 16). Oversampling can lead to peaks in the spectrum[43]. Ap-

proaches use simple classifiers to discriminate between real and fake images based on the characteristics of this space.

This method has shown promising results in the literature, as evidenced by the works of references [44][45][11]. It can even be adapted to detect specific objects within an unaltered image. However, it heavily depends on the GAN used. To counter this detection method, specialized loss functions in the spectral domain can be implemented during GAN training. This technique reduces the effectiveness of the detection method, as demonstrated[46]. A deep implementation of this approach[11] has been explored, where a ResNet34 takes the Fourier spectrum as input.

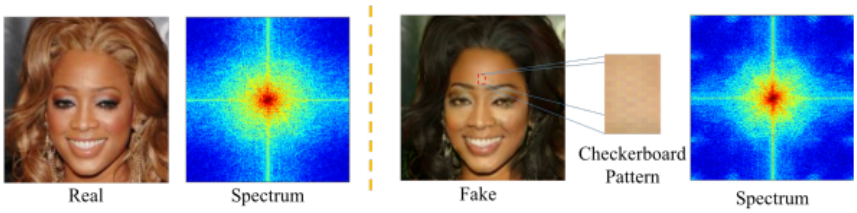


Figure 16: Comparison of Fourier spectrum from a real and fake image[11]

In 2020, a model attempted to combine several different frequency representations obtained using the Fourier spectrum with excellent results. In 2021, this model achieved the best results on FaceForensics++, with over 90% accuracy. It includes three frequency-sensitive indices: Frequency-aware Decomposition (FAD), Local Frequency Statistics, and standard input. This implementation, named F3Net, yields good results[47].

InceptionNet

Previous approaches have been criticized for seeking performance improvement through increased depth, leading to overfitting and high computational costs. Additionally, the choice of convolution kernel size depends on the object's size in the image, which is effective for faces but not for objects with varying relative sizes. To address these issues, InceptionNet[13] proposes using different convolution kernel sizes in parallel and concatenating them (Figure 17). Several implementations exist, utilizing Inception layers. InceptionNet V4[12] is a successful example (Figure 18).

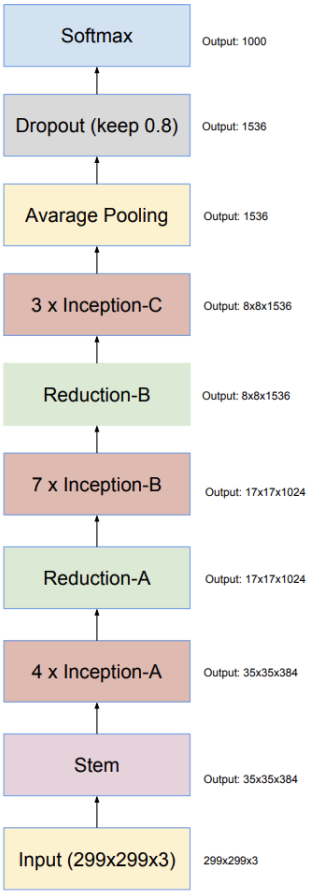


Figure 17: InceptionNet V4 Network Architecture[12]

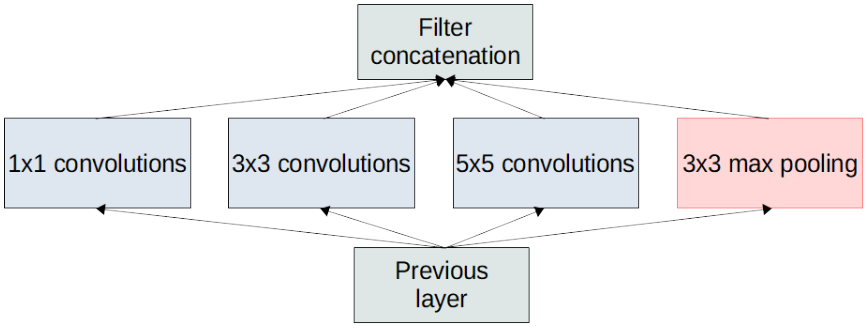


Figure 18: Inception Layer Schema[13]

XceptionNet

XceptionNet is a modification of InceptionNetV3. Its main distinction lies in the addition of a pointwise convolution (convolution with a 1x1 kernel) and a depthwise convolution in the conventional Inception layer (Figure19). The incorporation of residual convolutions

significantly enhances the results[48], contributing to an overall improved network (Figure 20)[14]. Several efficient variants of this network, such as MobileNet[49], have been developed. The previously discussed F3Net model utilizes an Xception-like architecture. In the literature, these networks have demonstrated superior performance to Inception networks.

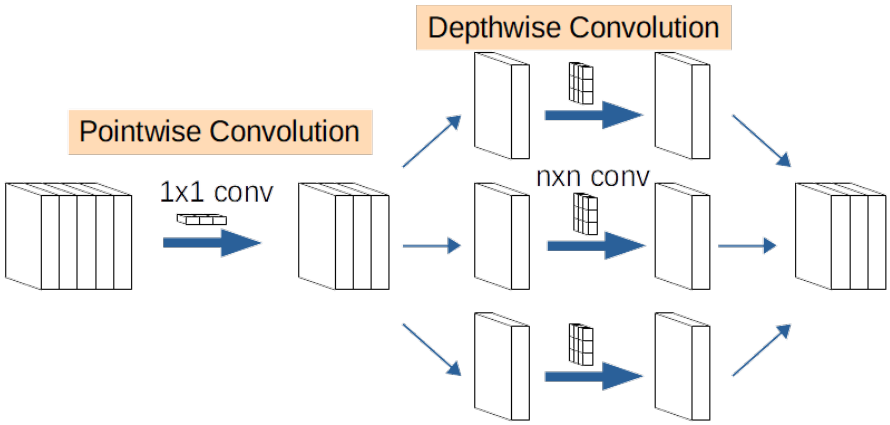


Figure 19: Xception Layer Schema

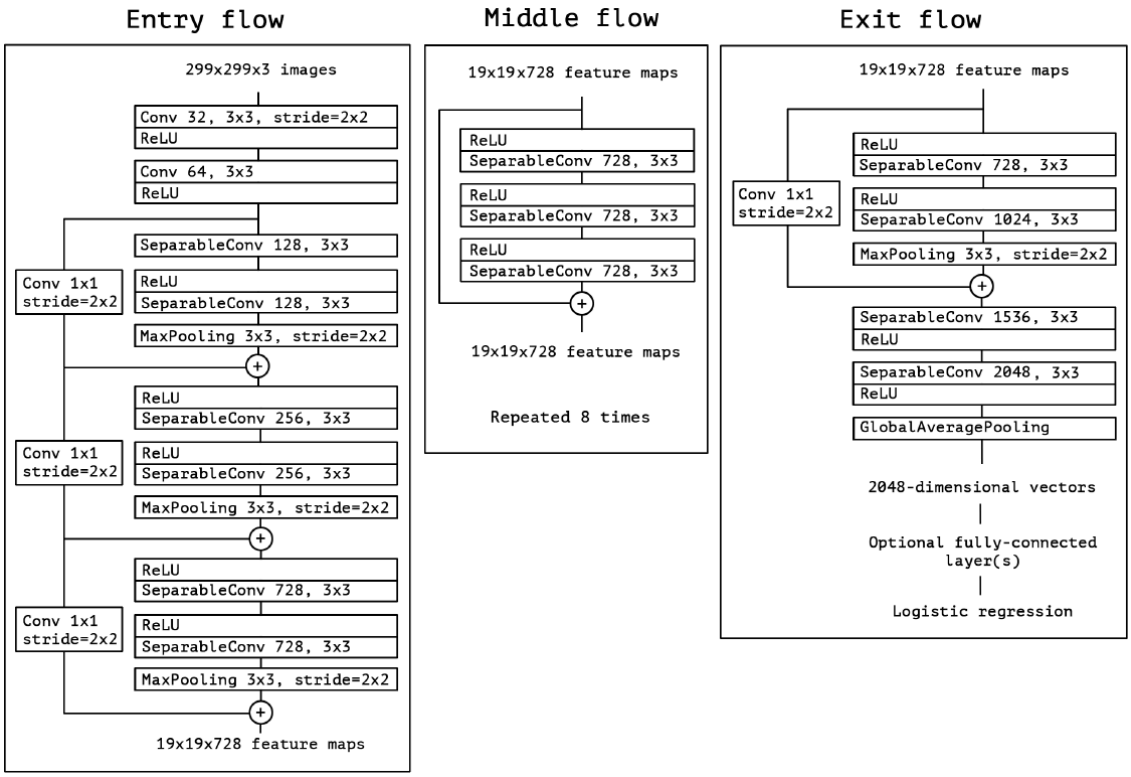


Figure 20: XceptionNet Architecture[14]

EfficientNet

EfficientNet is a convolutional neural network architecture[50] designed to address the limitations of previous architectures regarding complexity, performance, and training efficiency. It combines convolutional blocks with different sizes and variable depths to create a more efficient and versatile architecture. The model uses an automatic search method for optimal architecture based on input image size, network width, and network depth, optimizing resource usage while maximizing performance. This approach has resulted in a model outperforming previous networks while maintaining lower complexity.

However, these approaches represent general deepfake detection methods where the input image is categorized as binary. This does not align with the goal of this research. Therefore, a local adaptation of these methods is necessary for inpainting detection. To achieve this, image division and localized analysis could be considered to tailor these algorithms. A list of inpainting-specific methods can be found later in this document.

0.2.5 Detection of Fake Areas in Real Images

Very few studies have investigated deepfake detection in a real-world context, specifically deepfakes with inpainting. Available studies suggest that architectures with good results in object segmentation are effective for this task. However, limited testing has been conducted due to the lack of databases for deepfakes with inpainting. This section will present some popular object segmentation methods, along with recent inpainting deepfake detection approaches that have shown promising results in the literature.

CNN

A simple approach based on a Convolutional Neural Network (CNN) used for object detection is generally referred to as R-CNN (Regional CNN) (Figure 21)[15]. One of the latest and most advanced versions of this method is called Mask R-CNN, with the main difference being the algorithm's ability to perform instance segmentation rather than just semantic segmentation. Initially designed for segmentation, it has also been used for deepfake detection with good results. The advantage of this architecture is its ease of hybridization with other methods to improve deepfake detection performance.

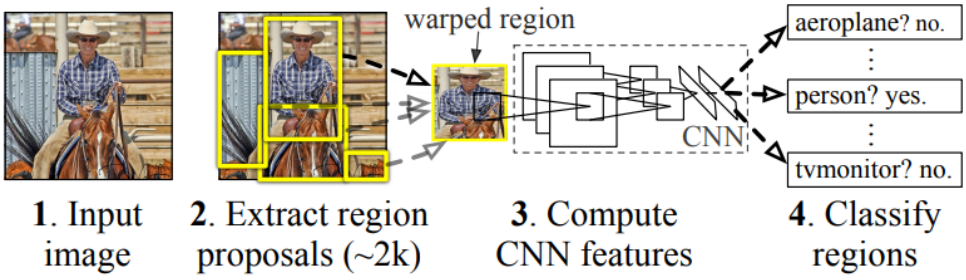


Figure 21: R-CNN[15]

IID-NET

This is a method specifically designed for inpainting detection[16]. A scheme of the method is presented in Figure 22. It consists of 3 blocks: the enhancement block, the extraction block, and the decision block. Specifically, the enhancement block aims to improve inpainting traces by using parallel input layers before combining them. The Neural Architecture Search (NAS) algorithm automatically designs the extraction block to extract features. Global and local attention modules are present in the decision block to optimize the extracted latent features further. This method is expected to outperform other approaches considered classical in inpainting detection. Another advantage of this method is its reduction of false positives compared to existing literature.

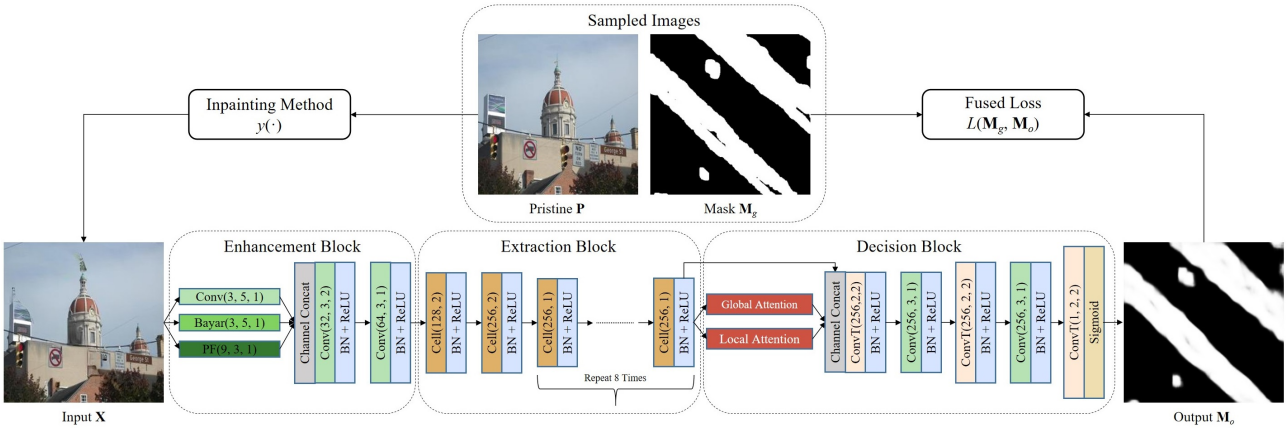


Figure 22: IID-NET Scheme[16]

MantraNet

MantraNet[17] has also shown good results in the literature. Its architecture can be found in Figure 23.

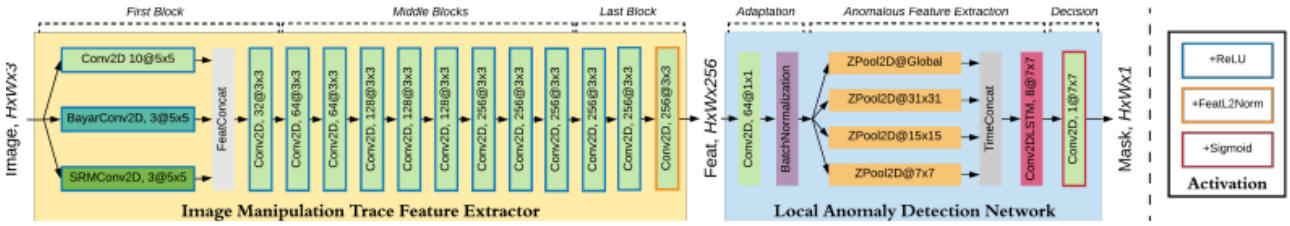


Figure 23: MantraNet Scheme[17]

Final Discussion on the State of the Art

The current state of the art in deepfake methods has been adapted for non-geospatial images. Classical machine learning benchmarks do not include any aerial images. Models in the literature are over-optimized for the production of realistic human faces. This affects input layer sizes, convolution matrix sizes, loss functions, and newly created training sets. Thus, none of the models in this state-of-the-art have ever been trained with geospatial data. For Example, OpenAI's generative model DALL-E cannot create credible geospatial images despite being one of the best image-generating AIs today. Figure 24 illustrates this paradox: a human face on the left and a geospatial image on the right. The created face is perfect in this figure, while the geospatial image is unrealistic, with poorly formed buildings resembling a cluster of shapes and colors.

Figure 24: Images produced by DALL-E

0.3 Issue

The use of deepfake models adapted to aerial photographs could generate highly impactful fake news. Their utilization by malicious individuals could significantly disrupt public discourse, thereby undermining the proper functioning of our democracies. Their use by governments may lead to authoritarianism, either through the state's intentional creation of fake news or citizens' fear in response to such fake news, which could then demand increased state

intervention for protection, an undesirable outcome. Finally, in the context of war, their use could result in poor tactical and strategic choices on the ground. Unlike false maps, producing false geospatial images can become much more viral and dangerous since an aerial photograph represents reality more faithfully than an abstract map.

However, the majority of deepfake research, whether for creation or detection, focuses on generating entire images. Given aerial images' large and contextual size, it would be more relevant to modify significant parts of real images, as geospatial images taken out of context have little value. To achieve this, "inpainting" methods are the best choices. However, the state of the art shows a lack of research for geospatial "inpainting". Additionally, classical deepfake detection methods are binary, limited to detecting falsified images as a whole. Furthermore, research on deepfake detection through "inpainting" is very underdeveloped.

Moreover, aerial images have unique characteristics that conventional images lack. Classical deepfake models are only trained on non-aerial images, and their structure and functioning are primarily optimized for face datasets. Elements in aerial images have very different shapes and color ranges. For Example, geospatial images also capture artificial structures such as roads, bridges, buildings, ports, airports, and industrial facilities. The representation of these elements is specific to this type of image. Moreover, these elements have spatial relationships vastly different from those of a face; the proximity of a residence to a road is an example. Their size is also a significant challenge. Classical models are often optimized for small images, a few hundred pixels. Aerial images of high precision have sizes of several thousand pixels, making it challenging due to the number of parameters to learn and the size of the fake area to produce. Therefore, the depth of models from the literature would need adaptation to these large images. The loss functions would also need modification, and a training benchmark would need to be established.

Therefore, it is crucial for comprehensive and rigorous scientific work to explore the possibilities of current technologies in the specific case of geospatial deepfakes and, most importantly, attempt to implement an open-source, efficient, and practical model for their detection. This work must be done in the public domain as early as possible before malicious individuals or governments misuse it.

The research issue is thus: currently, there are no in-depth studies to determine if the latest AI-based "inpainting" techniques are capable of generating credible geospatial deepfakes, and there is no suitable method for detecting them.

Therefore, it is essential to attempt to answer the following question:

Can geospatial "inpainting" deepfakes produced with the latest models be detected?

0.4 Hypotheses

This work is primarily exploratory, but the following hypotheses can nonetheless be posited:

The application of deep learning techniques to geospatial images enables the creation of high-quality geospatial deepfakes. Deep "inpainting" models, initially designed for other domains, can be successfully adapted to generate convincing geospatial deepfakes. Moreover, deepfake detection methods can be adapted, or a new detection method can be created to detect these geospatial deepfakes.

0.5 Objectives

The main objective of the research is the development of a geospatial deepfake detection method. Achieving this goal involves first achieving the following secondary objective: exploring and analyzing "inpainting" methods to produce convincing geospatial deepfakes, considering the spatial elements and relationships they exhibit in reality. The achievement of this sub-objective will be the focus of Chapter 1, while the achievement of the main objective will be addressed in Chapter 2.

0.6 Methodology

This research is primarily exploratory. To date, only one serious study has addressed the subject of deepfakes in the context of aerial images. Figure 25 outlines the different phases of the research.

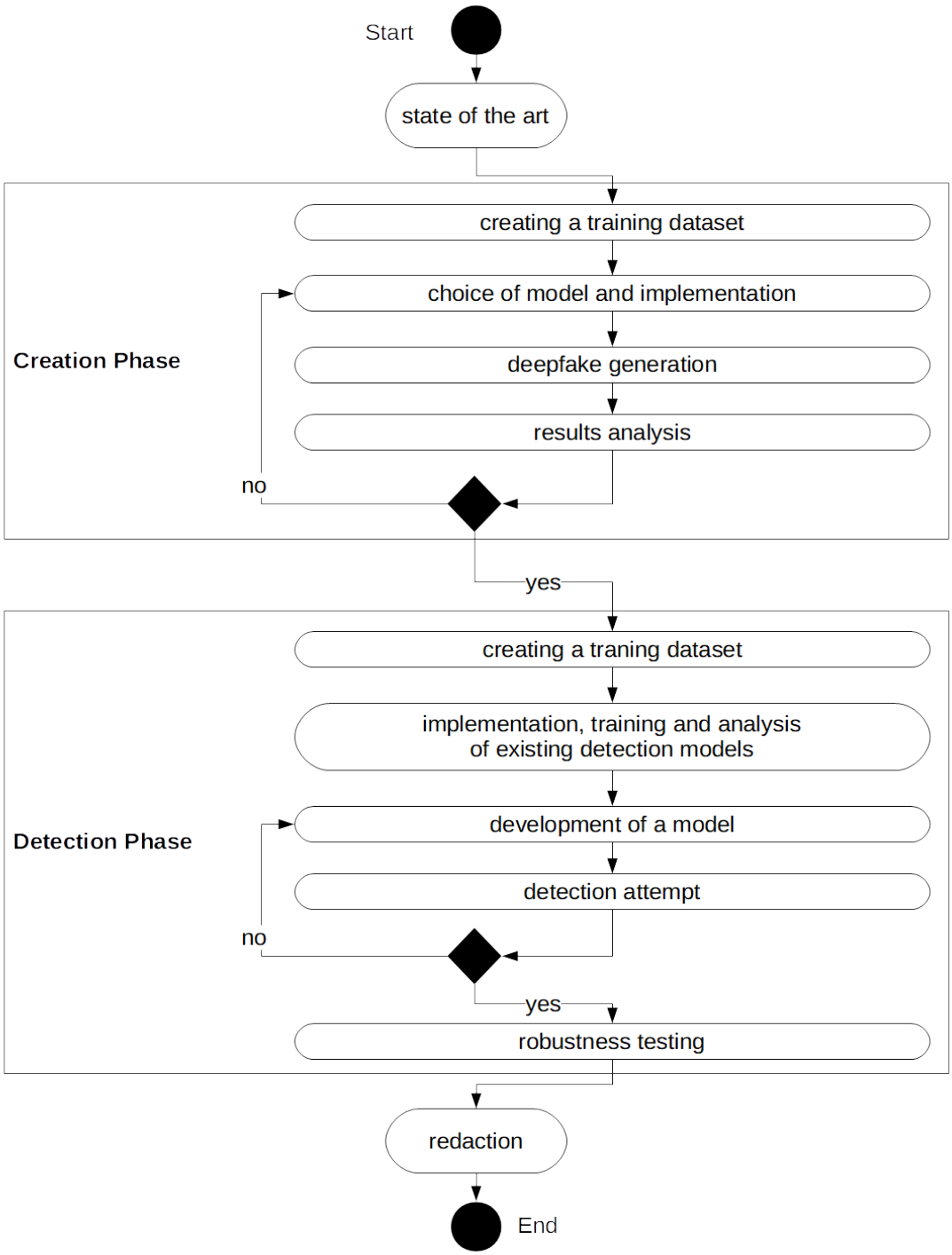


Figure 25: Project Methodology

0.6.1 Creation Phase

The objective of this phase is to generate synthetic geospatial data. In other words, realistic deepfakes will be created using aerial images in various contexts and at different difficulty levels. Deepfakes will be inserted into real images since there is a lack of geographical inpainting deepfakes. A deepfake database will be created using multiple datasets. These datasets will consist of high-resolution aerial images, posing a greater challenge for creating detailed and diverse patterns.

This step involves constructing a dataset for training and evaluating deepfake models. A geographical sampling of Quebec's space will be conducted to maximize diversity and representativeness. The required aerial images will be obtained from the Geoindex source. However, Geoindex's satellite data privacy policy will not make datasets and created deepfakes public. Stratified sampling will then be performed, grouping areas such as urban zones, forests, water bodies, agriculture, and roads. Using strata aims to have datasets with less diverse characteristics, limiting the number of features to learn, model depth, and training time. This simplification of image generation does not diminish the project's scope but enhances the detection challenge. Better fake images make detection more difficult. Moreover, training and evaluation data will be limited to images taken during the same period of the year for the same reasons.

- For the forest stratum, random sampling will be performed, and some images including cabins will be added with concealment attempts. Forests represent a significant portion of the Canadian landscape.
- For the water stratum, random sampling will be performed, and some images, including boats, will be added with an attempt of concealment. Water covers a vast area, and concealing small objects like boats poses an evident threat.
- For the rural stratum, simple random sampling will be performed. Rural areas also represent a significant part of the Canadian landscape.
- For the road stratum, simple random sampling will be performed. Only images containing road-like elements without buildings will be kept. The OpenStreetMap API will be used for this purpose. This choice of a road stratum was made because roads are a strategic and common infrastructure. Additionally, their presence is undersampled in other datasets.
- For the city stratum, simple random sampling will be performed. Only images containing a certain number of buildings will be kept. The open data from the city of Quebec will be used. Urban environments are distinct from other environments.

Some of the approaches mentioned in the state of the art will be tested. The primary generation methods highlighted in the state of the art are ContextEncoder, GL-GAN, Edge-Connect, and partial convolution with SN-GAN. All these methods need to be tested, and the guidance functionality will be utilized with geographic data when possible.

After creating deepfakes using the previously discussed models, an objective evaluation of the performance, limitations, and advantages of each model has been initiated. To support this qualitative analysis, roughness will be calculated as an additional quantitative comparison element. The evaluation of deepfake realism will involve a qualitative analysis of the fake area and a comparison with the best results from the state of the art. Once this phase is satisfactory, the detection phase will be initiated.

0.6.2 Detection Phase

In this phase, an attempt to detect the deepfakes produced in the creation phase will be made. For this, several previously seen methods will be tested. Then, depending on the results of these tests, an original model will be proposed. The evaluation will then take place in three parts.

First, classical deepfake detection methods will be tested. A confusion matrix or associated ratios will be used and compared to scores found in the literature to evaluate image classification as globally false or true. For this, implementation in the Python library Sk-learn will be used.

About inpainting detection classic FotoForensics methods will be explored. Next, classical deepfake detection methods will be tested. Finally, an original detection method will be designed and its robustness put to the test. The development of this method will be based on previous detection results. Its design will depend on characteristic artifacts present in the deepfakes and the best-tested classical detection method. The detection method will take the form of a neural network with supervised training. For evaluation, the IoU score detailed in the state of the art will be used. The Jaccard score has already been implemented in Scikit-learn.

During this phase, test procedures will be carried out on 50% of real images in the same sample as generated images. An attempt to train and detect fake images in a dataset composed of images from all samples will also be made. Ultimately, detection of deepfakes from several different models will be attempted, with the practical case providing few to no fake images and potentially coming from various models for training.

This phase will allow for the comparison of different detection approaches, their performance, relative effectiveness, and respective limitations under different difficulty levels and contexts. The performance of datasets will be compared to those found in the literature. All of this aims to propose a method for detecting inpainting deepfakes geography. Thus, everything

accomplished in this part will contribute to developing a detection method.

The work conducted in this study may draw attention to non-totalitarian solutions for addressing fake news in geography. At the end an article will be published to share the obtained results. Moreover, this specific research field is relatively new and will position Laval University as a pioneer in the field of deepfakes geography. The scope of this study aligns with current societal debates, particularly the concerns of Laval University's AI Observatory, implying that the obtained results must be exemplary and attempt to provide a solution to current social fears.

0.6.3 Tools

This section presents the list of various software and libraries that will be utilized in the course of this research:

- Python: Primary programming language for developing detection models and algorithms. Python's popularity justifies this choice, as it is widely used in the scientific community due to its simplicity and convenience. Additionally, most implementations and APIs mentioned in the state of the art are done in Python, facilitating the integration of these tools into our project.
- TensorFlow: Machine learning development platform for creating deepfake models.
- Keras: Python interface for utilizing TensorFlow and creating deep learning models.
- PyTorch: Open-source deep learning framework.
- Scikit-learn: Library for statistical analysis and image classification.
- OpenCV: Computer vision library for image manipulation and preprocessing operations.
- Geoindex: Source of aerial images for dataset creation.
- OpenStreetMap: Data source for road dataset creation.
- Git: Software used for backup, sharing, and work tracking.

All software and libraries will be used following their respective licenses.

The hardware choice will depend on the available server resources. Computations will be performed on a personal machine equipped with a GTX 1080 GPU with 8 GB of VRAM, an Intel i7-7700k CPU, and 16 GB of RAM, or on the CRDIG server equipped with an Nvidia Titan V GPU, which has 12 GB of dedicated graphics memory.

0.7 Expected Results

For deepfake creation, the goal is to produce realistic deepfakes by concealing artifacts. If artifacts cannot be realistically concealed, adjustments will be made to the algorithms, or alternative algorithms will be employed. Specific GAN models are expected to be more or less suitable depending on the different defined contexts. Since the topic is exploratory, predicting results is challenging, but an example of inpainting in an urban context is presented in Figure 26. The produced deepfakes should be at least as natural as the presented Example.

Figure 26: Example of inpainting output in an urban environment[7]

For binary detection, detecting between 70% and 95% of deepfakes would be a reasonable estimate. The literature has rarely exceeded 95% for realistic deepfakes. The detection percentage is challenging to estimate since commonly used benchmark datasets are often specialized or a mix of various deepfakes, including some of poor quality. Moreover, it depends heavily on the quality of the produced deepfakes. The Figure 26 tends to show that the created deepfakes will not be perfect and easily identifiable. We hope that for each model, a particular detection method will work well, and a hybrid method will give correct results compared to

existing general deepfake detection.

For segmentation methods, an IoU score of 0.6 is generally considered good[51]. However, the literature has achieved better results but on low-quality deepfakes. Moreover, given the nature of the problem, minimizing type 1 errors is a priority. Thus, even though it remains difficult to estimate a score for the above reasons, 0.6 is reasonable.

The creation of a specialized deepfakes geography dataset may also be made public to aid university research, subject to authorization from the GeoStat center. This work should make possible the development of an easily usable and scalable application, the possibilities of which will be expanded in the conclusion.

Chapter 1

Creation of Geographic Deepfakes

1.1 Introduction

This first chapter is dedicated to the creation of geographic deepfakes. This part involves the exploration and analysis of various methods and datasets used. Initially, different datasets will be presented, followed by detailed descriptions of the tested methods. The goal is to analyze existing technologies and compare different methods critically.

Although experimental, the contribution of this chapter is significant as deepfake models have been developed and optimized for non-aerial images, and, to our knowledge, no one has tested the creation of geographic deepfakes using inpainting models. It is essential to develop a similar method by considering the characteristics of aerial images, including adapting depth, loss functions, and architecture to the large size of the images and areas to be modified. Geographic data, specifically building footprints, must also guide the results, and training must adapt by grouping data with common geographic features. The method in this chapter is original, as it requires the creation of a testbed of aerial images to train these implemented models. This will document their results and identify which ones may pose a threat.

The goal of this phase is to mask elements in high-definition images and test the capabilities and limitations of different models on geospatial images. However, it should be noted that only small elements can be masked, as larger high-precision images require more memory.

The selection of methods was made based on their popularity in previous research. Thus, well-established methods such as Context Encoder and GL-GAN, as well as two partial convolution models, were tested. Additionally, an implementation of GL-GAN called Edge-Connect was used as it allows guiding results with geographic data.

It is expected that, given its shallow depth, Context Encoder will yield the poorest results and SN-GAN the best. With its partial convolution, SN-GAN is anticipated to handle

aerial images better due to their large characteristic size. Deepfakes in environments with the most complex shapes are expected to be the least realistic. Finally, EdgeConnect should generate good deepfakes because it can guide results with geographic data, namely building footprints.

1.2 Datasets

The first step of the work involved creating datasets. As mentioned in the introduction, images were separated based on the context, namely water, road, residential area, countryside, and forest. Unfortunately, these datasets will not be available as open source due to the privacy policy of the Geoindex satellite data source.

1.2.1 Water Environment Datasets

For the aquatic dataset, databases of MNT images were used, specifically images from a flyover of the city of Quebec. The dataset contains several boats and associated disturbances, shallow areas, zones with varied reflectivities, and several buoys. Shores and power lines were removed (Figure 27 and 28).

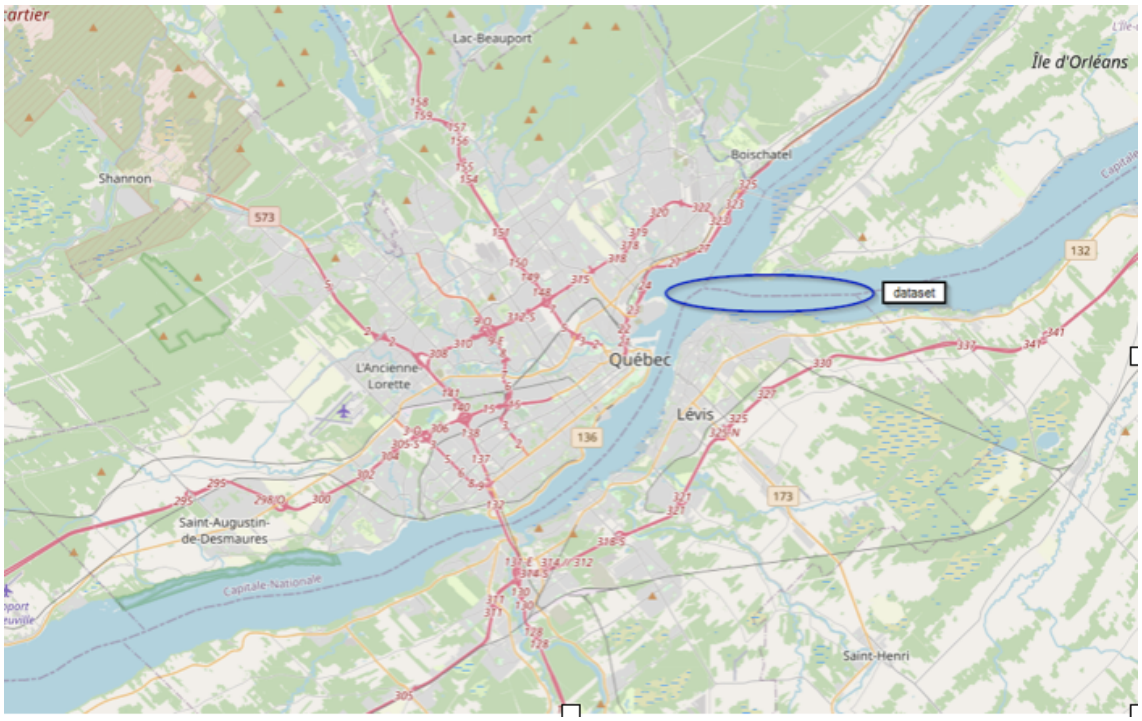


Figure 27: Space used for the aquatic dataset

The dataset consists of 8256 images of 256 pixels by 256 pixels, with an accuracy of 30 cm for each pixel.

Figure 28: Some examples of images from the aquatic dataset

1.2.2 Forest Environment Datasets

For the forest dataset, databases of MNT images were used, specifically images from the Lac Jacques-Cartier sector. Villages and significant roads were removed, but other human installations such as bridges, dams, isolated houses, and cars were retained. It should be noted that sharpness is not always constant due to the presence of mosaic borders. Brightness, as well as orientation, also vary. The forest itself is heavily exploited, with many cut forest spaces and paths, as well as the presence of numerous lakes (Figure 29 and 30).

Figure 29: Space used for the forest dataset

The dataset contains 50,000 images of 256 pixels by 256 pixels, with a per-pixel accuracy of 60 cm.

Figure 30: Some examples of images from the forest dataset

1.2.3 Rural Environment Dataset

For the countryside dataset, databases of MNT images were used. Specifically, images from the administrative area in the center of Quebec were used (Figure 31). Cities and forests were removed to retain only images representing rural spaces. The presence of roads, water areas, and various agricultural buildings should be noted.

Figure 31: Space used for the countryside dataset

The dataset contains 200,000 images of 256 pixels by 256 pixels, with a pixel precision of 30 cm.

Figure 32: Some examples of images from the rural dataset

1.2.4 Road Dataset

For the road dataset, the MNT databases were utilized, specifically, images from the central area of Quebec, the same ones used in the rural dataset (Figure 33). To create this dataset, Microsoft and OpenStreetMap APIs, along with ministry data, were used to retrieve only road images without buildings. Building footprints and road data were collected, and locations of roads without buildings nearby were stored. Images located at these locations were then extracted.

The dataset contains 41,347 images of 256 pixels by 256 pixels, with a pixel precision of 30 cm.

Figure 33: Some examples of images from the road dataset

1.2.5 Urban Dataset

For the urban dataset, MNT databases were used. Images from Quebec City were filtered based on the presence of buildings in the image exceeding a certain threshold. However, vacant lots were noted since building and image data are from different years. The presence of some buildings partially hidden by trees is also noteworthy. Building footprints found on the city's website were used here and kept to guide some of the selected models.

The dataset contains 146,030 images of 256 pixels by 256 pixels, with an associated pixel precision of 10 cm. Figure 34 shows examples of images from the urban dataset. The middle-left image represents building footprints.

Figure 34: Some examples of images from the urban dataset

1.3 Baseline

In this second part, a non-deep baseline model will be established. Initially, inpainting methods using diffusion will be tested. Subsequently, the goal is to achieve a more competitive baseline using the PatchMatch algorithm. As mentioned in the literature review, PatchMatch is the most widely used non-deep method. Finally, a quick comparison will be made with the most popular commercial inpainting solution, Photoshop.

1.3.1 Methodology

Two diffusion inpainting methods were tested. The first is based on Navier-Stokes [52] and is native to the OpenCV library. The second is a method from 2018, native to the Scikit-image library[53]. These two methods are the most used diffusion inpainting methods and are also easy to use. Drawing inspiration and optimizing[54], PatchMatch was implemented entirely in Python.

1.3.2 Results and Discussions

1.3.2.1 Diffusion Method

Figure 35: Sklearn

Figure 36: CV

It is evident that diffusion inpainting methods are not suitable for the research objective, as illustrated in Figure 35 and 36. These methods are only useful for small inpainting areas. In contrast, the goal is to detect modifications that alter the meaning of the images and are difficult for a human to perceive. This diffusion inpainting approach will be discarded for the rest of the work.

1.3.2.2 PatchMatch

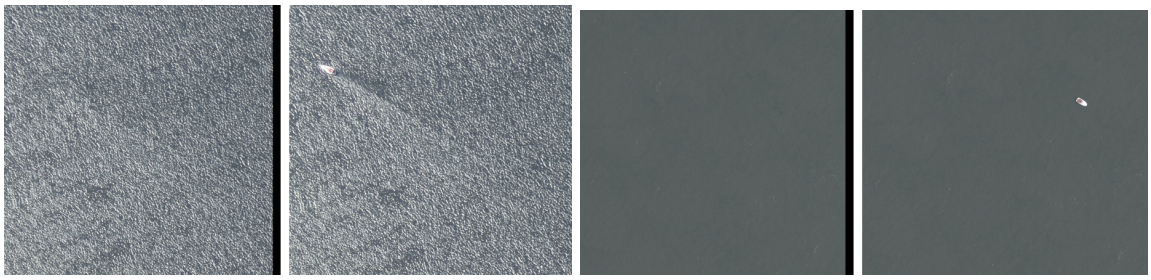


Figure 37: PatchMatch used on small areas to make boats disappear (left: modified image, right: original image)

Figure 38: PatchMatch used in a forest, the black image represents the mask used

Figure 39: PatchMatch in an urban environment

Figure 40: PatchMatch in a parking lot

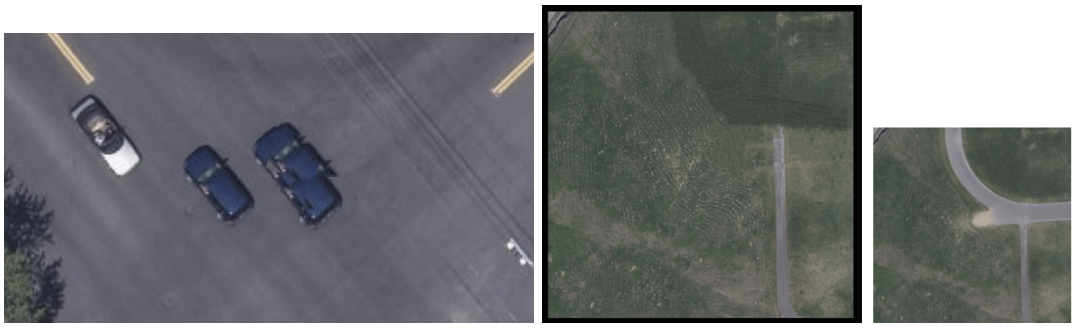


Figure 41: PatchMatch on roads

Figure 42: Example of a large area in a context with multiple complex shapes that is a failure

The PatchMatch method showed promising results even for significant inpainting areas, such as bodies of water (Figure 37). However, as the inpainting area and patterns become more complex, the algorithm produces less realistic results, even leading to entirely unrealistic outcomes (Figure 39). Indeed, this method lacks intelligence and an understanding of semantics. Additionally, the results exhibit sharpness issues when examined in detail (Figure 38) and difficulties in completing a pattern, as seen, for instance, with cars (Figure 40 and 41). This is likely due to the chosen patch size, the search radius in the implementation, and the lack of understanding of the pattern and its boundaries. Moreover, the result is confusing in large images with a substantial inpainting area (Figure 42). However, due to the ease

of implementing PatchMatch and the absence of training, this algorithm presents significant potential for producing misinformation in practical geospatial contexts. The result obtained with PatchMatch is similar to the result with Photoshop. Therefore, PatchMatch will be considered as the baseline. Notably, it is precisely an adaptation of PatchMatch that inspired GL-GAN. A better understanding of this method will provide a more critical perspective on GL-GAN. Nevertheless, before that, older deep methods will be tested.

1.4 Context Encoder

In this section, the first GAN method for inpainting problems, namely the Context Encoder, was tested.

1.4.1 Methodology

The following implementations were used[55][56]. However, improvements were added to enhance training stability. These improvements include adding a sigmoid layer, introducing a random mask, adding dropout layers, changing the activation function from Relu to Leaky Relu, reducing the depth of the discriminator, and training it every two iterations. A more suitable inpainting loss function was used based on a weight matrix that considers distances from the inpainting area. The weights are inversely proportional to the distance from the inpainting area. It should be noted that, although not always specified, similar modifications were made to the subsequent models when their training was unstable. This was often characterized by mode collapse or an inability to learn properly compared to the literature results. The goal was to rebalance the training game, although this was not always achievable. The depth of different layers in the generator model is presented in the source code attached to this document.

1.4.2 Results and Discussions

A balanced adversarial loss was observed for simpler datasets with homogeneous areas. Figure 43 shows the evolution of the discriminator and generator loss functions during training on the water dataset. In contrast, Figure 44 depicts the evolution of the L1 loss during this training. The training is balanced, meaning that neither the discriminator nor the generator becomes too strong compared to the other. However, as in the literature, in complex environments such as the urban dataset, the adversarial loss becomes highly unbalanced in favor of the discriminator, which almost always manages to detect fake images.

Figure 43: Evolution of adversarial losses during training

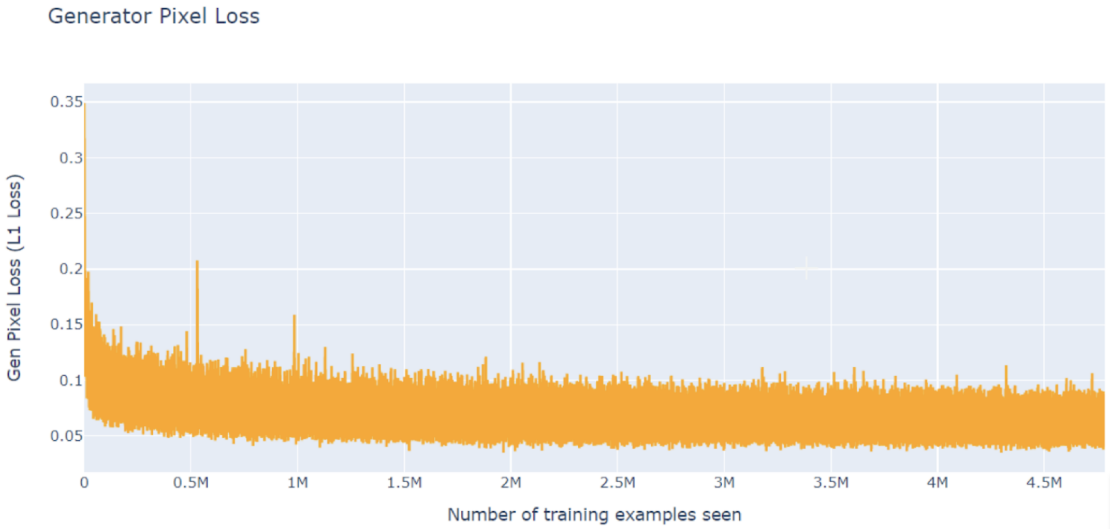


Figure 44: Evolution of L1 loss during training

The results were obtained after a training duration of 10 hours for simple datasets and 24 hours for the urban dataset. The training was conducted on an Nvidia Titan V graphics card with 12 GB of dedicated graphics memory. The training epochs ranged from 20 to 200, depending on the dataset. Attempts were made to train for longer durations, but no significant differences were observed.

Figure 45: Result in the aquatic environment

Figure 46: Result in the forest environment

Figure 47: Results in the urban environment

Results obtained with Context Encoder are unsatisfactory as the generated images are blurry with visible borders—a problem also found in the literature. The achieved results remain below those in the literature (Figure 48). Furthermore, results on road, countryside, forest, and water datasets are very disappointing due to the model's inability to capture details and the overemphasis on the L1 loss, leading to uninteresting uniform areas (Figure 45 and 46). However, in urban images, although the result is also blurry, the diversity of colors shows that the model has a genuine semantic understanding of the images, such as road boundaries and building shapes. It can even consistently create intersections (Figure 47). However, the absence of a balanced adversarial loss, as observed in the literature, to the extent that the weight of the adversarial loss is 1 against 999 for the L1 loss. Given that the context encoder often serves as preprocessing in subsequent methods, the algorithm's semantic understanding of images is a good sign that much better results could be achieved later. The tested method failed to outperform the established baseline. The overall results are disappointing, leading to the exclusion of this method during the geospatial deepfake detection phase, as its detection is straightforward. Additionally, it is worth noting that the obtained results have a quality similar to those presented in Section 0.2.3, Figure 8. A noticeable improvement would be the use of partial convolution. Becaus generally, the algorithm is more confident at the image

edges and shows some uncertainty in the middle of the inpainting area. In the literature, the discriminator only considers the inpainting area, but exploring the impact of allowing the discriminator to examine the entire image would be interesting. All of this will be tested in subsequent approaches.

Other improvements could be considered: training the model with a fixed mask for the first 300,000 iterations, then switching to a random mask. Tests have shown that the fixed mask quickly reaches a limit but learns faster initially. Another improvement would be to evolve the loss function with epochs by giving more weight to the middle of the image midway through training, as there is little evolution in the last 50 epochs in the middle of the inpainting area.

Figure 48: Classic results observed in the literature[4]

The conducted tests also highlighted another limitation of the technology. Indeed, a too-small inpainting area hindered learning due to an information overload. At the same time, larger images quickly increased the number of weights to memorize, exceeding the limit of 12 GB of dedicated graphics memory, making it impossible to process images larger than 256 pixels with current hardware.

Thus, the context encoder represents a shallow risk for the production of reasonably sized deepfakes, as they are easily identifiable. Partial convolution could, however, improve these results.

1.5 GL-GAN

This section will be dedicated to a more modern and complex approach offering more realistic results for inpainting problems: the GL-GAN. It is a popular choice receiving significant attention in the literature.

1.5.1 Methodology

The implementation[57] was used. It consists of two consecutive GL-GANs with the addition of a Context Attention module (Figure 49). Despite the popularity of this repository, minor but essential improvements were made to enhance training stability and result quality. The details can be found in the source code attached to this document.

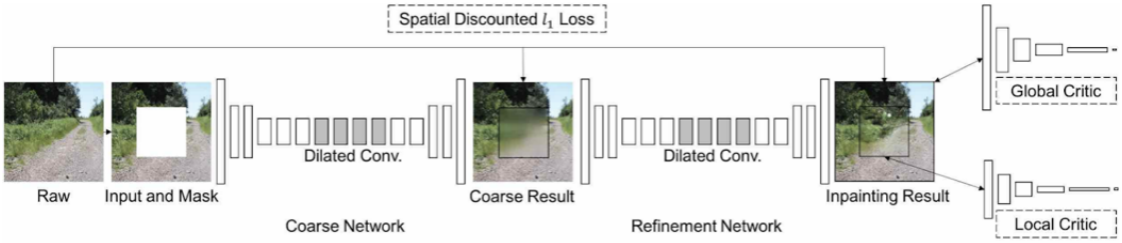


Figure 49: Architecture of the tested GL-GAN[18]

1.5.2 Results and Discussions

The results presented in this section were obtained after 1 to 2 days of training and several iterations between 60,000 and 180,000 for batches of 8 images, using the same hardware as in the previous experiments. The adversarial loss was observed to be more balanced in complex environments, even though the results obtained in these conditions were the least realistic. Nevertheless, the discriminator's loss remains slightly lower than the generator's, and the game's imbalance is not total. Examples of results obtained are presented below.

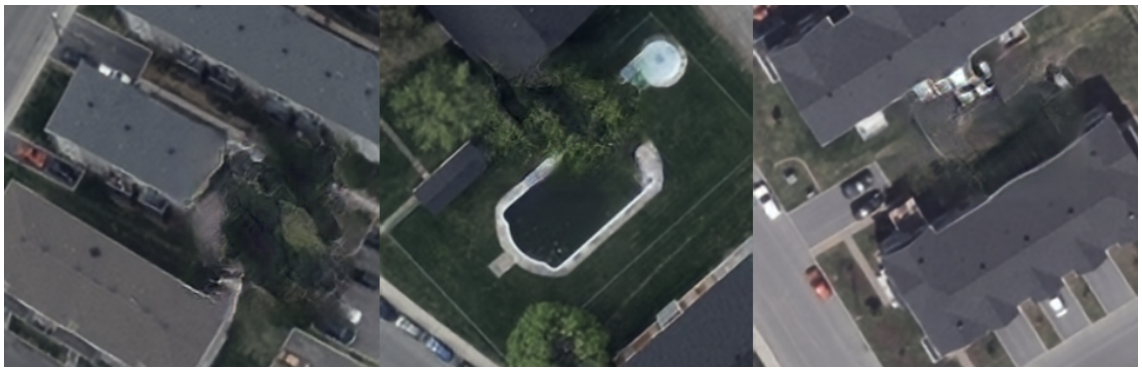


Figure 50: Examples of geospatial deepfakes produced with GL-GAN

Figure 51: Examples of failed geospatial deepfakes in the urban environment produced with GL-GAN

Figure 52: Example in the forest environment

Figure 53: Examples in the forest environment

Figure 54: Examples in the rural environment

Figure 55: Examples of failures in the rural environment

Figure 56: Examples in the aquatic environment

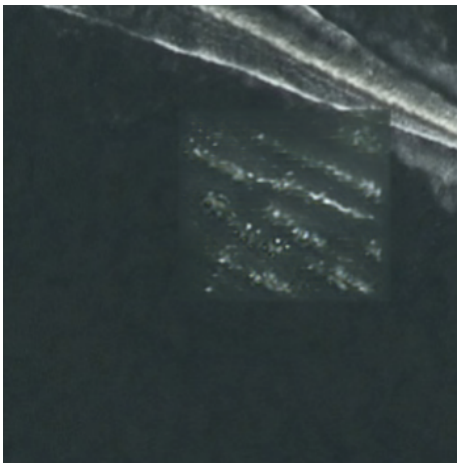


Figure 57: Example of failure in the aquatic environment

Thus, the results obtained with the GL-GAN approach are significantly better than those of the Context Encoder. The borders are much more discreet and almost invisible, and the inpainting area has much less blurriness. Moreover, the results are vastly better than those reported in the literature (Figure 8). Overall, this method outperforms the baseline. The deepfakes are particularly successful in forest environments (Figure 52 to 53), with realistic shadows. However, the quality may sometimes slightly decrease, the color may vary, or a blurry area may be present in the center of the modified area. The model even shows creativity at times (Figure 50, 51, and 55). In agricultural environments, the results are also excellent (Figure 54), although the complex shapes of the buildings pose problems (Figure 55). In the aquatic environment, however, the results are below the baseline (Figure 56), mainly due to visible borders and the difficulty in understanding waves (Figure 57), which are interdependent. Finally, in urban environments, there may be blurry areas that make the image less realistic, along with deformations (Figure 50 to 51). Artifacts from oversampling can also appear when shapes and colors are too complex. For example, in Figure 51, the algorithm attempted to generate multiple cars on the right image, but their small size, complex shape, and various colors confused the model. Overall, the models manage to capture details. These results are on par with the literature; some examples are presented below (Figure 58).

Figure 58: Some examples of images manipulated with the GL-GAN method in the literature[5]

The tests also showed that this method had limited stability. Indeed, mode collapse and the phenomenon of "vanishing gradient" were observed in the literature. However, they were also observed here, notably with the extreme case of predicting an entirely black image after extended training (Figure 59). This shows that the algorithm considers the adversarial loss much more, which was not the case with previous models.

Figure 59: Examples of observed "mode collapse"

The results obtained with GL-GAN have shown that the model can produce almost credible deepfakes in specific geographical contexts, with a clear improvement over the baseline, especially for forested and rural areas

1.6 EdgeConnect

As mentioned in the introduction, EdgeConnect is a hybridization between GL-GAN and PG-GAN while allowing the integration of geospatial data to guide desired results, such as building footprints.

1.6.1 Methodology

The official implementation done in PyTorch[58] was used, with no significant modifications other than adaptation to this dataset. A structure schematic can be found in the state of the art. The code is available in the appendix of this document. As mentioned earlier, this implementation allows building footprints to guide the results. Thus, building footprints were used during training and validation, but only in urban environments.

1.6.2 Results and Discussions

The results were obtained after 1 to 7 days of training with up to 900,000 batch iterations for the urban environment, which corresponds to 42 epochs for the urban environment. Training was performed on the same hardware as before. Furthermore, the adversarial loss and the training generally showed much better stability. However, the discriminator remains slightly superior to the generator. Quantitatively, Figures 60 and 61 show the evolution of loss functions during training. Qualitatively, this is evident by the absence of "mode collapse" or "vanishing gradient". Examples of results can be found below.

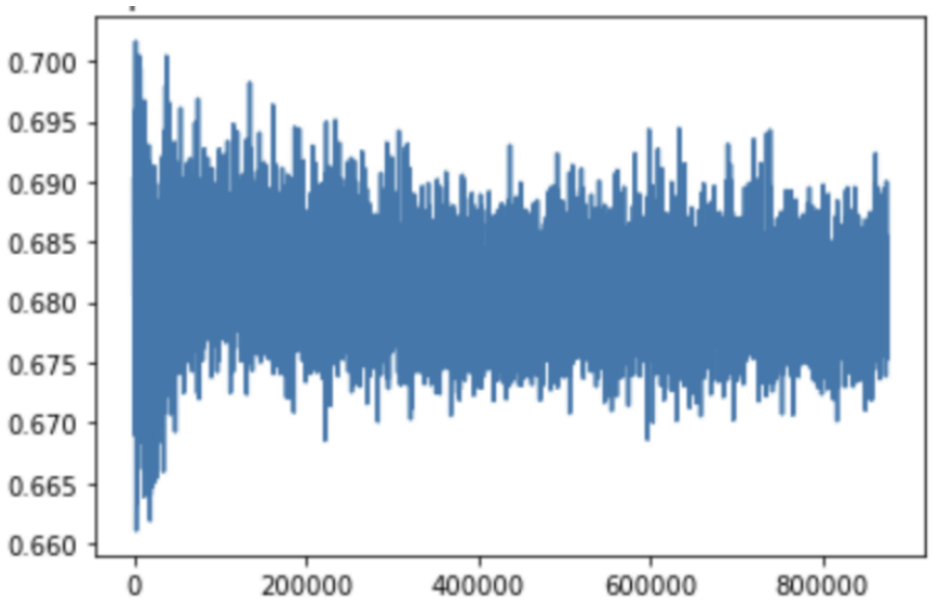


Figure 60: Evolution of discriminator loss as a function of training iteration in the urban environment

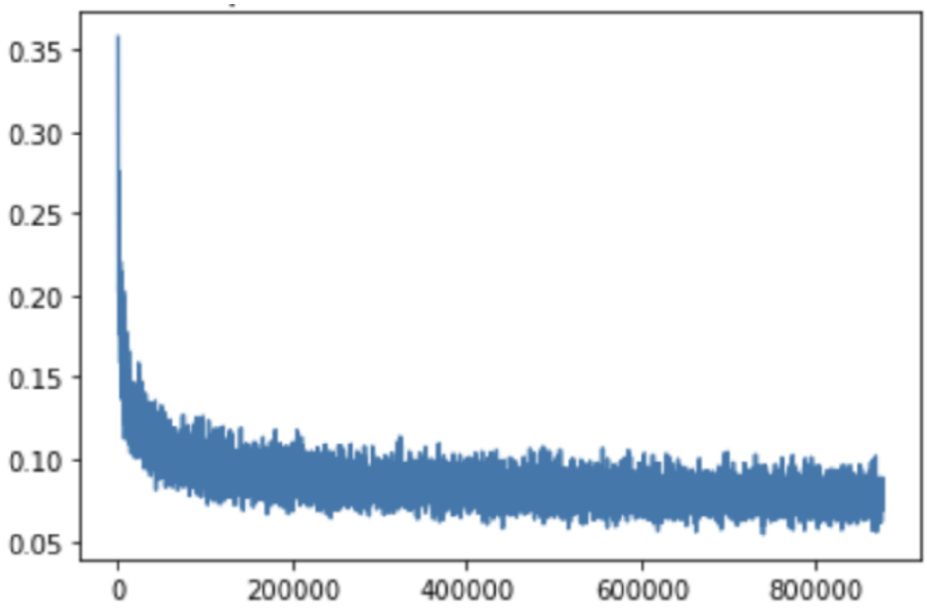


Figure 61: Evolution of L1 loss as a function of training iteration in the urban environment

Figure 62: Examples from the forest environment

Figure 63: Examples from the rural environment

Figure 64: Examples of roads

Figure 65: Examples from the urban environment

The results obtained with EdgeConnect are significantly better than those obtained with previous methods, especially in rural and forest environments (Figure 62 and 63), where the border between the original image and the generated image is very discreet, making it difficult to detect the inpainting area. However, in these environments, color patches may appear. As illustrated in Figure 62, a cluster of blue pixels is visible in the left image. Straight lines can also be distorted, especially in road datasets (Figure 64). Although the shapes are straighter and more relevant in complex environments, oversampling artifacts in the form of blurry brushstrokes may appear (Figure 65). However, these artifacts decrease over time. Indeed, after seven days of training, it was decided to stop training even though qualitative improvements were still observed. It should be noted that the literature shows that reducing the number of sketches to guide the model as input or reducing the algorithm's sensitivity to them can reduce these artifacts[8].

The results obtained are equivalent to those in the literature (Figure 66). Thanks to the stability of this method, the results obtained did not qualitatively converge at the time training was stopped, suggesting that even better results could be achieved. Also, a test with a pre-trained model showed that, although the results were not immediately better, training is much faster.

Figure 66: Example from the literature[8]

In conclusion, the results are promising, and the ability of this method to be guided by geospatial data makes it a potentially dangerous algorithm if used for malicious purposes. Since oversampling artifacts and color patches appear in the middle of the image, partial convolution could further improve these results.

1.7 PConv

In this section, the effect of partial convolution was tested. As mentioned in the literature and expected, partial convolution should improve the obtained results. To better visualize its impact, partial convolution was tested on a simple model, specifically the U-Net. The improvements can thus be easily observed.

1.7.1 Methodology

For this purpose, the following implementation was used[59], and it was decided not to add a discriminator, given the low impact observed in previous tests. This decision also minimizes the workload effort :). However, it is worth noting that this implementation includes a small pre-trained VGG network, although this should remain the same results. The depth of the different layers is the same as that of section 1.4.

1.7.2 Results and Discussions

The results were obtained after 24 hours of training on the same hardware as before. An example of a result obtained can be found below (Figure 67).

Figure 67: Results of a U-Net with partial convolution in an urban context

Thus, with only a few additional hours of training compared to the Context Encoder model and despite the absence of adversarial training, significantly better results are obtained. Indeed, in addition to a blurry area in the center of the image, the boundary is much less visible, and the model manages to produce more details as the blur is relatively less important. Thus, partial convolution brings a clear improvement in inpainting performance. But, it is essential to qualify this conclusion because a small pre-trained VGG preprocessing network has been added compared to the previously tested U-Net. However, several other improvements have not yet been implemented here. However, the difference in observed quality corresponds to what has been seen in the literature, as illustrated in Figure 68, where PConv generally produces less blur than Conv and provides higher-quality results. Moreover, given that the inpainting area is large, it is not surprising that partial convolution still produces blurry areas. It should also be noted that partial convolution alone is insufficient to produce credible large-scale geographic deepfakes, representing a limitation of this technology.

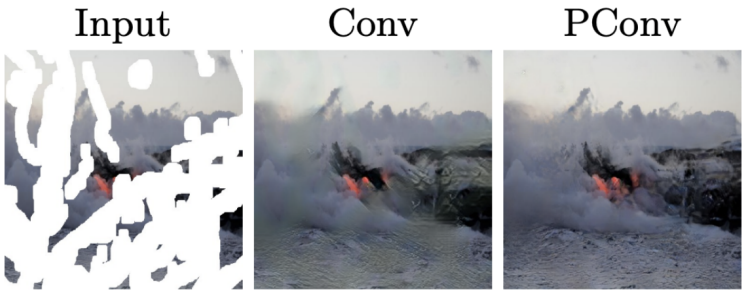


Figure 68: Example of observed differences in the literature[10]

Since the invention of partial convolution, improvements have been made, showing better results, notably the Gated convolution, which, instead of following a linear rule, proposes a smoother partial convolution. The literature has highlighted that this convolution generally yields better results. Moreover, given the encouraging results of GL-GAN and the observation that partial convolution alone is insufficient, it would be interesting to test this type of convolution in hybridization with the method tested in section 1.5 (GL-GAN). This is precisely what has been done in the next part through widespread implementation.

1.8 Gated convolution: SN-GAN

In this part, one of the most recent inpainting methods was tested: SN-GAN. This method is an improvement over the one presented in section 1.5 but uses partial convolution and, more specifically, gated convolution.

1.8.1 Methodology

For this test, the official implementation under TensorFlow[60] was used. The network architecture is illustrated in Figure 69, where the large green layers have a depth of 64 for input images of 256x256x3. The medium green layers have a depth of 128, while the large green and orange layers have a depth of 256. The yellow layer has a depth of 512. It is noteworthy that although the implementation allows the use of sketches, they were not used to reduce artifacts observed with EdgeConnect. No significant improvement other than adaptation to this data was made.

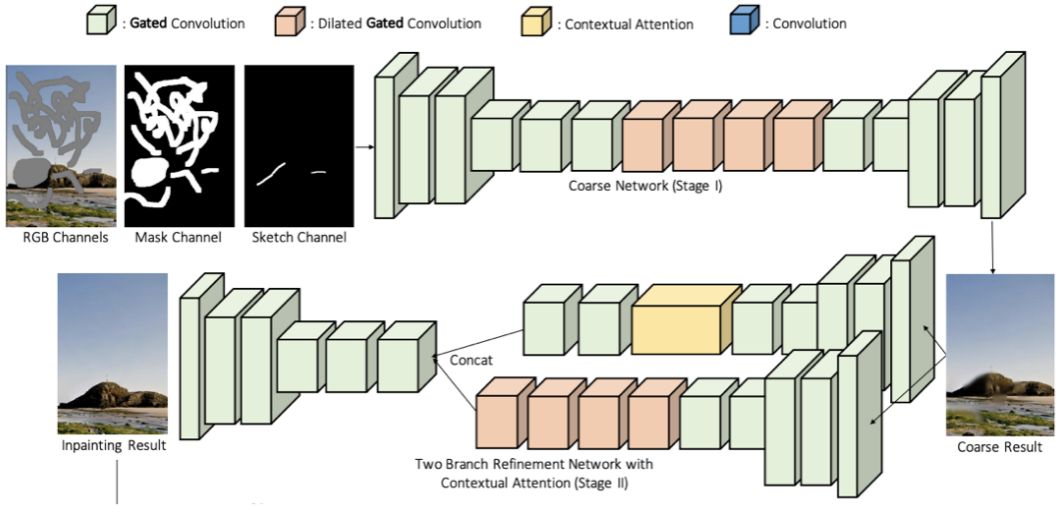


Figure 69: Architecture used[19]

1.8.2 Results and Discussions

In this section, the results will be presented. The results were obtained after 3 to 10 training days with batches of 5 images. Training was performed on the same hardware as before and was stable. First, the results from datasets that consistently produced non-identifiable fakes will be presented, then errors in these datasets will be shown. Finally, datasets that regularly failed to create credible deepfakes will be presented.

1.8.3 Unidentifiable Results

In this subsection, the results of the road, forest, and rural datasets will be presented.

Figure 70: Example of a path in the road dataset

Figure 71: Example of a large road with a car in the road dataset

Figure 72: Example of an image of a path hidden by trees in the road dataset

Figure 73: Example of a representative image from the forest dataset

Figure 74: Example of a representative image from the forest dataset

Figure 75: Example of a representative image from the forest dataset

Figure 76: Example of a representative image from the rural dataset

Figure 77: Example of a representative image from the rural dataset

Figure 78: Example of a representative image from the rural dataset

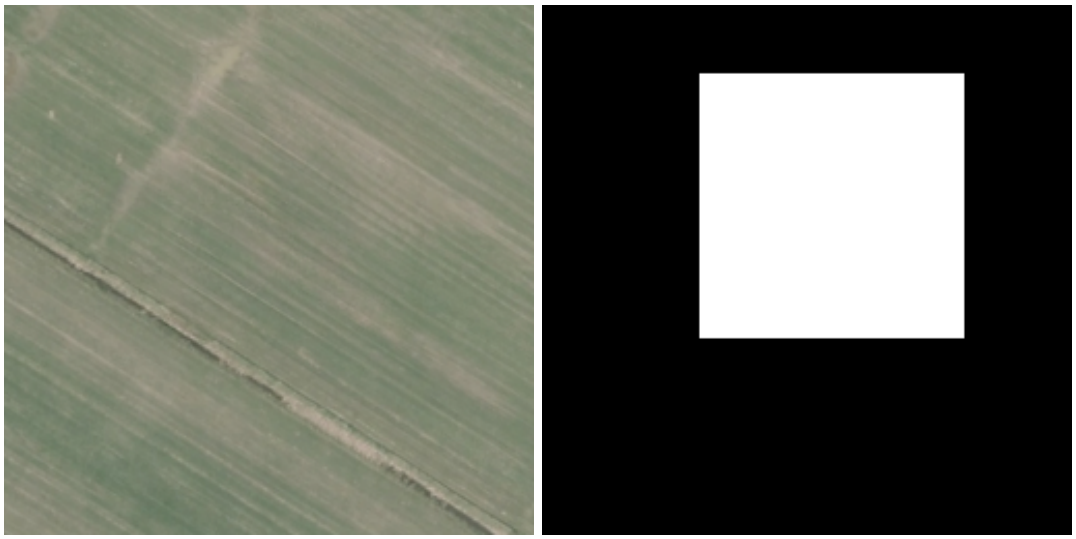


Figure 79: Example of a representative image from the rural dataset

These images are qualitatively convincing, and false areas cannot be identified by a human observer. Borders are no longer visible, and shadows are perfectly generated. There are no issues with sharpness or the presence of artifacts. The goal of creating geographically convincing deepfakes for a human observer is achieved in these datasets.

1.8.4 Rare Failures

In this subsection, examples of rare failures from the datasets in the previous section will be presented. These examples illustrate cases where the inpainting method failed to

generate credible images.

Figure 80: Examples of Failures

Figure 81: Examples of Failures

Figure 82: Example of Failure

The left image in Figure 80 shows the creation of a not-very-realistic pond in the city dataset. The right image in the same figure shows a road in the rural dataset. Since the dataset contains few roads, attempts to create cars were unsuccessful and easily identifiable. Thus, underrepresented elements in the dataset sometimes lead to errors. The right image in Figure 81 shows problems with not very straight lines, sometimes present in the road dataset. This is likely due to sharpness issues in real images being attempted to be generated. The left image in the same figure shows a sharpness problem in removing a small agricultural building. Finally, Figure 82 displays an oversampling artifact. However, these are much less common and more discreet than with other models. Overall, these issues are easily resolvable by regenerating results until the desired quality is obtained.

1.8.4.1 Identifiable Results

The results of the water and city datasets will be presented in this subsection.

Figure 83: Typical Result in the Water Dataset

Figure 84: Examples of Results in the Urban Dataset

Figure 85: Examples of Results in the Urban Dataset

Figure 86: Examples of Results in the Urban Dataset

Figure 87: Examples of Results in the Urban Dataset

Figure 88: Examples of Results in the Urban Dataset

Producing credible deepfakes geography in two datasets was unsuccessful or unsatisfactory. This subsection presents these results. The production of credible deepfakes was unsuccessful

in the water dataset. This is due to an overweighting of the type 1 error compared to the adversarial error, which was not balanced, and to the highly homogeneous nature of the images. The pixel-wise loss functions naturally encourage the production of a blurry image because a blurry image reduces the likelihood of making significant errors, aggravated by the uniform and monochromatic nature of the water. This ended up producing nearly the same output regardless of the input. During testing, balancing learning by weighting only the two types of losses was impossible. It is unlikely to be a mode collapse or vanishing gradient issue, as training was attempted three times with the same result, and the discriminator losses remain almost identical. It is also possible that the latent space is too tiny for this environment. Finally, Figures 84 to 88 show the type of results obtained in the urban dataset. Figure 84 demonstrates the algorithm's struggle to create perfectly round shapes. Other figures show recurring problems with sharpness and non-straight shapes. Failed images, such as the left image in Figure 85, are common in the dataset. However, this failure can be nuanced, considering the presence of convincing deepfakes. Given the generative nature of the model, it would be possible to recreate it several times until a more satisfactory result is obtained. Furthermore, the difficulty is doubled since accuracy is three times more critical in the urban dataset. On the one hand, this dataset has more complex shapes and colors; on the other hand, the details of these complex shapes and colors are three times higher than those of other environments. Despite this, shadows are generally present; shapes are often straight, and, knowing that aerial images often have sharpness issues, the deepfakes remain discreet.

Figures 89 and 90 present the evolution of the adversarial loss function during training. Figure 89 shows the evolution of the loss function in the city case. As training progresses, the discriminator becomes better at detecting fake images. In contrast, in the forest environment represented in Figure 90, the discriminator has increasing difficulty distinguishing between real and fake. However, in both cases, the training is not balanced at all, as the discriminator significantly underperforms the generator. This result is even slightly worse than a policy that gives an equal probability of true and false. This observation regarding the discriminator's ability to detect fake images is not reassuring for future detections. It also shows that improving training by rebalancing the discriminator and the generator should still be possible.

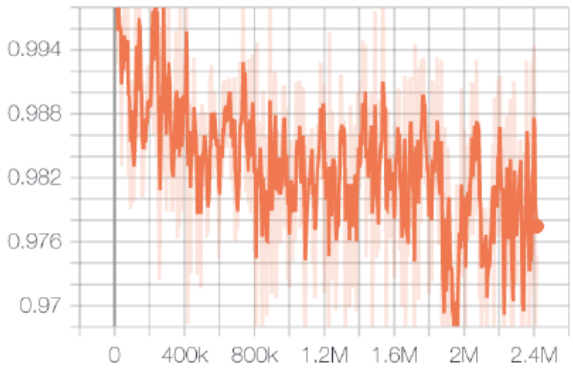


Figure 89: Loss Function evolution during training on urban dataset

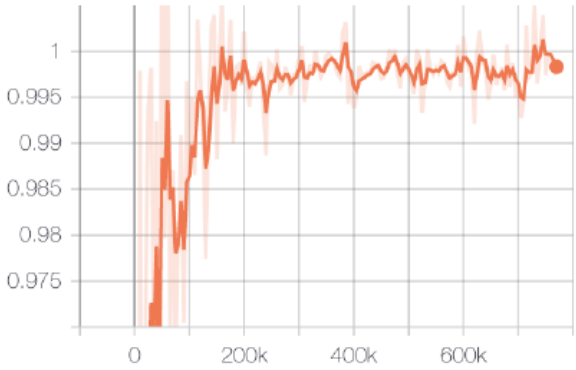


Figure 90: Loss Function evolution during training on forest dataset

Finally, roughness was examined to quantify the quality of these results and compare them to real images. Roughness is a characteristic that describes how irregular a surface is. If fake images exhibit roughness similar to real images in similar scenarios, it indicates that details and textures have been realistically modeled. On the other hand, if fake images have significantly different roughness from real images, it may indicate a lack of realism in textures or appropriate details.

To determine the average roughness of a set of images, the OpenCV library was used. Each image was converted to grayscale, simplifying roughness calculation while preserving essential texture information. Then, as roughness is the variation in pixel intensities over the surface of the image, the variance of pixel intensities within the grayscale version of the image was calculated. Variance is a proxy for roughness or texture, with higher values indicating greater roughness. In this case, the roughness of generated images will be compared within the same context, meaning the roughness of fake forest images will be compared to real forest images to avoid inherent roughness differences in the image context.

The average roughness (AR) of a set of N images can be calculated as follows:

$$AR = \frac{\sum_{i=1}^{N} V_i}{N} \tag{1.1}$$

Where:

ARrepresents the average roughness.

$N$is the total number of images.

$V_i$represents the variance of pixel intensities in the grayscale version of the i-th image.

Figure 91 presents the roughness obtained between fake and real images from each dataset. These results show that the roughness levels between these images are of the same order of magnitude, confirming that quantitatively the generated images have a quality level similar to real images. Furthermore, these results confirm that the urban environment is a more complex setting. The higher roughness of textures in the urban environment proves that modifications are more complicated and, therefore, more challenging to learn. This explains the performance difference between different datasets. However, it is important to note that roughness has limitations as a metric for measuring realism. Qualitative analysis remains the best source of comparison, as an image may have roughness similar to the dataset average but lack meaningful content. Nevertheless, it serves as a good indicator of the realism of textures. The next chapter will complement this quantitative analysis with deep methods that combine semantic understanding and statistical analysis. These are the deepfake detection methods presented in the state of the art. Indeed, geospatial deepfake detection tests will better analyze how discreet and realistic the produced deepfakes are.

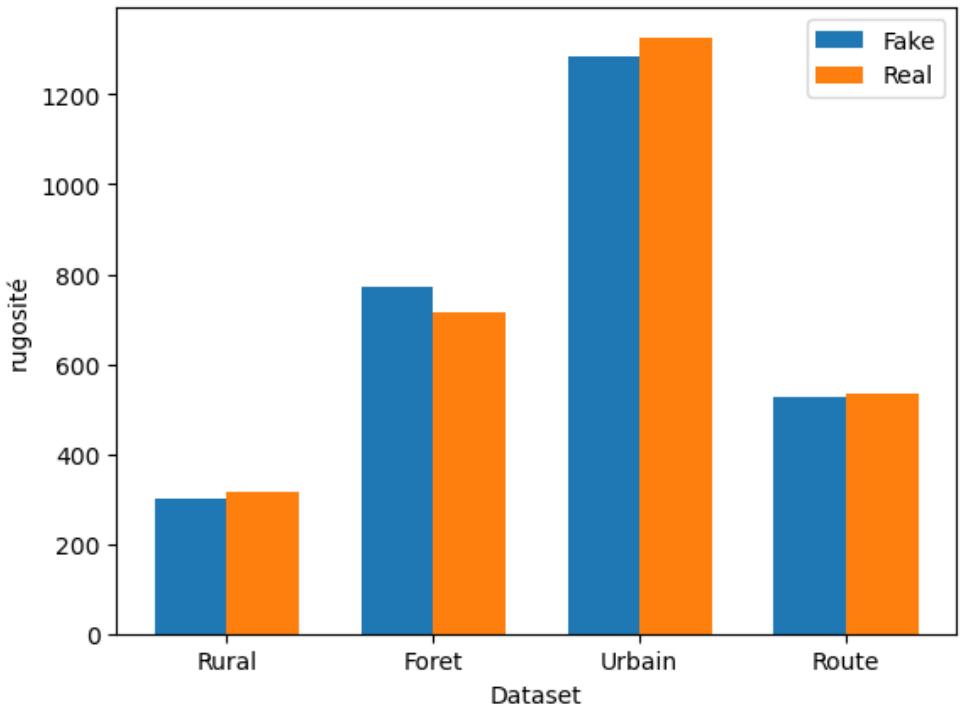


Figure 91: Roughness Comparison Between Fake and Real Images

The results allow us to conclude that SN-GAN is the most effective deepfake creation algorithm among the models tested so far. The results are qualitatively the best of all the models. Thus, the ability to produce credible geospatial deepfakes is highly elevated. Indeed, even in complex environments, the algorithm generates complex shapes and shadows and shows few oversampling artifacts. In simpler environments, the deepfakes are virtually undetectable to the naked eye. The goal of creating credible deepfakes is thus achieved. Prevention through automatic detection is, therefore, a priority.

Therefore, no further methods will be tested, especially since it would be challenging to visualize significant improvements given the already satisfactory results obtained. Moreover, the current state-of-the-art limit has been reached with this very recent method developed in 2021, which is still considered one of the best.

1.9 Conclusion

Thus, the results obtained with U-Net and Context Encoder are disappointing but can be greatly improved with the use of partial convolution. As for GL-GAN, in the form of two consecutive networks, it provides much better results that are good in regular environments but perform poorly in complex environments. With EdgeConnect, these results are greatly improved qualitatively and quantitatively with geospatial data and the PG-GAN hybrid.

Finally, SN-GAN is by far the one that provides the best results as it combines the performance of GL-GAN with the advantages of partial convolution. The results are almost always qualitatively undetectable in homogeneous environments and remain realistic in complicated areas, although the presence of errors should be noted. Moreover, it is the only GAN whose discriminator loss is significantly higher than the generator's, suggesting greater difficulty in detection and possible improvement. A summary table of the performance of the different algorithms can be found below:

Test Summary						
Tested Method	Baseline	Context Encoder	GL-GAN	EdgeConnect	Pconv	SN-GAN
Realism	✯		✯	✯✯	✯	✯✯✯
Geospatial Consideration				✯✯✯		✯✯✯
Stability and Balanced Loss	✯✯✯	✯	✯	✯✯		✯✯
Absence of Artifacts			✯	✯✯		✯✯✯
Popularity	✯✯✯	✯✯	✯	✯	✯	✯✯
Ease of Use	✯✯✯	✯	✯	✯✯	✯	✯✯
Convergence Speed	✯✯✯	✯✯	✯✯	✯	✯✯	✯
Sharpness	✯✯		✯	✯✯	✯	✯✯✯
Global/Risk	✯✯✯		✯	✯✯		✯✯✯

Table 1.1: Summary of Test Results

For the first time, thanks to these original tests, it is possible to understand the strengths and weaknesses of classical inpainting deepfake models. Additionally, this contribution has highlighted the presence of certain artifacts characteristic of large geospatial images, such as oversampling artifacts caused by the large size of the area to be modified. Even relatively old generations, like GL-GAN, can produce very discreet deepfakes.

The first part of the initial hypothesis is thus validated, namely that it is possible to create realistic deepfakes in the geospatial context. Moreover, many improvements are possible. For example, it is conceivable to enhance the discretion of deepfakes by integrating them into larger images, reducing the size of the areas to be completed, or decreasing the image quality, which also increases the masked area. However, in urban and aquatic environments, it is true that deepfakes are not perfect, and a human would be able to identify them in a majority of cases. This would require close attention, as sharpness issues are commonly observed in real images. In other environments, deepfakes are of sufficient quality to deceive a human. The next part will be dedicated to attempting to detect the created deepfakes.

Chapter 2

Methods for Deepfake Detection

This second chapter will be dedicated to the detection of previously created deepfakes. In this section, the conducted exploration will be developed and documented. Initially, the used datasets will be detailed. Subsequently, the classical FotoForensics methods will be explored, and their limitations will be analyzed. Some classical methods for general deepfake detection will also be tested. Finally, the limitations and gaps of methods specialized in "inpainting" detection will be explained. This will justify an original approach. An original implementation will then be developed and tested to have a method better suited to the problem. To conclude, an evaluation of the model's robustness will be conducted. The goal is to test the robustness of previously created deepfakes against classical deepfake detection methods and understand their limitations.

Compared to the generated images, it is expected that JPEG Ghost can easily identify oversampling artifacts due to the specificities of aerial images. The Zhao method should perform less effectively than reported in the literature. Given the quality of the produced deepfakes, only F3net is expected to yield good results. Nevertheless, it is crucial to verify this hypothesis.

The goal of this section is to detect at least 95% of deepfakes with the lowest possible type 1 error. Additionally, the method should empirically ensure a minimum level of stability. The methods chosen in this section have been motivated by their performance in the literature, and their details will be developed in each subsection. The objective of fake mask prediction methods is to surpass the performance of previously tested classical methods by proposing a method better suited for detecting fake zones in huge geospatial images. Our goal is to achieve a fake zone detection rate of 95%, with minimal false negatives and good stability. In the literature, an IoU score of 0.6-0.7 is considered sufficient for this goal[51].

2.1 Detection Datasets

The first part of this work involved creating a representative dataset of the created deepfakes.

To achieve this, two models were selected. The first selected model, PatchMatch, will be a baseline. The second selected model is the one that yielded the best results in the previous phase, namely SN-GAN.

For each model, six datasets were created, one for each environment, as well as a general set combining a mix of each environment in equal proportions. Specifically, each dataset included:

- a dataset for the aquatic environment, consisting of 5000 deepfake images with their associated masks and 5000 real images.
- a dataset for the urban environment, consisting of 5000 deepfake images with their associated masks and 5000 real images.
- a dataset for the rural environment, consisting of 5000 deepfake images with their associated masks and 5000 real images.
- a dataset for the forest environment, consisting of 5000 deepfake images with their associated masks and 5000 real images.
- a dataset for the road environment, consisting of 5000 deepfake images with their associated masks and 5000 real images.
- a general dataset consisting of all environments in equal proportions. This includes 1000 deepfake images and 1000 real images for each environment (aquatic, urban, rural, forest, and road), with their associated masks. The images were randomly sampled from the aforementioned thematic datasets. As a result, there are 5000 deepfake images with their associated masks for 5000 real images.

For image selection, random sampling was carried out from satellite images in the datasets described in Chapter 2. Once these images were excluded from the original datasets, retraining was performed, and deepfakes were generated. This ensures that none of the 5000 deepfakes were created from images used during training, except for the water dataset, due to the limited data. Additionally, the selected images representing real images were excluded from those used to generate deepfakes.

In addition to thematic datasets, a general dataset was also created by combining images from each dataset equally for both models. However, the water dataset from the SN-GAN model was excluded due to its simplicity, which would artificially inflate the results.

Thus, the general dataset consists of 500 fake images and their masks for the road, city, countryside, and forest environments for the baseline and 500 fake images and their masks for the road, city, and countryside environments, as well as the forest for the GAN model. This general dataset includes 1000 real images for each environment (water, road, forest, city, countryside). In total, this general dataset contains 4500 fake images with their associated masks and 5000 real images.

Post-training evaluation will always be conducted on a test dataset, representing 20% of the total dataset, which the agent has not seen during training.

2.2 Exploration of Classical FotoForensics Methods

Research on image manipulation detection methods, also known as FotoForensics, is quite mature. These are non-deep methods developed to detect images manipulated by human actors and requiring interpretation. This section explores the most common methods that have proven effective or could be adapted for deepfake detection. Qualitative results are presented. These approaches were developed chronologically between 2005 and 2015, for the most part. Several methods were tested, and those that yielded the most interesting results will be presented.

2.2.1 JPEG Ghost

Several oversampling artifacts have been observed in deepfakes, a phenomenon commonly documented in the scientific literature. These artifacts are characterized by random colors and a blurry area (Figure 92). They are clearly distinct from their environment and can thus be easily detected. To identify these artifacts, a method called "JPEG Ghost" was used[61]. This approach involves calculating a squared difference between the original image and the same image with reduced quality, highlighting parts of the image with different quality. The squared difference is chosen for its sensitivity to extreme values. The results of this approach are mixed. While this method can perfectly detect artifacts of significant size (Figure 92), it generally fails to detect artifacts of smaller size. Applying a square root to the difference between images could enhance the sensitivity of the approach but at the cost of over-detection. Moreover, this method does not work on the baseline PatchMatch, yielding few false positives. Ultimately, it is not scalable, non-adaptive, and relies on artifacts that tend to disappear with more recent deepfake methods. A more tailored training would render this method obsolete. The results are insufficient to detect these deepfakes geography, as the artifacts are not adequately detected, and only a tiny proportion of images exhibit such artifacts.

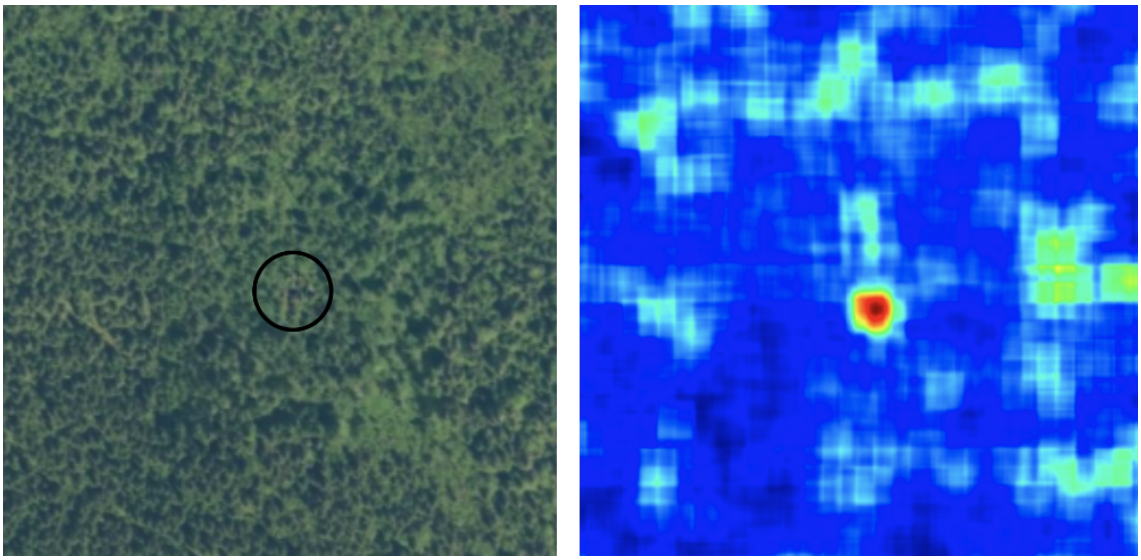


Figure 92: Example of artifact detection with JPEG Ghost

2.2.2 PCA

The Principal Component Analysis (PCA), a well-known method, involves re-expressing multidimensional data in a lower-dimensional coordinate system. It identifies principal axes capturing maximum data variance and projects the data onto these axes. While typically used to transform color images into black and white, PCA has shown some results in detecting manipulated photos[62]. This method has been tested.

The results of this approach are inconclusive. Out of the 20 images tested, only one was perfectly detected, while others produced imprecise results requiring interpretation. Figure 93 illustrates a well-detected result using the third principal component, extracting a unique feature for the fake zone. Another example of an insufficient result, though interesting, is shown in Figure 94, also using the third principal component. However, unlike the previous figure, no distinct feature was identified between the fake and real zones. The results are, therefore, inadequate for reliable application.

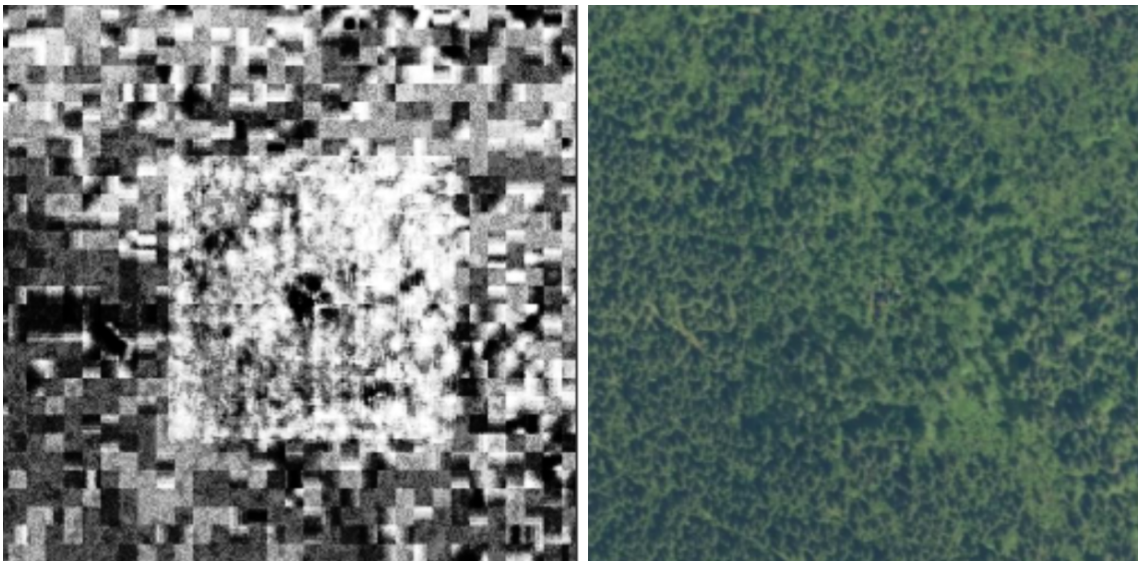


Figure 93: Third PCA component

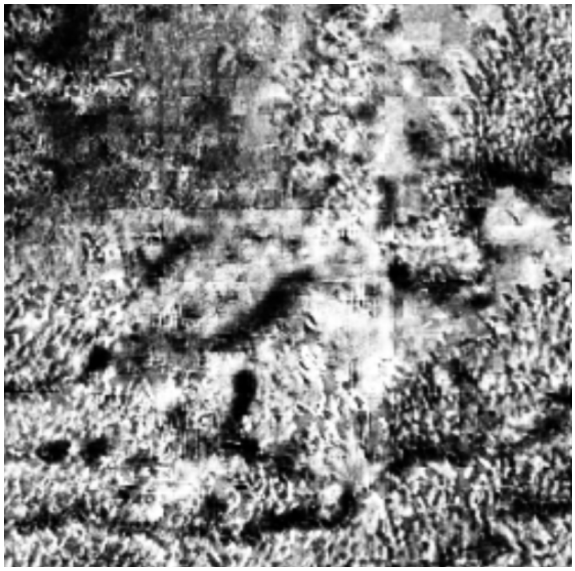


Figure 94: Third PCA component

2.2.3 Compression Difference Methods with Non-Quadratic Error

In this category of approaches, the Error Level Analysis (ELA) method was initially tested. ELA, the most well-known method in FotoForensics, aims to detect different compression levels. It identifies areas in the image with a different source or different sharpness. Despite being widely used, ELA provides no theoretical guarantees. Essentially, it is the same method used for JPEG Ghost but with a non-quadratic error. For implementation, the difference with an image compressed to 99% was calculated.

Quantitative results for the ELA method are positive but insufficient. This approach performs well in more complex environments, almost perfectly in urban settings, with some easily correctable over-detection. Figure 95 on the left shows results in urban settings. However, it fails in homogeneous environments. The image on the right illustrates this failure in a forest environment. The fake zone is uniformly noisy throughout the image, indicating lower quality than the rest, making it identifiable. However, in a forest, the image is uniformly noisy, preventing the identification of the fake zone. The results are, therefore, inadequate. Other methods based on the same principle were tested, yielding similar results—excellent in urban areas and of no interest in more homogeneous environments.

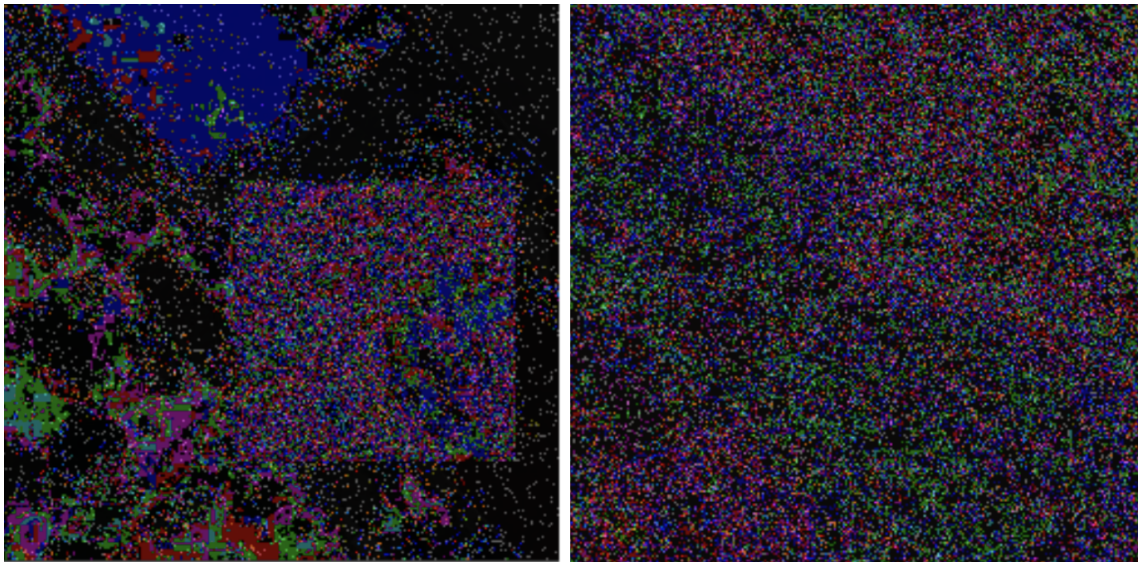


Figure 95: Results of the ELA method

The observed difference in performance is likely due to variations in deepfake quality. In urban areas, there is a more significant difference in quality between the fake zone and the rest of the image, making it easily identifiable by the ELA method. In contrast, in homogeneous environments, there is no difference in quality between fake and real zones, making it challenging for the method to discriminate between them.

2.2.4 Synthesis

Thus, the exploration of FotoForensics methods has shown interesting but insufficient results for robust and versatile exploitation. However, a common approach combining different methods associated with a deep model, such as a deep method taking input images with previously tested treatments, could yield more usable results. Nevertheless, a lack of detection is felt in homogeneous environments. Furthermore, one of the main problems with FotoForensics methods is their rigidity, making them easily circumventable. Therefore, deeper approaches adapted to deepfakes will be tested in the continuation of this work.

Additionally, clone search methods worked on images from the baseline for obvious reasons but did not work on images generated by the GAN.

2.3 Classical Deepfake Detection Method

In this section, classical approaches to deepfake detection will be tested, focusing on binary identification of real or fake images. The best methods and most significant progress have been made for this task. Although these approaches are unsuitable for previously produced "inpainting" deepfakes, testing them to determine which ones would be relevant for implementation in a hybrid approach would be interesting.

To use these classical approaches, only the fake part out of context was used to detect it. A random mask of equivalent size was created for real images to have images of the same dimension.

First, Zhao's approach for deepfakes geography detection will be tested. Then, approaches based on the Fourier spectrum will be explored. Finally, the F3Net model presented in the introduction will be tested on the datasets. Some other deep methods discussed in the introduction showed too much instability or too little technical advancement compared to the F3Net architecture and will not be discussed here.

2.3.1 Zhao's Method

This sub-section will address Zhao's method for deepfakes geography detection. First, a technical point will be recalled, and then the results will be presented and discussed.

2.3.1.1 Method

Zhao's method adopts a synthetic and pragmatic approach by selecting 20 features for discriminating real images from deepfakes based on existing literature. An SVM is then trained using these features. With this method, Zhao achieved a detection rate exceeding 90% with fewer than 10,000 images[1]. However, given the sometimes questionable methodology of this approach and the choice of features post-testing, lower results are expected, likely around 75%, given the more realistic nature of the deepfakes used in our study. Additional details on this method can be found in the literature review of this document.

2.3.1.2 Results

Figure 96 summarizes the scores obtained by applying this method to each dataset.

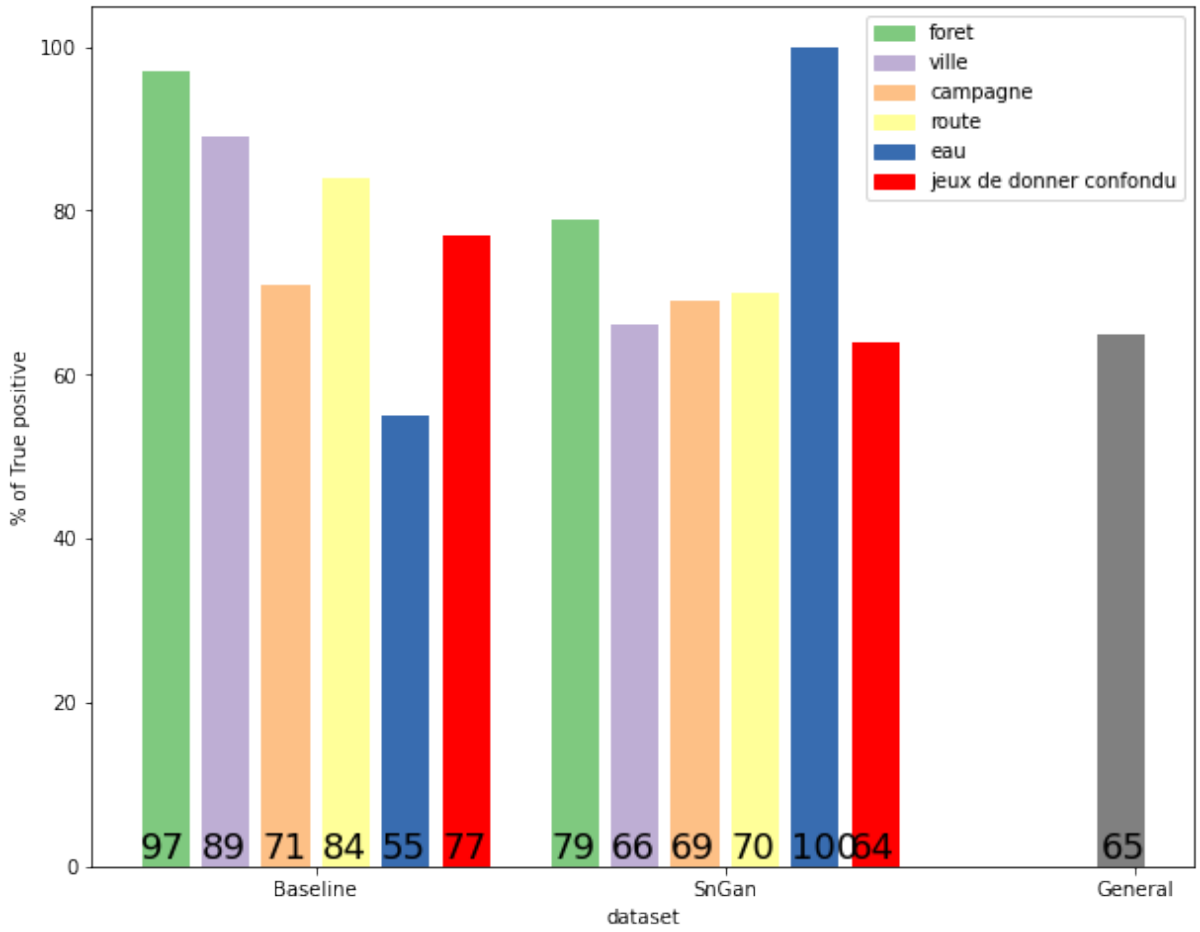


Figure 96: Quantitative results of Zhao's method(Green is forest, purple is urban, brown is rural, yellow is road, blue is sea, red is a mix)

2.3.1.3 Discussion

The results are, in some cases, in line with expectations. As expected, the detection of the SN-GAN water dataset is perfect, as there was a total failure in producing credible deepfakes. For the baseline, the most credible datasets, like water and countryside, are more difficult to detect, whereas city and forest, intuitively easier, are better detected. The overall results for a model are slightly below the average of the individual datasets, which is consistent.

However, some results are counterintuitive. Indeed, the baseline is generally better detected than the GAN, which is surprising because the approach is adapted for GANs. However, fake images are less realistic and more singular than real ones. Additionally, the frequency domain is sensitive to the low diversity of elements, allowing discrimination between real and fake. This is particularly true in the forest environment, precisely the best detected. About the performance on SN-GAN produced deepfakes : the forest is well detected, even though it was intuitively one of the most realistic. Conversely, the city, easily identifiable by a human,

is the least well detected in the GAN model. One limitation of this approach has thus been revealed: the meaning of an image is not considered. Overall this results also shows that the selected GAN is much more robust to detection than the baseline.

In conclusion, Zhao's method is not sufficient to predict deepfakes. Although a score of 0.65 is better than a random chance, it is still insufficient. Furthermore, the overall score is improved thanks to images produced with the baseline. Thus, no precise analysis of the impact of features has been performed. Its unrealistic approach also makes it challenging to apply. These results will serve as a benchmark for the future. One possibility for improvement would be to add a feature that characterizes the sharpness of the image to facilitate discrimination in urban environments.

2.3.2 Fourier Spectrum

As seen in the literature review, GANs and their upsampling network leave visible traces in the Fourier spectrum. This section will explore this approach.

The Fourier spectrum is a mathematical representation that allows the decomposition of an image into frequencies. For this, a discrete Fourier transform is used. In implementations and literature, only the magnitude (i.e., the real part) is considered relevant and retained. Indeed, this part contains most of the geometric helpful information for deepfake detection.

However, this method has some limitations. Like the previous method, it only works on archaic GANs. Even the literature acknowledges that some GANs do not produce artifacts. Moreover, several images were transformed, but none of the patterns described in the literature were observed. Indeed, such artifacts are caused by repeating characteristic patterns like upsampling artifacts. However, the models tested here are deeper, reducing the number of upsampling artifacts. The image on the right in Figure 97 shows the artifacts observed in the literature, while the image on the left shows the Fourier spectrum observed on a fake image. The center image is real.

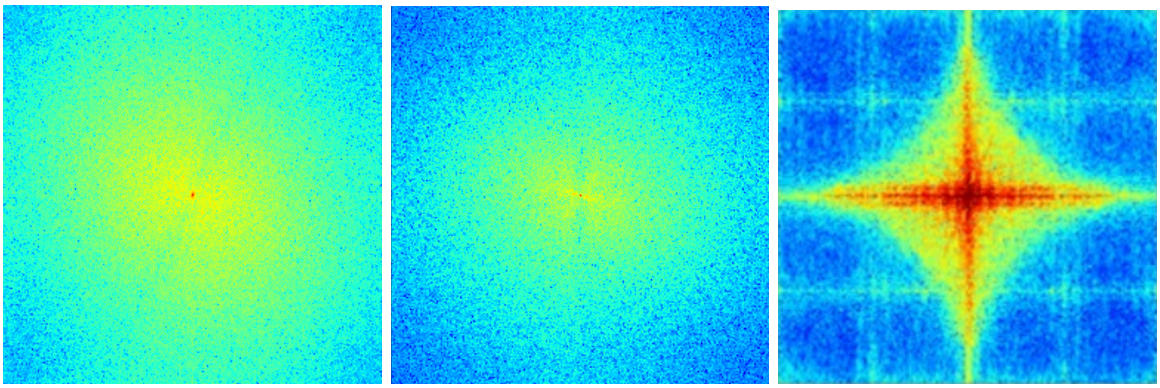


Figure 97: Fourier Spectrum, left: spectrum of a fake image, center: spectrum of a real image, right: typical spectrum of a fake image in the literature[11]

2.3.2.1 Method

However, some observations could be made. Real images have higher frequencies, reflected in the Fourier spectrum by a more vigorous intensity away from the center. This is visible in the previous images. This observation has also been documented in the literature[63][44]. Thus, some approaches have been developed to exploit this feature. Two groups independently implemented the idea. However, the one developed in the great state of Texas [44] is much more elegant than the other [63] as it integrates over θ. Both approaches, however, served as inspiration. The principle is to reduce to a discriminant representation in 2 dimensions. This involves calculating the normalized azimuthal mean for each frequency and then using this data for each image to train an SVM.

2.3.2.2 Results

The training results are presented in Figure 98. Figure 99 shows the mean and standard deviation of real and fake images after processing on the rural dataset.

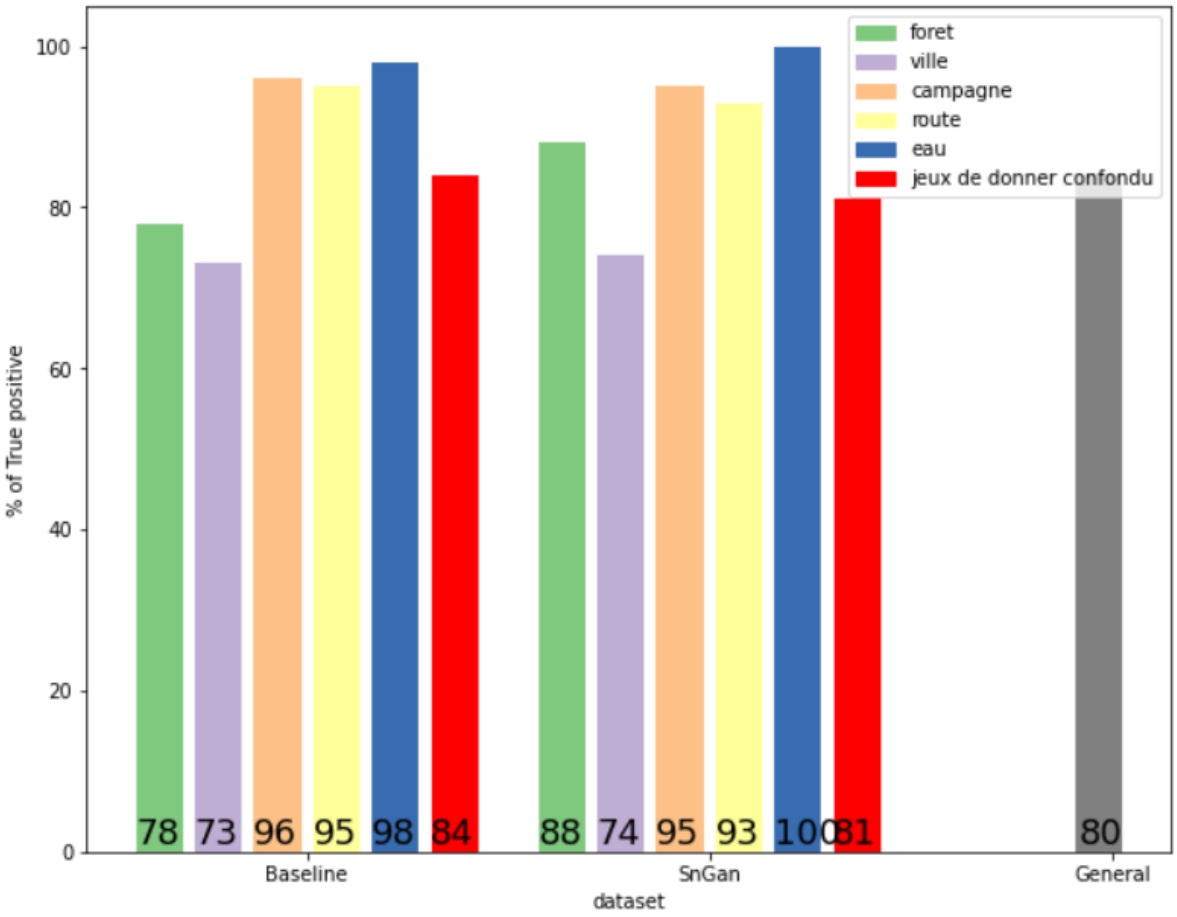


Figure 98: Quantitative results of the Fourier method

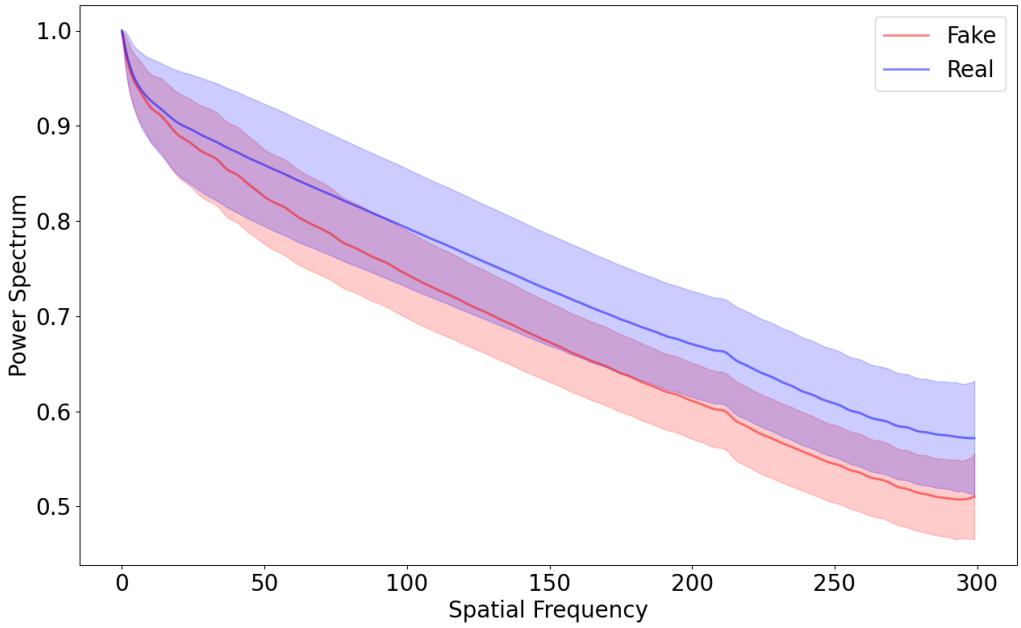


Figure 99: Power of different frequencies for images from the forest dataset

2.3.2.3 Discussion

Figure 99 clearly illustrates that it is possible to distinguish fake images from real ones at high frequency. However, there is an overlap in confidence intervals, explaining why the performance does not exceed 88% for this dataset.

The quantitative results obtained with the Fourier spectrum-based approach are pretty good, given the simplicity of the method. Indeed, they are as good as those obtained in the literature[11]. Furthermore, these results surpass those obtained previously with Zhao's approach. Though challenging, the water dataset from the baseline is well detected with this method. In contrast, the urban environment of the SN-GAN model, intuitively easily identifiable, has the worst performance. Once again, it is worth noting that deepfakes geography are more complicated to detect than images produced with the baseline. This may be explained by the low diversity of elements with the PatchMatch method. However, the results are insufficient, so other methods have been tested.

2.3.3 Deep Method: F3Net

As mentioned in the literature review, F3Net is the most popular deepfake detection method identified in the literature.

2.3.3.1 Method

Indeed, the article and the associated GitHub repository[64] have a relatively high popularity compared to other research on the subject. This implementation adhered to the original article and made no significant modifications.

The principle of this approach involves providing an Xception-type residual network with an image enriched with multiple features. The image is input multiple times with pre-processing to highlight modifications, such as high-pass and low-pass filters and the previously discussed ELA method. Then, supervised training is conducted with this enriched dataset using the provided pre-trained model. In the literature, on high-quality datasets, results exceed 0.95 precision, and similar results are expected for our study.

2.3.3.2 Results

The results of this approach are much better than those of previous approaches. However, training showed significant instability. Models quickly plateaued, i.e., within less than 10 hours. Subsequently, they exhibited highly oscillating performances, indicating an inability to converge, sometimes even regressing significantly, meaning the model remained stuck in a local minimum for a long time. Individual results may not represent the method's performance on the entire dataset. An average was calculated over the last 10 epochs (Figure 100). Figure 101 represents the square root of the variance over the last 10 epochs, well after the plateau.

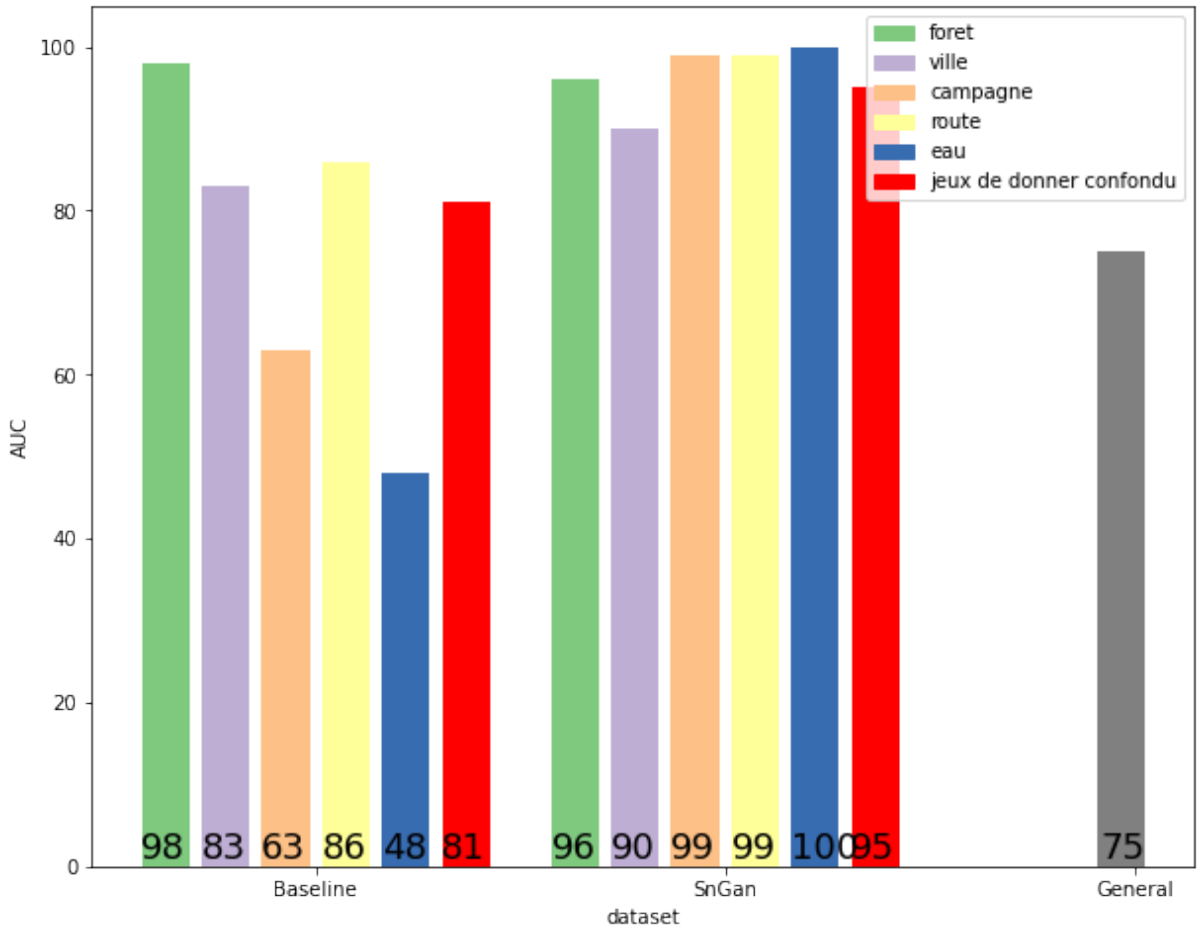


Figure 100: Quantitative result: Average AUC of the F3Net method over the last 10 epochs by dataset

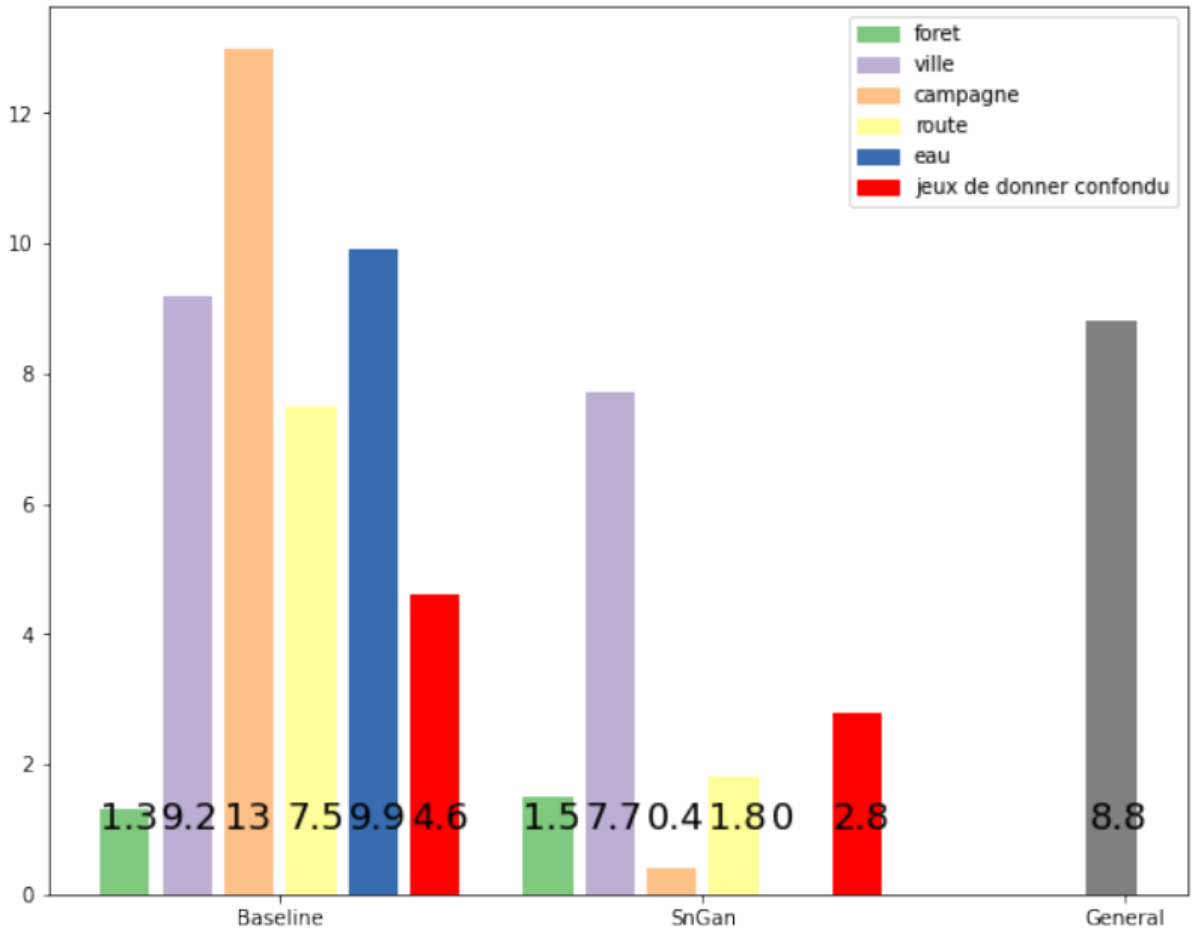


Figure 101: Average standard deviation of the F3Net method over the last 10 epochs by dataset

2.3.3.3 Discussion

The results obtained with the F3Net approach are very positive, with an AUC exceeding 0.99 in some GAN cases, making this method more realistic for practical application. However, the lack of stability in the baseline datasets is a significant problem. Indeed, the precision standard deviation can reach 0.13 after the plateau, making it difficult to guarantee its practical functionality and preventing meaningful discussion of the results. The high dimensionality of the input space could explain this instability. However, the results have much less variance on the GAN datasets. Moreover, the GAN is better detected than images produced with the baseline with this method in general. This can be explained by the model's specialization in deepfake detection, which the baseline lacks. Overall, the obtained results are not as good as those in the literature[64].

One possibility to stabilize the model would be to parallelize multiple training sessions and average the predictions. However, the goal is not achieved. For this reason, a qualitative

analysis of the features was not performed.

2.3.4 Conclusion

The classical approaches studied in this section do not yield sufficient results or are too unstable. Moreover, some methods lacking flexibility are obsolete, and their performance is expected to diminish as deepfakes become more sophisticated. The objective set in the introduction of this chapter has not been achieved. However, the hypothesis that GANs and their deconvolution networks leave a detectable imprint is validated since several traces have been detected, and classical methods have achieved performance well above random chance. However, these traces are not systematic. Despite these encouraging results, none of the tested methods provide sufficient results to achieve the set goal and be applied practically. These methods are also unrealistic regarding the problem chosen to be considered because their prediction is binary.

2.4 Method adapted to "inpainting"

In this section, methods adapted to the detection of "inpainting" deepfakes will be tested, i.e., methods that produce a mask of the fake regions. Complete deepfakes will be used.

2.4.1 Noise

As mentioned in the Introduction, this method involves using noise. Indeed, the image is subtracted from itself but with a transformation reducing the noise, such as SSIM. The literature highlights the ease of discrimination and outstanding scores, with an IoU ratio of 0.95. An implementation was therefore considered.

However, this method was quickly discarded as no preliminary results of similar qualitative level were observed. Figure 102 shows the result obtained with the same kernel used in the literature. This approach can only work for GANs that generate entirely blurred regions, such as U-Net, which is not the case here. The literature is somewhat dishonest in this regard, as it focuses on old GANs, exaggerating the power of the proposed method. Deep approaches based on this idea were therefore not tested on these datasets.

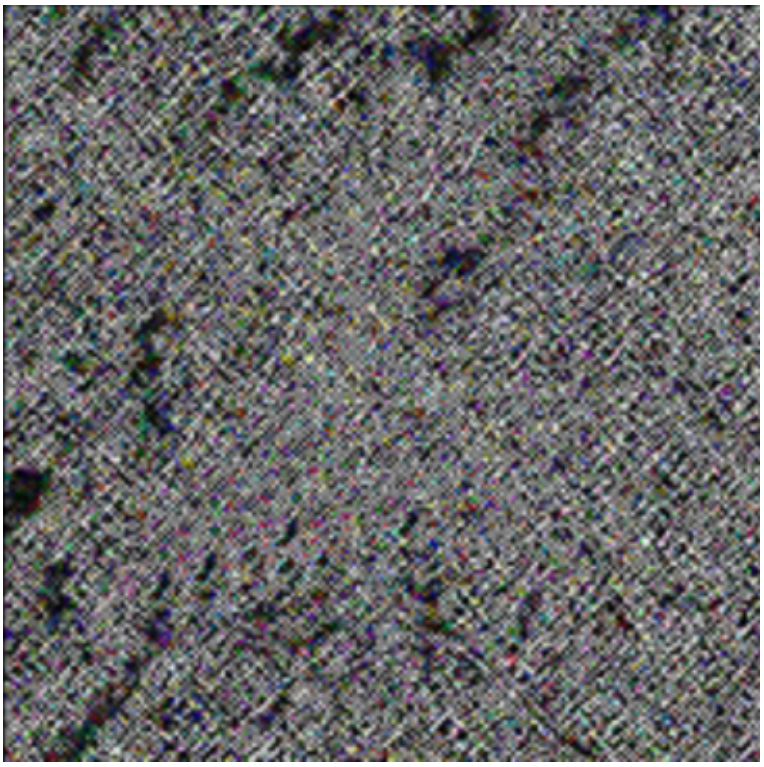


Figure 102: Example of the method supposed to highlight modified areas on an easily manipulated image with a blurred area. The kernel used here is the same as that proposed in the article.

2.4.2 IID Method and Mantra-Net

Only two specific deep learning models for detecting "inpainting" deepfakes, not based on noise, were found. These methods are IID-Net and Mantra-Net.

IID-Net was deemed unusable for our problem due to a total lack of stability. Despite all attempts to correct this flaw, it remained insurmountable. Additionally, given its relatively low popularity compared to other tested methods, this method was abandoned.

For Mantra-Net, no implementation was shared by the article's author. Furthermore, the article does not provide enough details for a faithful implementation. Only the pre-trained model has been shared, but it does not yield usable results with our geographic data. Thus, considering the low popularity of these two models and the problems they encountered, it was decided to continue the investigation.

2.4.3 Synthesis

Research on detecting "inpainting" deepfakes is limited. The few functional studies conducted are already obsolete. In the next section, an original method for detecting deepfakes geography, specialized in "inpainting" deepfake detection, will be explored. This method is

hoped to be more elegant, stable, and practical. The methods tested in this chapter will serve as a point of comparison and inspiration.

2.5 Development of an Original Method for Detecting deepfakes geography

What emerges from the previous tests is that the produced deepfakes are not detected well enough. Moreover, no open-source model can detect "inpainting" deepfakes in general. Indeed, the literature is limited in this regard. Therefore, implementing an original method for detecting "inpainting" deepfakes is necessary.

The goal was then to adapt the best model for whole-image deepfake detection to the detection of falsified regions. The first observation was that F3net, with its Xception architecture, showed excellent results for binary detection of deepfakes geography in previous tests. It was, therefore, chosen to attempt to adapt this network to segmentation and thus "inpainting" deepfake detection. The detection method was developed by adding deconvolution layers to enable the prediction of a potentially fake region mask.

Deepfake detection methods are segmentation methods adapted to delimit modified areas. The first step was to investigate whether a segmentation method like Xception had already been used for aerial images. Recently, Xception has indeed been used for satellite image segmentation, specifically building footprints[65]. Thus, the experience shows that this architecture works in the geospatial context and yields good segmentation results[65]. This suggests that this architecture can effectively detect deepfakes geography.

2.5.1 Method

In this section, the proposed method for detecting "inpainting" deepfakes will be detailed. An existing segmentation method based on the Xception architecture will be re-implemented.

2.5.1.1 Source Implementation

This work is original in that it adapts an existing method to an unexplored problem. However, the previously mentioned work[65] did not provide source code. Therefore, the presented implementation draws significant inspiration from an Apache 2.0 licensed work[66]. This work uses an Xception architecture adapted to segmentation through hybridization with U-Net, as introduced earlier. This notebook was developed on July 28, 2019, for a Kaggle competition on medical image segmentation, specifically for pneumothorax detection.

2.5.1.2 New Implementation

From this notebook, an implementation was carried out by improving the network's depth. In particular, the latent space dimension was increased due to the task's difficulty. As mentioned earlier, the architecture used is an Xception U-Net model, and Figure 103 illustrates the implemented architecture. For this implementation, Keras was used. In the diagram, blue elements represent standard convolution, red elements represent separable convolution (Figure 19), and yellow represents multiple separable convolutions. Finally, green elements represent modules of standard convolution followed by residual convolutions. The depth is the same as a standard Xception network, i.e., the latent space has a size of 19x19x728. The training was conducted conventionally. The images were not augmented as in the original work.

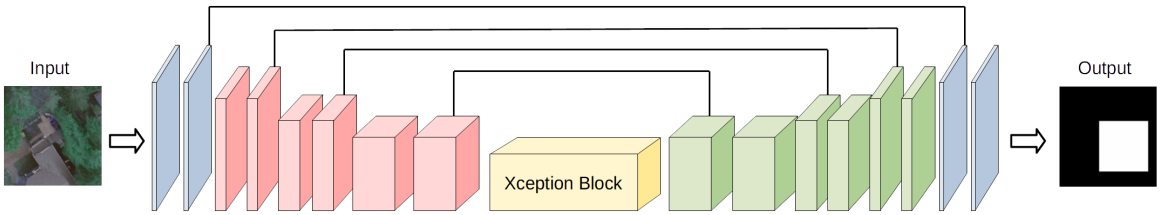


Figure 103: Architecture diagram of the implemented method

Unlike the previous section, the datasets used will contain the entire image associated with a mask of the potentially fake region. An empty mask was created for real images. As in the previous section, there will be a dataset for each different context for each model, as well as a multi-context dataset per model and a multi-context multi-agent dataset. The number of images is the same as detailed in the first section of this chapter. The data was split into an 80% training dataset and a 20% test dataset, which was used to measure performance and was not used for training.

Performance evaluation is carried out using a metric different from the one used previously. This metric has already been presented in the Introduction, section 0.2.3 (Formula 3), and it is the Intersection over Union (IoU) method. It is suitable for the segmentation problem and, therefore, for "inpainting" deepfake detection. It involves a ratio of the identified fake area to the total predicted area. The ratio is between 0 and 1, where 1 represents the best possible result, indicating that the entire predicted area is indeed fake, and 0 represents the worst result, indicating that the entire predicted area is real.

2.5.2 Results

This section will detail the obtained results. The previously described implementation was applied to each deepfake dataset. First, quantitative results will be presented, followed by qualitative results.

2.5.2.1 Quantitative Results

Figure 104 represents the IoU scores obtained after training on the test set, consisting of 2000 images. The training was very stable and converged quickly in only 10 epochs, which is less than 10 hours on the previously presented hardware. The results have been rounded to the nearest tenth.

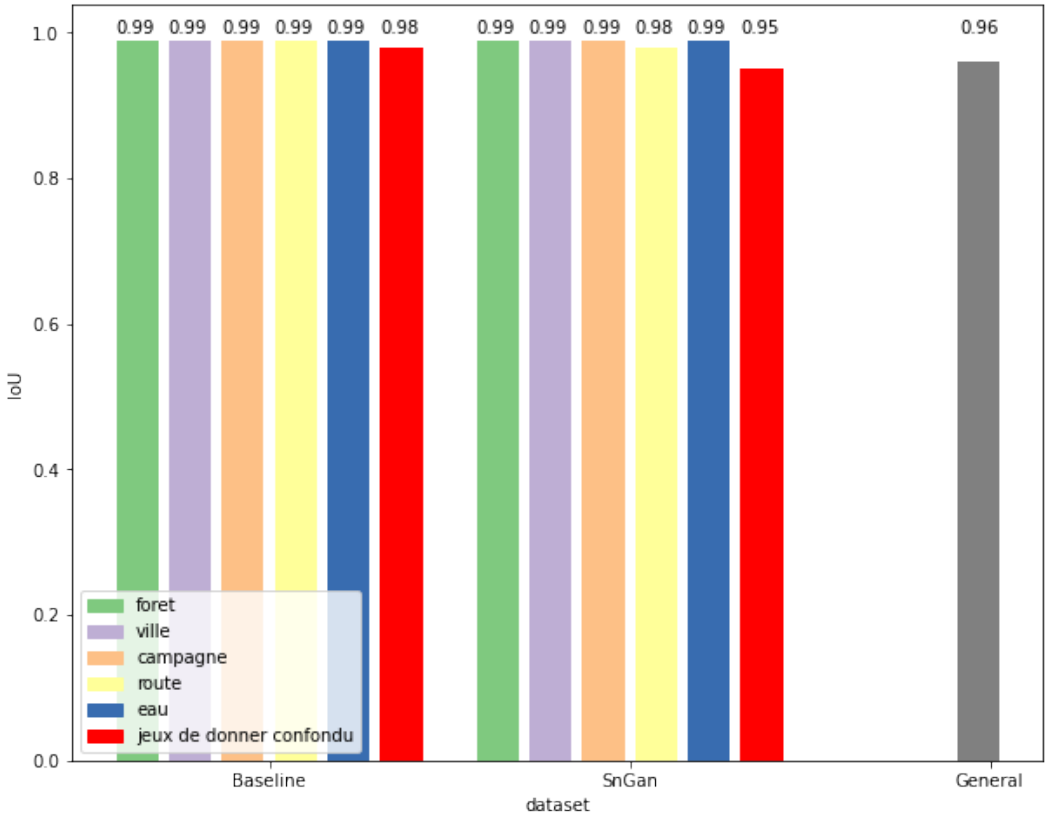


Figure 104: Results obtained on the test dataset

2.5.2.2 Qualitative Results

Figures 105 and 106 represent qualitative results obtained on the test set after training on the urban dataset. The input images are presented on the left, and the produced mask is on the right. It perfectly matches the true mask of the fake area.

Figure 105: Qualitative results on the urban dataset

Figure 106: Other qualitative results on the urban dataset

2.5.3 Discussion

The obtained results are excellent but show slightly lower performance on the multi-context dataset, which may be due to the lack of data for generalization and the greater difficulty in generalizing. The baseline is more easily detected than the SN-GAN model. The worst result corresponds to the multi-environment dataset of the GAN model. Unlike F3Net, the training is stable. However, all fake zones are rectangular, making it easier for the detection algorithm. The fake areas are also significant compared to the image. Thus, minor errors have a relatively low impact.

This result is considered very good in the literature for object identification tasks. Indeed, a quick qualitative analysis concludes that all fake areas are well identified. For comparison, in March 2022, the IID-Net network obtained a PixelWise F1 score (also called

Dice coefficient) between 75% and 94% on lower-quality "inpainting" deepfakes[67]. The Dice score is given by the formula $Dice = \frac{2\times|A\cap B|}{|A|+|B|}$. Dice and IoU scores are very close, as shown by the demonstration (2.1). However, since IoU is between 0 and 1, Dice is necessarily greater than or equal to it. Even if close to 1, the difference is minimal. The proof by contradiction can be found later (2.2). IID-Net is still surpassed. Thus, the set objective has been achieved with stability and satisfactory results in a realistic context.

$$
\begin{aligned}
IoU &= \frac{|A \cap B|}{|A \cup B|} \\
&= \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \\
&= \frac{2|A \cap B|}{2(|A| + |B| - |A \cap B|)} \\
Dice &= \frac{2|A \cap B|}{|A| + |B|} \\
&= \frac{IoU \times 2(|A| + |B| - |A \cap B|)}{|A| + |B|} \\
&= \frac{IoU \times 2(|A| + |B| - |A \cap B|)}{|A| + |B| - |A \cap B|} \times \frac{|A| + |B| - |A \cap B|}{|A| + |B|} \\
&= \frac{2 \times IoU}{\frac{|A|+|B|-|A\cap B|+|A\cap B|}{|A|+|B|-|A\cap B|}} \\
&= \frac{2 \times IoU}{1 + \frac{|A\cap B|}{|A|+|B|-|A\cap B|}} \\
&= \frac{2 \times IoU}{IoU + 1}
\end{aligned}
\tag{2.1}
$$

$$
\begin{aligned}
IoU &> Dice \\
IoU &> \frac{2 \times IoU}{IoU + 1} \\
IoU \times (IoU + 1) &> 2 \times IoU \\
IoU + 1 &> 2
\end{aligned}
\tag{2.2}
$$

Intuitively, the results of fake image zone detection could potentially have been worse than those for the more straightforward task of binary identification of fake images without enrichment. However, in the case of high-quality deepfakes, it is not uncommon to encounter images that are challenging to detect. Typical GAN artifacts observed in the literature were rarely present in these data. Thus, seeking a boundary between fake and real makes more sense for identifying fake zones. There could be small shifts, imperceptible sharpness differences to the naked eye, or lost meaning, which are easier to detect. Moreover, after creating deepfakes,

qualitative analysis primarily focused on the border areas. The improvement in stability is likely also due to the reduction in input dimension, reducing the complexity of the problem.

However, it would be interesting to test this approach with enriched input data, but this would not allow for measuring an improvement in results since the IoU score is already almost always above 0.99. Similarly, testing with a more advanced architecture, such as EfficientNet, would not allow conclusions about the superiority of one method over another. There is no need to delve further into these experiments. However, as mentioned in the state of the art, EfficientNet is known for its superior performance in segmentation tasks. A test on non-rectangular masks could be performed, but it would require complete retraining, which is not feasible with the available computing resources and time. Nevertheless, tests of the method's robustness will be conducted.

2.6 Robustness of the Original Method

To test the robustness of the previously trained model, a reduction in image quality was performed by applying Gaussian blur to the images. The goal was to make the boundary more discreet by blurring the image. For this, a Gaussian blur was applied to the images. The first test was performed with a kernel of 5 and a σ of 1. The second test was performed with a kernel of 9 and a σ of 3. Figure 107 shows an example of the blur effect during the second test, with the original image on the left and the blurred image on the right.

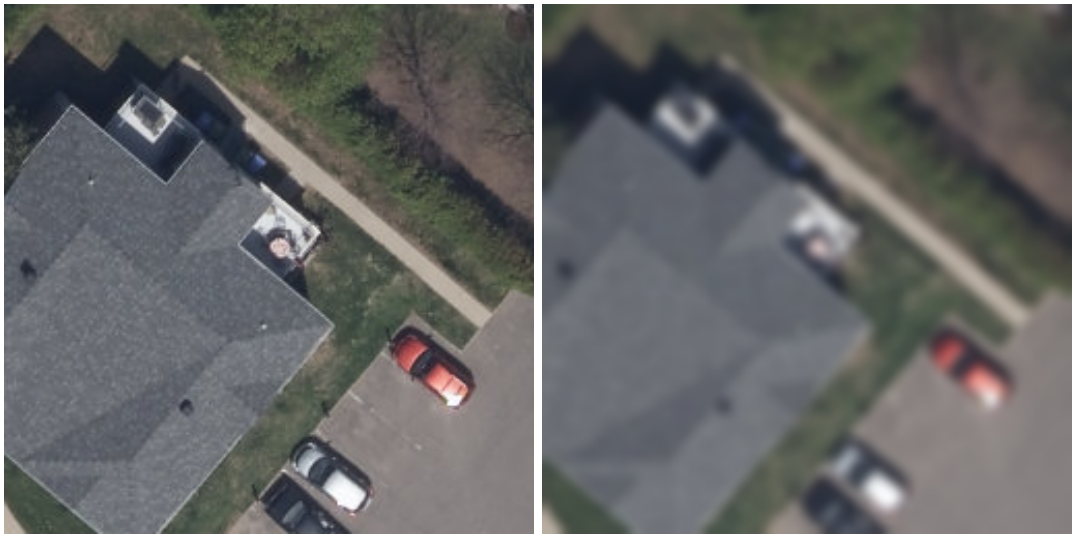


Figure 107: Effect of the most blurring filter applied to the images

Training was then carried out with a control group on the same images but without being modified by the filter. The trainings were performed on the SN general dataset. The quantitative results are presented in Figure 108. In both cases, the performances are hindered.

Finally, qualitative results are presented in Figures 109 and 110. In these figures, the left image corresponds to the input, the middle one is the mask of the fake area, and the right image is the prediction. Doubtful areas were generally identified, but the boundary was challenging to identify. Indeed, the model fails to predict a straight line, showing its hesitation. No detection was observed on real images.

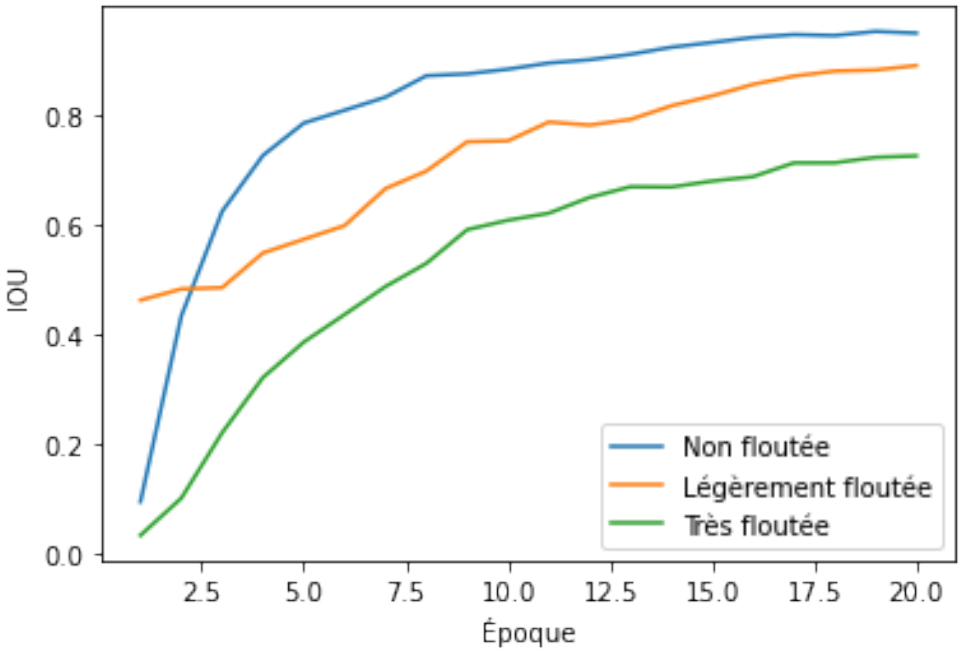


Figure 108: IoU obtained during training for each experimentation(Blue = No blurry, Orange = Little bit blurry, Green = Very blurry)

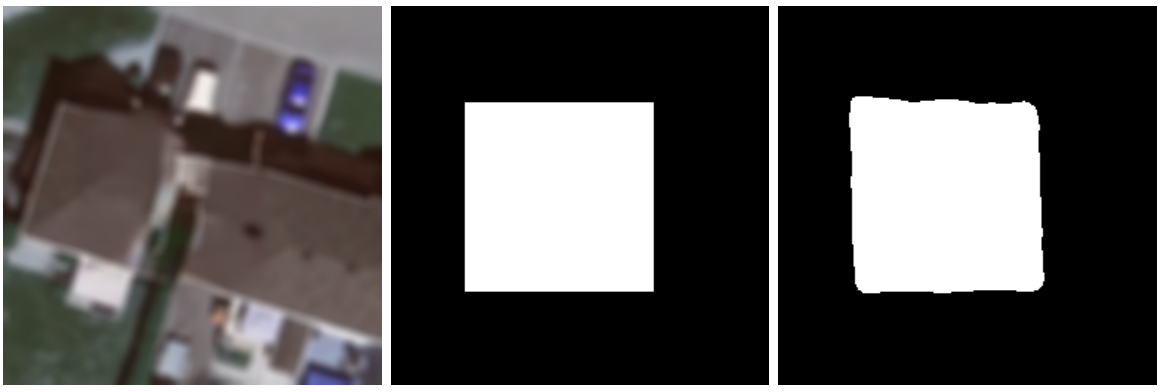


Figure 109: Example of qualitative result on urban images

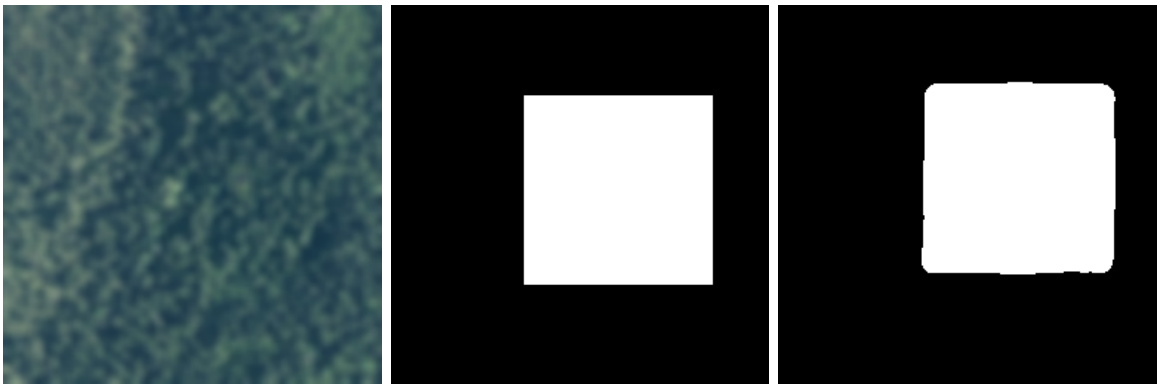


Figure 110: Example of qualitative result on forest images

This experiment demonstrates that the proposed model does not focus only on borders and technical points, as they are blurred. It shows that the model truly understands the meaning of images.

2.7 Conclusion

The targeted result has been achieved here, as the IoU score is consistently above 0.94, and the deepfakes have been detected stably, with low over-detection. These results surpass expectations, and it would be challenging to improve further. Better-quality deepfakes would have been required to go beyond.

Thus, an original method has been proposed, outperforming existing methods by better addressing the issue compared to classical methods. It is also more elegant and versatile due to its simplicity and excellent stability compared to IID-Net and F3Net. Moreover, this method will be offered as open source, facilitating its use by the scientific community and the public. It is worth noting that simple tests have confirmed good robustness. Furthermore, there are still opportunities for improvement. An interesting perspective would be to replace the Xception network with EfficientNet, which has shown better results in the literature for segmentation tasks.

Conclusion and Perspectives

This work has explored a new research area: the generation and detection of deepfakes in the context of aerial images. Initially, high-quality deepfakes geography, invisible to the human eye, were successfully produced. The deepfakes effectively concealed significant areas in authentic images, particularly in high-resolution images with intricate details. However, deepfakes produced in an urban context were slightly less realistic due to the higher resolution and increased color and element diversity in the urban dataset.

Subsequently, attempts were made to detect deepfakes using existing methods. Some classical methods showed interesting results but were unsuitable for our problem. Approaches specific to detecting inpainting deepfakes proved to be impractical or outdated.

Finally, this research has introduced an original approach for deepfake detection based on a segmentation model. This method was implemented and tested on various datasets, yielding excellent results. The proposed method outperforms existing methods while being more straightforward and more elegant. The results demonstrated high stability and were achieved in just a few hours of training. Additionally, experiments showed that the proposed original model is robust and comprehends the context of the images. In conclusion, this deepfake detection method holds great promise in combating the dissemination of falsified content online and in the media. This model can detect false geospatial data, contributing to the defense against geospatial fake news and, consequently, in geomatics.

GAN models can generate credible deepfakes geography within aerial images, which can be detected using the implemented original method. Therefore, there is no need to regulate the dissemination of fake images, especially since means are available for their detection. Moreover, thanks to this work, propagandist governments will find it challenging to employ deepfakes geography for their political and strategic objectives.

This work opens up several avenues for further research. The detection of deepfakes is constantly in competition with generation models. To anticipate the next generations of deepfakes geography, implementing original generation models or enhancing existing ones would be necessary. Several avenues could be explored to improve the creation phase.

- Initially, integration with deep learning could be performed to address the lack of semantic understanding of the baseline. For example, applying patch search in a similar semantic area would be possible. The Context Encoder could be used to predict a semantic segmentation of the inpainting area, and then the PatchMatch method would be applied only locally in areas of similar semantics.

- It would also be interesting to add the result of PatchMatch in the input layers for the baseline. However, training would be significantly slowed down as PatchMatch is a resource-intensive process on thousands of images. The PatchMatch principle could also be used as a loss function.

- Improvements for deep models involve the number of data used, training duration, and model depth. In the literature, the best results are obtained with millions of images, several weeks of training, and smaller images, implying a relatively higher model depth. Using inpainting-specific loss functions implemented in the Context Encoder could improve results. Another example would be a radial factor loss function relative to the inpainting area.

- It would also be interesting to explore the addition of other types of geographical information, such as land use, which could be determined using a U-Net[68]. Loss functions or discriminators specific to geographical data could also be implemented, for example, using the building detection algorithm developed and pretrained at CRDIG as a style function. Specific post-processing for sharpness and borders or non-straight lines could also significantly improve, especially since the artifacts are generally similar.

- In the case of SN-GAN, hybridization with EdgeConnect could yield better results. In particular, EdgeConnect with partial convolution. That becaus EdgeConnect has performed much better than GL-GAN. Moreover, the artifacts observed in EdgeConnect, being typical of deconvolution network defects, replacing these networks with convolutions resolving these issues would be interesting to test[69]. Furthermore, guidance with geographical data would result in straighter shapes, even if this might lead to some loss of sharpness, as shown by the EdgeConnect method[8]. A significant modification of the weight of loss functions would be required for highly uniform environments, such as the aquatic environment.

- Since the start of the research, new and highly promising generation algorithms have been developed, including the Palette diffusion method[40], as well as LAMA Cleaner[70]. This recently published solution has garnered much attention for its highly satisfactory empirical results. Due to time constraints, other methods whose testing would have been desirable, [71][72] have yet to be analyzed.

Regarding detection methods, several possibilities exist.

- Further adversarial attacks could be attempted to document the flaws of the implemented method more comprehensively.
- Enriching inputs with explored detection methods could improve results like F3Net.
- Since the start of the research, a zone prediction method applied to deepfakes has been successfully tested using the EfficientNet architecture[73]. The results obtained are similar to those found independently with the original approach. It would be interesting to test this method on our data.
- Test unsupervised method which is more suitable for real-world conditions.

Finally, to enable widespread use by the general public, a user-friendly application needs to be developed. Doing so in a decentralized environment, as proposed by Adobe[74] and Stanford[75], would be a plus. Reflection is also needed to integrate this model into a realistic flow, adapting to the few identified images and evolving with each newly detected deepfake. The parameterization of inpainting would also be an exciting avenue to explore. More general tests on inpainting deepfakes with the original approach would be interesting to conduct. Finally, discussions are underway for the release of the deepfakes and datasets produced to create a benchmark dataset for future research. Testing in a realistic and critical environment, such as the Russo-Ukrainian conflict with the omnipresence of strategic aerial images, would be another avenue to explore.

Bibliography

[1] Bo Zhao, Shaozeng Zhang, Chunxue Xu, Sun Yifan, and Chengbin Deng. Deep fake geography? when geospatial data encounter artificial intelligence. pages 338–352, 04 2021. doi: 10.1080/15230406.2021.1910075.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *in 2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, oct. 2017. doi: 10.1109/ICCV.2017.244.

[3] M. Massi. English : Autoencoder schema, 2019. URL https://commons.wikimedia.org/wiki/File:Autoencoder_schema.png. Consulté le : déc. 09, 2021. [En ligne].

[4] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting, 2016. Consulté le : déc. 09, 2021. [En ligne].

[5] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36:107:1–107:14, 07 2017. doi: 10.1145/3072959.3073659.

[6] Shenlong Lou, Qiancong Fan, Feng Chen, Cheng Wang, and Jonathan Li. Preliminary investigation on single remote sensing image inpainting through a modified gan. pages 1–6, 08 2018. doi: 10.1109/PRRS.2018.8486163.

[7] Ugur Demir and Gozde Unal. Patch-based image inpainting with generative adversarial networks, mars 2018. Consulté le : déc. 09, 2021. [En ligne].

[8] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning, 2019. Consulté le : déc. 09, 2021. [En ligne].

[9] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement, 2018. Consulté le : déc. 09, 2021. [En ligne].

[10] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions, 2018. Consulté le : déc. 09, 2021. [En ligne].

[11] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images, 2019. Consulté le : déc. 09, 2021. [En ligne].

[12] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016. Consulté le : déc. 09, 2021. [En ligne].

[13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. Consulté le : déc. 09, 2021. [En ligne].

[14] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

[16] Haiwei Wu and Jiantao Zhou. Iid-net: Image inpainting detection network via neural architecture search and attention. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1172–1185, 2022. doi: 10.1109/TCSVT.2021.3075039.

[17] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9535–9544, 2019. doi: 10.1109/CVPR.2019.00977.

[18] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention, 2018.

[19] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Free-form image inpainting with gated convolution, 2019.

[20] Jayson Harsin. Post-truth and critical communication studies, 12 2018. URL https://oxfordre.com/communication/view/10.1093/acrefore/9780190228613.001.0001/acrefore-9780190228613-e-757.

[21] Megha rajagopalan, alison killing and christo buschek of buzzfeed news. URL https://www.pulitzer.org/winners/megha-rajagopalan-alison-killing-and-christo-buschek-buzzfeed-news. Consulté le : déc. 09, 2021. [En ligne].

[22] Shaozeng Zhao, Bo et Zhang. Rethinking spatial data quality: Pokémon go as a case study of location spoofing. *The Professional Geographer*, 71, no 1:96–108, janv. 2019. doi: 10.1080/00330124.2018.1479973.

[23] Mika Westerlund. The emergence of deepfake technology: A review. *Technology Innovation Management Review*, 9:40–53, 11/2019 2019. ISSN 1927-0321. doi: http://doi.org/10.22215/timreview/1282.

[24] T. Tjukanov. Mapdreamer — ai cartography, mai 06 2020. URL https://medium.com/@tjukanov/mapdreamer-ai-cartography-4f2f6a40ef55. Consulté le : déc. 09, 2021. [En ligne].

[25] Qian Shi, Xiaoping Liu, and Xia Li. Road detection from remote sensing images by generative adversarial networks. *IEEE Access*, 6:25486–25494, 2018. doi: 10.1109/ACCESS.2017.2773142.

[26] A. Hindupur. The gan zoo, 2021. URL https://github.com/hindupuravinash/the-gan-zoo. Consulté le : déc. 09, 2021. [En ligne].

[27] R. Cole. Introduction, 2021. URL https://github.com/robmarkcole/satellite-image-deep-learning. Consulté le : déc. 09, 2021. [En ligne].

[28] Praveer Singh and Nikos Komodakis. Cloud-GAN: Cloud Removal for Sentinel-2 Imagery Using a Cyclic Consistent Generative Adversarial Network. In *in IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 1772–1775, Valencia, Spain, juill. 2018. doi: 10.1109/IGARSS.2018.8519033.

[29] Behnood Rasti, Yi Chang, Emanuele Dalsasso, Loic Denis, and Pedram Ghamisi. Image restoration for remote sensing: Overview and toolbox. *IEEE Geoscience and Remote Sensing Magazine*, 10(2):201–230, jun 2022. doi: 10.1109/mgrs.2021.3121761. URL https://doi.org/10.1109%2Fmgrs.2021.3121761.

[30] P. Markuse. It's a faaaake... — or not?, oct. 01, 2019. URL https://medium.com/sentinel-hub/its-a-faaaake-or-not-bace4f0c01ec. Consulté le : déc. 09, 2021. [En ligne].

[31] Shaozeng Zhang, Bo Zhao, and Yuanyuan Tian. Stand with #standingrock: Envisioning an epistemological shift in understanding geospatial big data in the "post-truth" era. *Annals of the American Association of Geographers*, 111:1–21, 08 2020. doi: 10.1080/24694452.2020.1782166.

[32] P. Tucker. The newest ai-enabled weapon: 'deep-faking' photos of the earth, mars 31, 2019. URL https://www.defenseone.com/technology/2019/03/next-phase-ai-deep-faking-whole-world-and-china-ahead/155944/. Consulté le : déc. 09, 2021. [En ligne].

[33] Ian J. Goodfellow et al. Generative adversarial networks, juin 2014. Consulté le : déc. 09, 2021. [En ligne].

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. Consulté le : déc. 09, 2021. [En ligne].

[35] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), aug 2009.

[36] M. Bertalmio, Andrea Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. volume 1, pages I–355, 02 2001. ISBN 0-7695-1272-0. doi: 10.1109/CVPR.2001.990497.

[37] Naoufal Amrani, Joan Serra-Sagristà, Pascal Peter, and Joachim Weickert. Diffusion-based inpainting for coding remote-sensing data. *IEEE Geoscience and Remote Sensing Letters*, PP:1–5, 06 2017. doi: 10.1109/LGRS.2017.2702106.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, déc. 2015. Consulté le : déc. 09, 2021. [En ligne].

[39] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, avr. 2017. Consulté le : déc. 09, 2021. [En ligne].

[40] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2022. Consulté le : déc. 09, 2021. [En ligne].

[41] Ning Yu, Larry Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints, août 2019. Consulté le : déc. 09, 2021. [En ligne].

[42] Javier Galbally and Sébastien Marcel. Face anti-spoofing based on general image quality assessment. pages 1173–1178, 08 2014. doi: 10.1109/ICPR.2014.211.

[43] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition, 2020. Consulté le : déc. 09, 2021. [En ligne].

[44] Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier spectrum discrepancies in deep network generated images, 2020. Consulté le : déc. 09, 2021. [En ligne].

[45] Oliver Giudice, Luca Guarnera, and Sebastiano Battiato. Fighting deepfakes by detecting GAN DCT anomalies. *Journal of Imaging*, 7(8):128, jul 2021. doi: 10.3390/jimaging7080128. URL https://doi.org/10.3390%2Fjimaging7080128.

[46] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions, 2020.

[47] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues, 2020. Consulté le : déc. 09, 2021. [En ligne].

[48] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2017. Consulté le : déc. 09, 2021. [En ligne].

[49] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. Consulté le : déc. 09, 2021. [En ligne].

[50] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. Consulté le : déc. 09, 2021. [En ligne].

[51] Hasty.ai. Intersection over union (iou), 2023. URL https://hasty.ai/docs/mp-wiki/metrics/iou-intersection-over-union. Consulté le : mar. 09, 2023. [En ligne].

[52] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9:23–24, 01 2004. doi: 10.1080/10867651.2004.10487596.

[53] S. B. Damelin and N. S. Hoang. On surface completion and image inpainting by biharmonic functions: Numerical aspects. *International Journal of Mathematics and Mathematical Sciences*, 2018:1–8, 2018. doi: 10.1155/2018/3950312. URL https://doi.org/10.1155%2F2018%2F3950312.

[54] Alexandre Sevin. Patch-match, 2021. URL https://github.com/AlexandreSev/Patch-Match. Consulté le : déc. 09, 2021. [En ligne].

[55] Boyuan Jiang. context_encoder_pytorch, 2017. URL https://github.com/BoyuanJiang/context_encoder_pytorch. Consulté le : déc. 09, 2021. [En ligne].

[56] Erik Linder-Norén. Pytorch-gan, 2019. URL https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/context_encoder/context_encoder.py. Consulté le : déc. 09, 2021. [En ligne].

[57] Du Ang. generative-inpainting-pytorch, 2021. URL https://github.com/daa233/generative-inpainting-pytorch. Consulté le : déc. 09, 2021. [En ligne].

[58] Kamyar Nazeri. edge-connect, 2022. URL https://github.com/knazeri/edge-connect. Consulté le : déc. 09, 2021. [En ligne].

[59] Naoto Inoue. pytorch-inpainting-with-partial-conv, 2020. URL https://github.com/naoto0804/pytorch-inpainting-with-partial-conv. Consulté le : déc. 09, 2021. [En ligne].

[60] Jiahui Yu. generative_inpainting, 2020. URL https://github.com/JiahuiYu/generative_inpainting. Consulté le : déc. 09, 2021. [En ligne].

[61] Hany Farid. Exposing digital forgeries from jpeg ghosts. *IEEE Transactions on Information Forensics and Security*, 4(1):154–160, 2009. doi: 10.1109/TIFS.2008.2012215.

[62] Ian Jolliffe and Jorge Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202, 04 2016. doi: 10.1098/rsta.2015.0202.

[63] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. Unmasking deepfakes with simple features, 2020.

[64] Yike Yuan. F3net, 2022. URL https://github.com/yyk-wew/F3Net. Consulté le : déc. 09, 2022. [En ligne].

[65] Kepeng Xu, Yunye Zhang, Wenxin Yu, Zhiqiang Zhang, Jingwei Lu, Yibo Fan, Gang He, and Zhuo Yang. Segmentation of building footprints with xception and iouloss. *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 420–425, 2019.

[66] Siddhartha. Unet xception keras for pneumothorax segmentation. 2019. URL https://www.kaggle.com/code/meaninglesslives/unet-xception-keras-for-pneumothorax-segmentation. Consulté le : déc. 09, 2021. [En ligne].

[67] Haiwei Wu and Jiantao Zhou. Iid-net: Image inpainting detection network via neural architecture search and attention. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2021. doi: 10.1109/TCSVT.2021.3075039.

[68] Srimannarayana Baratam. Land cover classification with u-net. *Medium*, 2021. URL https://baratam-tarunkumar.medium.com/land-cover-classification-with-u-net-aa618ea64a1b. Consulté le : déc. 09, 2021. [En ligne].

[69] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. doi: 10.1109/CVPR.2016.207.

[70] Sanster Qing. lama-cleaner, 2022. URL https://github.com/Sanster/lama-cleaner. Consulté le : déc. 09, 2022. [En ligne].

[71] Multimedia Research. F3net, 2021. URL https://github.com/researchmm/AOT-GAN-for-Inpainting. Consulté le : déc. 09, 2021. [En ligne].

[72] Haiwei Wu, Jiantao Zhou, and Yuanman Li. Deep generative model for image inpainting with local binary pattern learning and spatial attention, sept. 2020.

[73] Serhat AtaŞ, İsmail İlhan, and Mehmet KarakÖse. An efficient deepfake video detection approach with combination of efficientnet and xception models using deep learning. In *2022 26th International Conference on Information Technology (IT)*, pages 1–4, 2022. doi: 10.1109/IT54280.2022.9743542.

[74] Adobe Communications Team. Introducing the content authenticity initiative, 2019. URL https://blog.adobe.com/en/publish/2019/11/04/content-authenticity-initiative. Consulté le : déc. 09, 2022. [En ligne].

[75] starlinglab. URL https://www.starlinglab.org/. Consulté le : déc. 09, 2022. [En ligne].