Modèle de diffusion frugal pour l'inpainting d'images

Nicolas CHEREL[1] Andrés ALMANSA[2] Yann GOUSSEAU[1] Alasdair NEWSON[1]

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, France

[2]MAP5 & CNRS, Université Paris Cité, France

Résumé – Très récemment, les modèles de diffusion probabilistes ont permis d'améliorer significativement l'état de l'art en synthèse d'images. Ces modèles génératifs se sont avérés particulièrement efficaces pour la résolution de problèmes inverses comme la super-résolution, le défloutage ou l'inpainting. Cependant beaucoup de ces modèles de diffusion reposent sur des centaines de millions de paramètres et le temps d'apprentissage se compte en dizaines de jours-GPU. Ces contraintes matérielles rendent leur utilisation particulièrement lourde et couteuse. Nous proposons un modèle de diffusion léger pour l'inpainting, adapté à des entraînements sur une ou quelques images, ne nécessitant qu'un apprentissage relativement rapide. Nous analysons précisément l'apport de la diffusion, à architecture constante, et nous comparons notre réseau léger avec l'état de l'art dans l'inpainting image.

Abstract – Very recently, probabilistic diffusion models have significantly improved the state of the art for the synthesis of images. These generative models are very efficient for solving inverse problems such as super-resolution, deblurring or inpainting. However, many of these diffusion models have hundreds of millions of parameters and the learning time is in tens of GPU days. These hardware constraints make them especially heavy and costly to use. We propose a lightweight diffusion model for inpainting, suitable for training on one or a few images, and requiring relatively little learning.

1 Introduction et contexte

Les modèles génératifs reposant sur l'apprentissage profond ont connu un essor considérable dans les dernières années, depuis l'introduction des "Generative Adversarial Network" (GAN) de Goodfellow *et al*. [1]. Par la suite, les modèles de diffusion [2] ont surpassé ces approches, et se sont imposés comme étant l'état de l'art pour la synthèse et l'édition des images. En effet, ces modèles conduisent à des résultats souvent de meilleure qualité qu'avec des GANs, avec un entraînement plus stable. Néanmoins, ces modèles de diffusion sont massifs, comprennent possiblement jusqu'à plusieurs centaines de millions de paramètres et requièrent des ressources computationnelles énormes, rendant cet entraînement impossible pour la majorité des utilisateurs et discutable d'un point de vue environnemental.

Par ailleurs, les *a priori* d'images trouvés implicitement par modèles génératifs ont été exploités pour la résolution de problèmes inverses [7]. Un exemple de problème inverse est l'inpainting d'image, qui consiste à remplir une zone inconnue d'une image de manière visuellement convaincante. Dans ce papier, nous nous intéressons à l'utilisation des modèles de diffusion pour l'inpainting d'image. En contraste avec l'état de l'art qui propose des architectures massives, nous proposons des modèles de taille très réduite. Ces modèles reposent uniquement sur des opérations de convolution, de sous/sur-échantillonnage et de non-linéarités, contrairement à la plupart des modèles de diffusion qui font intervenir d'autres modules plus complexes, par exemple des modules d'attention. Notre but est ainsi de proposer et d'étudier des modèles "frugaux" de diffusion pour l'inpainting afin de mieux comprendre leur fonctionnement et de permettre leur utilisation sans avoir recours à des ressources computationnelles exorbitantes.

Plus précisément, nos contributions sont les suivantes. Premièrement, nous proposons une architecture de modèle de diffusion de taille réduite qui est compétitive avec l'état de l'art sur certains types d'images. Nous montrons qu'il est possible avec peu de modifications d'utiliser la même architecture pour la résolution de problèmes plus simples avec un entraînement très rapide. Nous étudions précisément l'apport de la diffusion par rapport à une approche standard d'apprentissage profond, à architecture de réseau fixée. Enfin, nous comparons notre méthode avec l'état de l'art par diffusion en termes de ressources computationnelles et d'impact écologique ;

Diffusion Les modèles de diffusion pour la génération d'image ont été introduits par Sohl-Dickstein *et al*. [9] en 2015. Ho *et al*. [2] ont popularisé la méthode en reformulant le problème en un problème de débruitage. De nombreuses extensions ont été proposées pour la génération conditionnelle à partir de texte, ou pour la résolution de problèmes inverses [7]. Le coût matériel et des temps d'apprentissage longs ont poussé la recherche vers des méthodes d'entraînement plus efficaces. Rombach *et al*. [6] réalisent l'apprentissage d'un modèle de diffusion dans un espace latent plus petit que l'espace image.

Inpainting Les approches classiques d'inpainting par patchs [10, 4] s'appuient sur l'auto-similarité des images pour construire une solution vraisemblable. Depuis 2016, des méthodes par réseaux de neurones ont utilisé de larges bases de données d'images pour apprendre à résoudre ce problème [5]. Yu *et al*. [11] intègrent une couche d'attention pour intégrer une composante non-locale dans leur réseau autrement purement convolutif. Très récemment, des méthodes reposant sur des modèles de diffusion sont apparues, très gourmandes en ressources, telles Repaint [3] et Palette [7].

2 Méthode

Nous appelons x l'image de référence, et M le masque qui indique la zone à remplir, égal à 1 dans la zone, et 0 partout ailleurs. Soit $y = x \odot M$ l'image masquée (et donc à remplir), où $\odot$ représente la multiplication élément par élément.

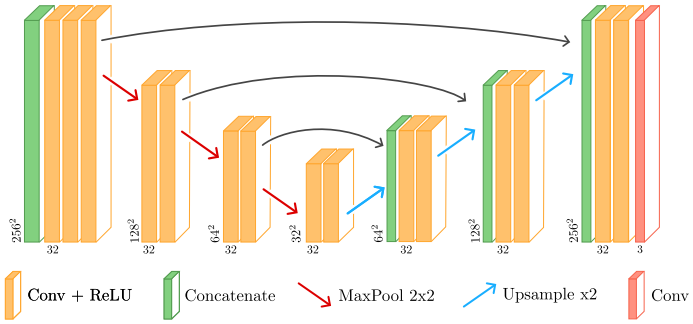

FIGURE 1 : Architecture du réseau UNet

2.1 Modèles de diffusion pour l'inpainting

Les modèles de diffusion [2] sont des modèles génératifs dont le but est généralement d'apprendre une distribution d'images naturelles. Ils s'appuient sur un processus *forward* destructif et un processus *backward* génératif. Le processus forward consiste à transformer une image d'entrée, qui est supposée être un échantillon de la distribution des images naturelles, en une image de bruit blanc, par l'ajout de bruit de faible amplitude pendant un certain nombre d'itérations. Le processus backward effectue les étapes inverses : des *débruitages* (appris) itératifs d'un échantillon initialement tiré d'un bruit gaussien. Si le débruiteur est appris correctement, il permet de passer d'un bruit blanc à un exemple d'image. On considère que l'image de référence correspond à l'image au "temps" $t = 0$, et l'image finale bruitée à celle au temps $t = T$. Ainsi, on établit une succession d'images $\tilde{x}_t$ avec un niveau de bruit qui augmente avec le temps.

Entraînement Pour l'inpainting, on entraîne le débruiteur à passer d'une image $\tilde{x}_t$ (pour un temps t) à l'image de référence x. Plus précisément, on minimise l'erreur suivante :

$$\mathcal{L}_\theta(x) = \|x - f_\theta(\tilde{x}_t, y, M, \sigma_t^2)\|^2 \quad (1)$$

où f_θ représente un réseau de neurones qui effectue le débruitage. L'apprentissage est fait pour tous les niveaux de bruit, couvrant ainsi l'intervalle de variances $\sigma_0^2 = 0 < ... < \sigma_T^2 = 1$. Le paramètre t contrôle le niveau de bruit ajouté à l'image x pour obtenir $\tilde{x}_t$. Nous notons que f_θ prend en entrée à la fois l'image bruitée $\tilde{x}_t$, l'image de référence masquée y, le masque M et le niveau de bruit associé à l'étape t : σ_t^2.

Synthèse (inférence) Une fois le débruiteur f_θ entraîné, la synthèse consiste à tirer un échantillon aléatoire x_T et à le débruiter successivement de $t = T, \ldots, 0$. Pour plus de détails, nous renvoyons le lecteur vers les articles fondateurs [9, 2].

2.2 Un réseau léger pour l'inpainting mono-image

Pour l'architecture du réseau, nous avons choisi de simplifier au maximum l'architecture de Ho *et al.* [2]. Il s'agit donc d'un réseau de type UNet sans couche d'attention (Figure 1). Les entrées du réseau sont concaténées avant la première convolution. Dans le cas de l'inpainting mono-image, ou pour une modalité spécifique, il n'est pas nécessaire de multiplier les paramètres, nous limitons ainsi le nombre de canaux à 32. Finalement notre réseau a 160k paramètres contre 550M pour RePaint [3].

L'entraînement est fait en tirant aléatoirement des régions de taille 256x256 depuis une ou plusieurs images, qui sont ensuite bruitées et masquées. Le masque M est généré synthétiquement selon une formulation de Yu *et al.* [11].

Il est nécessaire d'entraîner le réseau à réaliser le débruitage pour tous les pas de temps, ils sont donc tirés uniformément dans $[0, T]$. Nous introduisons l'information temporelle au débruiteur sous la forme de la variance σ_t^2 associée à l'étape t.

3 Résultats

3.1 Expériences

Nous comparons notre réseau de diffusion frugal avec trois autres approches. La première est une approche classique par patchs [4], qui ne nécessite pas de phase d'apprentissage. La seconde est la formulation d'inpainting de Lugmayr *et al.* [3] qui représente l'état de l'art en inpainting. Leur réseau a plusieurs centaines de millions de paramètres et nécessite plusieurs dizaines de jours-GPU d'entraînement. Enfin, nous comparons notre inpainting par diffusion avec une approche standard par apprentissage sur un problème de **régression**.

Comparaison directe avec la régression à réseau fixe : Cette comparaison est effectuée *à architecture fixée* : nous isolons précisément l'apport de la modélisation par diffusion. Pour la régression, le réseau est entraîné en utilisant uniquement l'erreur L2 de reconstruction de l'image sur la zone masquée. Pour le réseau de diffusion, nous appliquons l'algorithme d'apprentissage classique [9, 2], en bruitant à différents pas de temps les images partiellement masquées. Les deux réseaux sont entraînés pendant le même temps (45 minutes).

Données : Nous nous plaçons dans le cas où nous avons accès à un jeu de données restreint, ou même à une seule image (cas "mono-image"). Nous considérons des images de textures et de petits jeux de données auto-similaires où peu d'images, de l'ordre de la dizaine ou de plusieurs dizaines, sont disponibles. Pour les images de textures une seule image est utilisée. Pour les dessins [8], plusieurs dizaines d'images sont disponibles pour l'entraînement. L'apprentissage est fait selon les données disponibles en préservant une région de test (Figure 2). La généralisation à de nouvelles textures ou modalités jamais vues ne fait pas partie de nos objectifs. Notons que notre méthode ne vise pas à traiter des grandes bases de données, contrairement aux réseaux massif de type RePaint.

Hyperparamètres : Nous entraînons notre réseau pour 15000 itérations avec des *batchs* de 16 images de taille 256x256. L'optimiseur est Adam dont le taux d'apprentissage initial est 1e-4. L'entraînement dure environ 40 minutes.

3.2 Résultats

Résultats qualitatifs La Figure 3 montre les résultats pour les différents modèles. On observe que la méthode par régression directe conduit à des résultats dans lesquels les variations stochastiques des images ne sont pas reproduits, contrairement aux méthodes par diffusion ou par patchs. Pour les images très régulières (avec des périodicités par exemple), la méthode par régression est capable de reconstruire partiellement la structure principale mais les résultats restent flou (Figure 3, ligne du haut). Ces pertes de détail et le flou des résultats sont le résultat

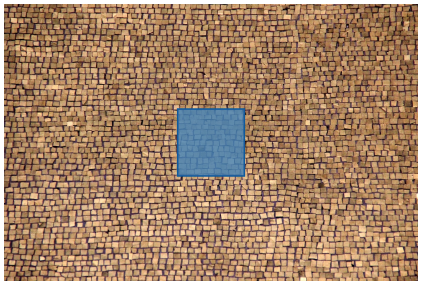

FIGURE 2 : Exemple de texture dans le cas mono-image. La zone de test (en bleu) est écartée pendant l'apprentissage.

TABLE 1 : Erreur de reconstruction sur 10 images de textures

Méthode	PSNR (dB) ↑	LPIPS ↓	SSIM ↑	#Param.
Patch	21.18	**0.072**	0.871	-
Régression	**24.30**	0.159	**0.902**	160k
Diffusion	21.32	**0.072**	0.881	160k
RePaint	21.28	0.078	0.879	553M

prévisible d'un effet de moyennage de la méthode par régression. Il est intéressant de constater que dans ce cas simple, et à architecture constante, l'apprentissage par diffusion seul permet d'obtenir des résultats nets et détaillés.

Observons également que, bien que non entraîné sur ces textures spécifiques, RePaint produit un inpainting satisfaisant. Cette capacité de généralisation peut être expliquée par la taille du réseau et la base de données conséquente d'entraînement.

Résultats quantitatifs Quantitativement, nous mesurons l'erreur de reconstruction entre la vérité terrain et une prédiction pour chaque méthode. Selon le tableau 1, le PSNR est meilleur pour la méthode par régression minimisant directement l'erreur quadratique comparée à une méthode par diffusion ou par patchs. Selon la mesure perceptuelle LPIPS[12], l'échantillon obtenu par diffusion est meilleur. Observons par ailleurs que ces résultats indiquent (confirment) que la mesure traditionnelle du PSNR pour l'évaluation en inpainting est insuffisante et ne reflète pas la qualité d'une méthode.

3.3 Frugalité

Les sections précédentes montrent que RePaint [3], produit des résultats de très bonne qualité et est capable de généraliser à des problèmes très spécifiques : textures particulières, nouvelle modalité (dessin). Ces capacités ont néanmoins été obtenues après une longue phase d'apprentissage et donc au prix d'un impact environnemental conséquent. Cet aspect est bien entendu important au vu des défis du changement climatique. Nous reportons dans le tableau 2 les consommations électriques estimées pour chaque méthode incluant le temps d'entraînement et le temps d'une inférence. L'équivalent CO2 est fourni à titre informatif pour une électricité française[1].

4 Conclusion

Les méthodes par diffusion se sont imposées comme l'état de l'art pour la génération d'images et la résolution de nombreux problèmes inverses. Cependant le coût d'entraînement de ces modèles est un obstacle majeur à leur utilisation pratique et pose des question sur leur caractère durable. Nous avons présenté un réseau reposant sur le principe de l'entraînement par diffusion mais ne gardant que l'essentiel pour son architecture, divisant le nombre de paramètres par plus que 1000 et le temps d'entraînement par 500 par rapport aux méthodes de l'état de l'art, sans compromettre la qualité des résultats dans les cas où un petit nombre d'image permet l'apprentissage.

[1] 56.9g éq. CO2/kWh en 2021. Source : Base Empreinte® (ADEME)

Références

[1] Ian GOODFELLOW, Jean POUGET-ABADIE, Mehdi MIRZA, Bing XU, David WARDE-FARLEY, Sherjil OZAIR, Aaron COURVILLE et Yoshua BENGIO : Generative Adversarial Nets. *In NIPS*. 2014.

[2] Jonathan HO, Ajay JAIN et Pieter ABBEEL : Denoising Diffusion Probabilistic Models. *In NIPS*, 2020.

[3] Andreas LUGMAYR, Martin DANELLJAN, Andres ROMERO, Fisher YU, Radu TIMOFTE et Luc VAN GOOL : RePaint : Inpainting using Denoising Diffusion Probabilistic Models. *In CVPR*, 2022.

[4] Alasdair NEWSON, Andrés ALMANSA, Yann GOUSSEAU et Patrick PÉREZ : Non-Local Patch-Based Image Inpainting. *IPOL*, 2017.

[5] Deepak PATHAK, Philipp KRAHENBUHL, Jeff DONAHUE, Trevor DARRELL et Alexei A. EFROS : Context Encoders : Feature Learning by Inpainting. *In CVPR*, 2016.

[6] Robin ROMBACH, Andreas BLATTMANN, Dominik LORENZ, Patrick ESSER et Björn OMMER : High-Resolution Image Synthesis With Latent Diffusion Models. *In CVPR*, 2022.

[7] Chitwan SAHARIA, William CHAN, Huiwen CHANG, Chris LEE, Jonathan HO, Tim SALIMANS, David FLEET et Mohammad NOROUZI : Palette : Image-to-Image Diffusion Models. *In SIGGRAPH*, 2022.

[8] Kazuma SASAKI, Satoshi IIZUKA, Edgar SIMO-SERRA et Hiroshi ISHIKAWA : Learning to restore deteriorated line drawing. *IJCG*, 2018.

[9] Jascha SOHL-DICKSTEIN, Eric WEISS, Niru MAHESWARANATHAN et Surya GANGULI : Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *In ICML*, 2015.

[10] Yonatan WEXLER, Eli SHECHTMAN et Michal IRANI : Space-time completion of video. *PAMI*, 2007.

[11] Jiahui YU, Zhe L. LIN, Jimei YANG, Xiaohui SHEN, Xin LU et Thomas S. HUANG : Generative Image Inpainting with Contextual Attention. *CVPR*, 2018.

[12] Richard ZHANG, Phillip ISOLA, Alexei A. EFROS, Eli SHECHTMAN et Oliver WANG : The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *In CVPR*, 2018.

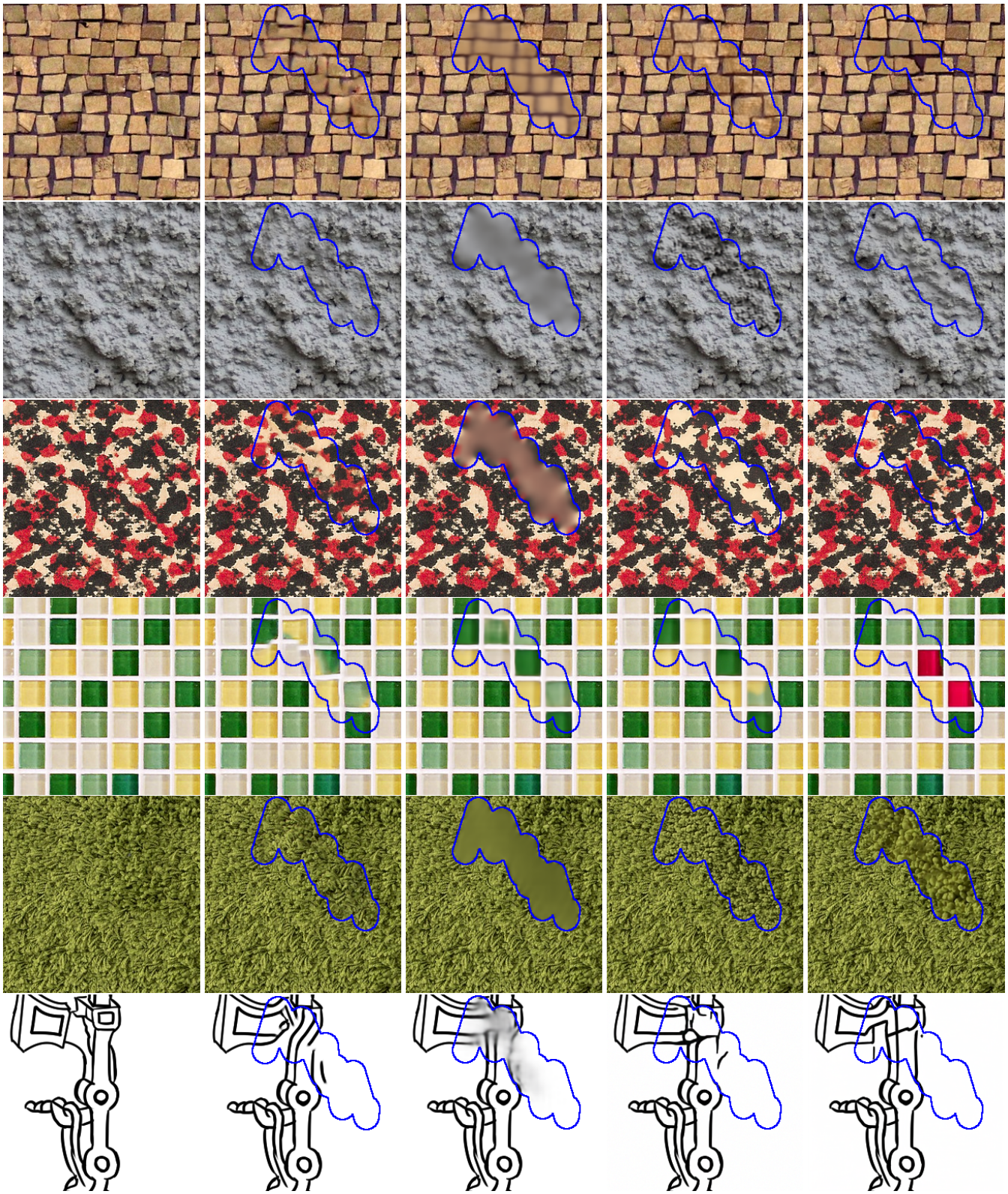

FIGURE 3 : Vérité-terrain et résultats par différentes méthodes (de gauche à droite) : par patchs, régression et diffusion mono-image, et par RePaint.

TABLE 2 : Impact environnemental de chaque méthode, principalement du temps d'entraînement et d'inférence

Méthode	Temps (h) Entraînement / Inférence	Électricité (kWh)	gCO2eq
Patch	- / 0.03	0.004	0.3
Diffusion	1 / 0.001	0.25	14
RePaint	576 / 0.17	172.8	9832