

Apprentissage de textures par réseau adversarial périodique (PSGAN)

Fatima-Zahrae EL-BOUKKOURI - Imane SI SALAH

November 2023

1 Introduction

Etant caractérisées par leur bruit aléatoire et leur périodicité, les textures jouent un rôle crucial dans la perception et les arts visuels. La synthèse de textures vise à apprendre les processus de génération à partir d'images exemples, les méthodes classiques s'appuyant sur des approches basées sur les instances ou paramétriques. L'apprentissage profond, illustré par Gatys et al. (2015a), a introduit une synthèse de texture paramétrique efficace et pilotée par les données, améliorée ultérieurement par Ulyanov et al. (2016) et Johnson et al. (2016) pour gagner en rapidité. Cependant, ces méthodes sont limitées par des descripteurs spécifiques, tels que la matrice de Gram, et sont confinées à la synthèse d'un seul exemple.

Les réseaux adversariels génératifs (GAN) présentent une approche alternative qui excelle dans la synthèse de texture guidée par les données, mais qui est contrainte par des tailles de sortie fixes. Li Wand (2016) ont proposé une architecture hybride combinant les GAN avec le descripteur de Gatys et al. (2015a) pour une synthèse de texture rapide. Les GAN spatiaux (SGAN) ont démontré une évolutivité en termes de vitesse et de mémoire, mais ont été limités dans le traitement de certaines classes de textures. La présente étude introduit le PSGAN, une nouvelle méthode de synthèse de texture neuronale non supervisée capable d'apprendre des textures périodiques et non périodiques à partir d'images uniques ou de grands ensembles de données. PSGAN permet un échantillonnage flexible dans l'espace de bruit, permettant la création de nouvelles textures avec une taille de sortie infinie et des transitions douces entre elles.

2 Méthodes de génération de textures

2.1 GANs

Les réseaux adversariaux génératifs (GAN) sont une approche de modélisation générative qui utilise des réseaux de neurones convolutifs profonds. Ils sont composés de deux sous-modèles : un modèle générateur G qui prend en entrée un vecteur de bruit et apprend à générer des images issues d'une distribution de données, et un discriminateur D qui tente de distinguer les images générées des vraies images. La fonction objectif du GAN est formulée comme un problème min-max comme suit :

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Ici x est une donnée réelle issue de la distribution p_{data} , et z est un vecteur de bruit tiré d'une distribution p_z . Le discriminateur D tente de maximiser sa capacité à distinguer les vraies images des images générées, tandis que le générateur G tente de minimiser la capacité de D à faire cette distinction, en d'autres termes, G cherche à produire des images qui sont indiscernables des vraies images pour D .

2.2 PSGAN

Dans le cadre de la génération de textures, les modèles GAN classiques peuvent rapidement être limités, car les images générées ont une taille fixe, déterminée au moment de l'entraînement, et la résolution des pixels est généralement basse.

Ainsi, le SGAN (Spatial GAN) a été introduit, remplaçant le vecteur de bruit en entrée du générateur des GANs par un tenseur z , intégrant les dimensions spatiales et permettant une meilleure scalabilité. De manière similaire, le discriminateur D est étendu mapper une image d'entrée X vers un champ bidimensionnel de taille spatiale $L \times M$. Chaque position du discriminateur résultant $D_{\lambda\mu}(X)$, pour $1 \leq \lambda \leq L$ et $1 \leq \mu \leq M$, répond uniquement à une partie locale X , que nous appelons le champ récepteur effectif de $D_{\lambda\mu}$. La réponse de $D_{\lambda\mu}(X)$ représente la probabilité estimée que la partie respective de X soit réelle plutôt que générée par G .

Cependant, si les composants de z sont échantillonnés indépendamment, cela rend le processus de génération ergodique et stationnaire, ce qui est problématique, en plus de ne pas pouvoir aligner les régions éloignées de l'image d'origine.

Le PSGAN, quant à lui, a été développé spécifiquement pour la synthèse de textures, en particulier celles présentant des caractéristiques périodiques. Son architecture se distingue par un tenseur d'entrée structuré pour intégrer des dimensions locales, globales et périodiques. Le vecteur d'entrée z du PSGAN est formulé comme $z = [z_{local}, z_{global}, z_{periodic}]$, où chaque composant joue un rôle distinct :

- z_{local} capture les détails et variations fines au niveau local de la texture.
- z_{global} influence les aspects globaux, affectant l'image dans son ensemble.
- $z_{periodic}$ introduit une périodicité contrôlée ce qui permet de générer des motifs répétitifs avec plus de cohérence.

Ces éléments interagissent au sein du réseau pour créer des textures riches et variées, illustrant la capacité du PSGAN à gérer différentes échelles de texture.

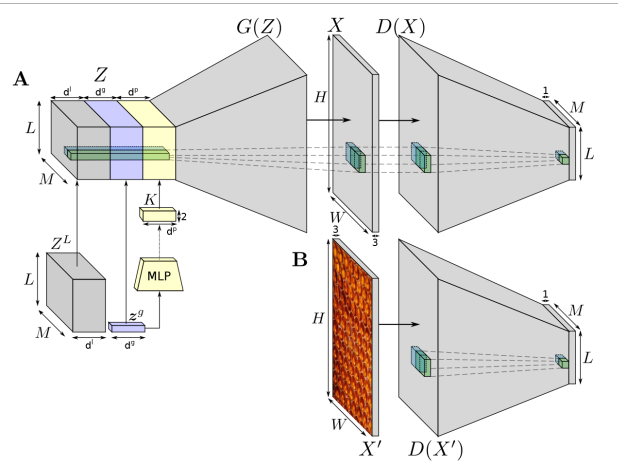


Figure 1: Architecture du PSGAN

Le PSGAN emploie une fonction de perte similaire à celle du GAN, seulement marginalisée sur les dimensions spatiales de la sortie du discriminateur. Cette fonction est minimisée sur G et maximisée sur D :

$$V(D, G) = \frac{1}{LM} \sum_{\lambda=1}^L \sum_{\mu=1}^M \mathbb{E}_{Z \sim p_z(Z)} [\log(1 - D_{\lambda\mu}(G(Z)))] + \frac{1}{LM} \sum_{\lambda=1}^L \sum_{\mu=1}^M \mathbb{E}_{X \sim p_{data}(X)} [\log(D_{\lambda\mu}(X'))]$$

Comme le modèle est conçu pour pouvoir apprendre sur une seule image, des patches X' de l'image sont sélectionnés à des positions aléatoires durant l'entraînement.

2.3 Méthode de la littérature: Synthèse par Copier-Coller de Patches

Cette méthode de synthèse de texture repose sur l'assemblage de patches extraits d'une image exemple pour créer une nouvelle image de texture.

1. **Extraction de Patches** : On considère l'ensemble des patches d'une image exemple A . Ces patches sont échantillonnés pour synthétiser une nouvelle image de texture. Pour un pixel p de A , soit $N_A(p) = \{A(p + a_1), \dots, A(p + a_K)\}$, le patch placé en p où $F = \{a_1, \dots, a_K\}$ constitue un patch
2. **Distance entre Patches** : Pour synthétiser la valeur d'un pixel p' en B , on évalue la distance entre le patch localisé à p' en B ($N_B(p')$) avec tous les patches de A .

$$d_v(N_A(p), N_B(p')) = \frac{1}{K} \sum_{i=1}^K (A(p + a_i) - B(p' + a_i))^2$$

3. **Sélection de Pixels Candidats** : Pour synthétiser la valeur en un pixel p' de B , on considère un ensemble de pixels candidats dans A , noté $O_\varepsilon(p') = \{p \mid d_v(N_A(p), N_B(p')) < \varepsilon\}$ pour un $\varepsilon > 0$. On tire alors un pixel p de O_ε et on affecte la valeur $B(p') = A(p)$.
4. **Synthèse Gloutonne** : La synthèse s'effectue de manière gloutonne (greedy), en choisissant pour chaque pixel de la zone à synthétiser le patch dont le voisinage est le plus proche de l'exemple source.

La méthode de synthèse par copier-coller de patches a été choisie pour plusieurs raisons. Premièrement, sa simplicité et son implémentation facile. Deuxièmement, elle illustre les principes de base de la synthèse de texture, offrant un point de départ pour comprendre des méthodes plus complexes. Enfin, elle permet l'exploration directe de l'impact de la réutilisation de motifs existants sur la création de nouvelles textures, un aspect central dans l'étude des textures.

2.4 Comparaison entre les deux méthodes

La comparaison entre le PSGAN et la méthode de synthèse par copier-coller de patches révèle des plusieurs différences.

Avantages du PSGAN : Le PSGAN, grâce à son architecture basée sur les réseaux de neurones, est capable de générer des textures plus complexes et variées. Il excelle dans la création de textures qui ne peuvent pas être facilement répliquées par des méthodes classiques. De plus, le PSGAN peut apprendre de grandes variétés de textures et produire des résultats qui sont souvent plus naturels et moins répétitifs.

Inconvénients du PSGAN : La complexité du PSGAN est nettement plus élevée que celle de la méthode de synthèse par patch.

Avantages de la Synthèse par Patch : La synthèse par patch est simple à comprendre et à implémenter, la rendant idéale pour les applications où la rapidité et la simplicité sont prioritaires.

Inconvénients de la Synthèse par Patch : Cette méthode peut conduire à des textures répétitives et manque souvent de la complexité et de la variété offertes par des approches comme le PSGAN. Elle est moins efficace pour créer des textures qui nécessitent une grande diversité ou des motifs non répétitifs.

En résumé, tandis que le PSGAN offre une solution avancée et flexible pour la génération de textures variées et complexes, la méthode de synthèse par patch reste une option viable pour des applications nécessitant une approche plus directe et moins computationnellement intensive.

3 Expérimentation

3.1 Données d'entraînement

Le PSGAN a été entraîné sur des jeux de données contenant un mélange de motifs périodiques et non périodiques, tels que l'Oxford Describable Textures Dataset.

3.2 Protocole expérimental

3.2.1 Architecture du PSGAN

Le PSGAN implémenté utilise une architecture de réseau de neurones profonds avec les caractéristiques suivantes :

- **Générateur** : Le générateur du PSGAN est composé de plusieurs couches convolutionnelles. Chaque couche est suivie d'une fonction d'activation non linéaire, comme ReLU, et est normalisée par batch.
- **Discriminateur** : Le discriminateur suit une structure similaire, avec des couches convolutionnelles entrecoupées de fonctions d'activation et de normalisation par batch.
- **Dimension des Tenseurs d'Entrée** : Les dimensions du tenseur d'entrée z sont définies pour capturer les aspects locaux, globaux et périodiques des textures.

3.2.2 Hyperparamètres

Les hyperparamètres utilisés pour l'entraînement du PSGAN sont les suivants :

- **Taux d'Apprentissage** : Un taux d'apprentissage initial de 0.0002 a été utilisé, avec une décroissance adaptative au fil des époques.
- **Optimiseur** : L'optimiseur Adam a été utilisé pour son efficacité dans la formation des réseaux de neurones profonds.
- **Fonction de Perte** : La fonction de perte combinée du PSGAN, intégrant à la fois la perte adversative et la perte de contenu, est utilisée pour entraîner le réseau.

3.3 Résultats

3.3.1 Résultats du PSGAN

Les textures générées par le PSGAN démontrent une variété et une complexité remarquables. Les images suivantes illustrent des exemples de textures synthétisées à l'aide du PSGAN. On observe que les textures générées sont riches en détails et présentent une variabilité naturelle, sans répétitions visibles excessives. Pour illustrer la capacité du PSGAN à traiter des scénarios variés, nous examinons deux cas distincts. Le premier exemple met en avant des textures complexes, offrant une vue aérienne de la ville de Milan. Les résultats obtenus révèlent la capacité du PSGAN à capturer les détails complexes et la diversité des structures urbaines, allant des bâtiments emblématiques aux zones verdoyantes.



Figure 2: Comparaison des textures originales et celles générées par le PSGAN avec $n_{epoch} = 100$, $depth_{UnetGenerator} = 5$, $depth_{DiscrimblenMoinator} = 5$, $batchsize = 64$, $channels_{generator} = 120$, $channels_{discriminator} = 120$

Dans le deuxième exemple, nous nous concentrons sur des textures simples, en l'occurrence un motif de ruche. Ce cas met en lumière la capacité du PSGAN à générer des textures avec des motifs réguliers tout en préservant la clarté et la netteté des détails. Ces résultats démontrent la polyvalence du PSGAN dans la génération de textures, qu'elles soient complexes ou simples, tout en maintenant une qualité visuelle élevée.

3.3.2 Résultats de la Synthèse par Copier-Coller de Patches

La méthode de synthèse par copier-coller de patches produit des textures qui, bien que plus simples et parfois répétitives, sont visuellement cohérentes. Les images ci-dessous montrent des textures obtenues par cette méthode. Ces résultats illustrent la capacité de la méthode à créer des textures homogènes à partir d'un ensemble limité de patches.

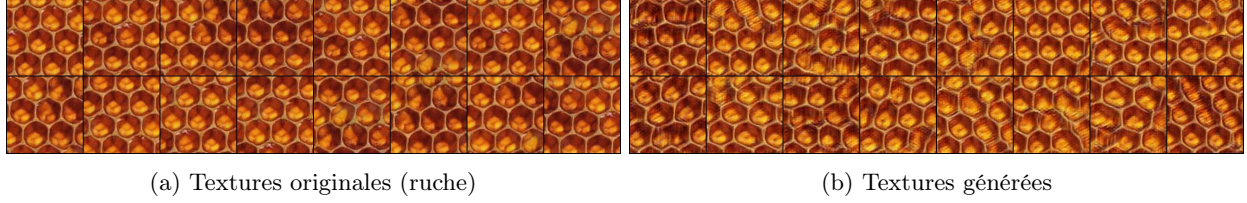


Figure 3: Comparaison des textures originales et celles générées par le PSGAN avec $n_{epoch} = 25$, $depth_{U_{net}Generator} = 5$, $depth_{DiscrimbleMoinator} = 5$, $batchsize = 64$, $channels_{generator} = 120$, $channels_{discriminator} = 120$

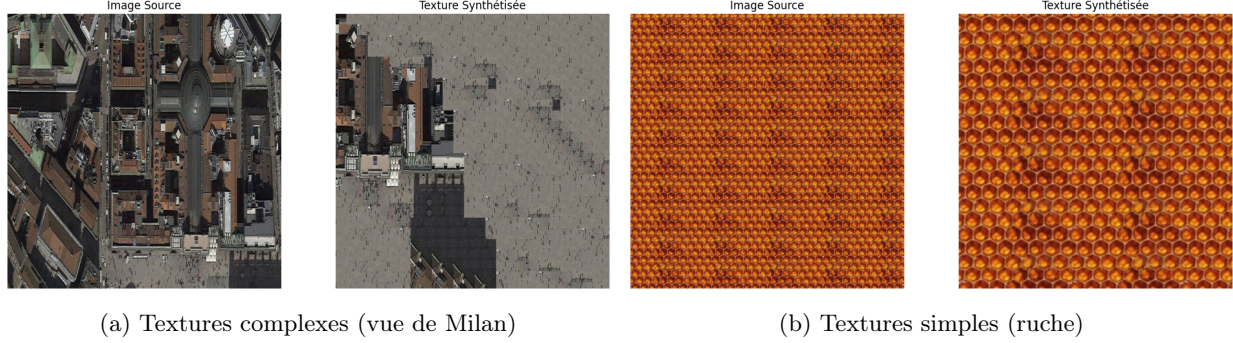


Figure 4: Textures synthétisées par la méthode de copier-coller de patches

4 Discussion

Les expériences menées ont révélé des différences notables dans la capacité des deux méthodes à gérer des textures de complexité variable.

4.0.1 Textures Complexes - Image de la Ville de Milan

Lors de l'application des méthodes sur une image complexe, comme une vue aérienne de la ville de Milan, le PSGAN a produit des résultats impressionnants, démontrant sa capacité à gérer des détails complexes et à créer une texture réaliste et variée. En revanche, la méthode de synthèse par copier-coller de patches a montré ses limites : le rendu était médiocre, avec des répétitions visibles et une incapacité à reproduire fidèlement la complexité de la texture originale. Pour évaluer la stabilité du modèle PSGAN au fil de l'entraînement, nous avons tracé les valeurs de $D(x)$ (discriminateur pour les images réelles) et $D(G(z))$ (discriminateur pour les images générées) en fonction des epochs. La figure 5 illustre l'évolution de ces valeurs au cours de l'apprentissage.

4.0.2 Textures Moins Complexes - Structure en ruche

En contrast, lors de l'utilisation d'une texture moins complexe, telle qu'une structure en ruche, les deux méthodes ont donné de bons résultats. Cela suggère que, bien que la méthode de copier-coller de patches soit moins capable de gérer des textures complexes, elle peut être tout à fait adéquate pour des textures plus simples et régulières.

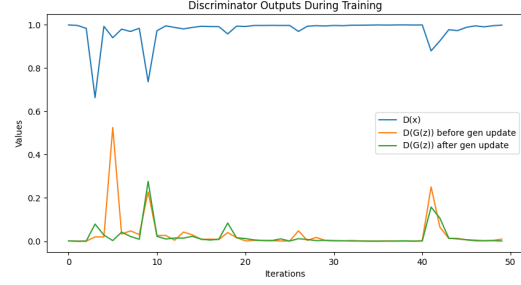
L'évaluation de la stabilité du modèle PSGAN dans le cas de textures moins complexes a révélé une convergence rapide de $D(x)$ vers 1 et de $D(G(z))$ vers 0. Ces comportements indiquent une stabilité accrue dans l'apprentissage, où le discriminateur parvient rapidement à distinguer entre les images réelles et générées.

4.1 Implications et Applications Pratiques

Ces observations mettent en lumière les domaines d'application spécifiques pour lesquels chaque méthode est la plus adaptée. Le PSGAN, avec sa capacité à gérer des textures complexes, est idéal pour des applications



(a) Cas des textures complexes (vue de Milan)



(b) Cas des textures simples (ruche)

Figure 5: Tracé de $D(x)$ et de $D(G(z))$ lors de l'entraînement su PSGAN

nécessitant un haut degré de réalisme et de variété, comme dans la modélisation urbaine ou la création de contenus pour des jeux vidéo.

D'autre part, la méthode de copier-coller de patches, en raison de sa simplicité et de sa rapidité, peut être préférée pour des applications où la texture est moins complexe ou lorsque la rapidité de génération est un facteur critique, comme dans des applications de design rapide ou dans des contextes où les ressources de calcul sont limitées.

En conclusion, bien que le PSGAN offre une flexibilité et une performance supérieures dans la création de textures complexes, la méthode de copier-coller de patches demeure une option viable pour des textures plus simples et des applications moins exigeantes en termes de variété et de réalisme.