HAL open science

Apprentissage d'une fonction de régularisation locale pour la restauration d'images

Jean Prost, Antoine Houdard, Andrés Almansa, Nicolas Papadakis

To cite this version:

Jean Prost, Antoine Houdard, Andrés Almansa, Nicolas Papadakis. Apprentissage d'une fonction de régularisation locale pour la restauration d'images. Congrès des jeunes chercheurs en vision par ordinateur (ORASIS'21), Centre National de la Recherche Scientifique [CNRS], Sep 2021, Saint Ferréol, France. hal-03339625

HAL Id: hal-03339625

https://hal.science/hal-03339625v1

Submitted on 9 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Apprentissage d'une fonction de régularisation locale pour la restauration d'images
Learning a local regularization function for image restoration

Jean Prost[1] Antoine Houdard [1] Andrés Almansa[2] Nicolas Papadakis[1]

[1] Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251,F-33400 Talence, France
[2] Université de Paris, MAP5, CNRS, F-75006 Paris, France

jean.prost@math.u-bordeaux.fr

Résumé

Dans ce travail, nous élaborons une stratégie afin d'apprendre une fonction de régularisation pour résoudre des problèmes de restauration d'images. Une fonction de régularisation locale, paramétrée par un réseau de neurones convolutif, est entraînée telle une "critique" entre deux distributions non appariées de patches propres et dégradés, en utilisant un critère d'apprentissage inspiré par les Wasserstein-GAN. La fonction de régularisation globale est ensuite définie comme la valeur moyenne de la fonction locale sur l'ensemble des patches de l'image. L'efficacité de cette méthode est démontrée sur des problèmes de débruitage et de défloutage.

Mots Clef

Restauration d'image, problèmes inverses, régularisation, réseaux de neurones convolutifs, patches

Abstract

In this work, we propose a framework to learn a local regularization model for solving general image restoration problems. This regularizer is defined with a fully convolutional neural network that sees the image through a receptive field corresponding to small image patches. The regularizer is then learned as a critic between unpaired distributions of clean and degraded patches using a Wasserstein generative adversarial network based energy. This yields a regularization function that can be incorporated in any image restoration problem. The efficiency of the framework is finally shown on denoising and deblurring applications.

Keywords

Image restoration, Inverse problem, regularization, CNN, patches

1 Introduction

1.1 Définition du problème

L'objectif de ce papier est de proposer une méthode de restauration d'images pour résoudre des problèmes inverses. Ces problèmes s'expriment de la manière suivante : à partir d'une observation corrompue y, retrouver l'image originale et propre $x^\star$. On considère que les images $x^\star$ et y sont reliées par le modèle de dégradation suivant:

$$y = Ax^\star + \epsilon, \tag{1}$$

où $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ est un bruit blanc gaussien, et A est l'opérateur de dégradation, qui peut être par exemple l'opérateur identité (débruitage), un masque (inpainting), ou un opérateur de floutage (défloutage). Les problèmes de ce genre étant généralement mal-posés, il est nécessaire d'incorporer une forme de savoir *a-priori* sur la solution attendue pour pouvoir les résoudre. Cela peut se faire en introduisant une formulation variationnelle, incorporant un terme de régularisation $R(x)$. Une estimation $\hat{x}$ de la solution du problème $x^\star$ est alors définie comme :

$$\hat{x} \in \arg\min_x \frac{1}{2\sigma^2}\|Ax - y\|^2 + \lambda R(x), \tag{2}$$

où $\frac{1}{2\sigma^2}\|Ax-y\|^2$ est le terme d'attache aux données qui assure que la solution soit consistante avec l'observation dégradées y, $R(x)$ est la fonction de régularisation, et $\lambda \geq 0$ contrôle l'importance relative des deux termes. On peut alors noter que, dans le cas où $R(x) = -\log(P_X(x)) + C$, la solution $\hat{x}$ de (2) correspond à l'estimateur du maximum a posteriori. Un choix judicieux serait donc de définir $R(x) = -\log(P_X(x))$, mais, étant donné la grande dimension et la haute complexité des images naturelles, modéliser directement la distribution *a-priori* $P_X(x)$ est trop difficile. Il est alors nécessaire de définir une fonction de régularisation $R(x)$ dont le comportement se rapproche le plus possible de celui de $-\log(P_X(x))$, en pénalisant les "mauvaises" solutions tout en favorisant les "bonnes".

Ainsi, le choix de la fonction de régularisation a un impact important sur les propriétés de la solution du problème variationnel (2). De nombreux termes de régularisation ont été définis pour résoudre les problèmes inverses. Nous revenons sur ces différents termes ci-dessous.

1.2 Les fonctions de régularisation

Régularisation convexe Parmi les termes de régularisation les plus populaires pour la restauration d'images, on retrouve des termes convexes non-lisses, tels que la variation totale [19], la norme ℓ^1 des coefficients d'une décomposition en ondelettes [6, 4], d'une transformée de Fourier ou d'une DCT locale [24]. Ce type de régularisation ne correspond néanmoins qu'à une approximation grossière des statistiques et de la géométrie des images naturelles et a tendance à produire des résultats trop lisses, pouvant contenir un effet marche d'escalier ou d'autres type d'artefacts.

Régularisation non convexe Plus récemment, des a priori plus adaptés aux images naturelles sont apparus, sous la forme de termes de régularisation non convexe et paramétriques, généralement déterminés en utilisant des méthodes statistiques. Ces termes prennent par exemple la forme de mélanges de gaussiennes sur les patches, ou de réseaux de neurones convolutifs (CNN). Néanmoins, les réseaux convolutifs employés à des fins de régularisation sont généralement entraînés de telle sorte que la fonction de régularisation ne soit connue que de manière implicite, par son gradient [2, 18, 17], ou par son opérateur proximal [22, 14, 26, 10, 20]. Ainsi, la convergence des algorithmes d'optimisation mettant en jeu une régularisation par réseaux convolutifs n'est pas garantie, ou alors seulement sous des conditions restrictives [17, 20]. Pour surpasser ces limitations, tout en exploitant la puissance des réseaux convolutifs sur les problèmes d'images, des méthodes basées sur une régularisation explicite par réseaux convolutifs ont été développées, notamment en définissant la fonction de régularisation comme la mesure "push-forward" d'un réseau génératif [3] ou d'un autoencodeur variationnel [7], ou en définissant la fonction de régularisation plus directement comme un réseau [13]. Toutes ces approches sont toutefois limitées à des classes particulières d'images, et ne se généralisent pas à des images de taille arbitraire.

Régularisation non convexe sur les patches Une autre manière d'encoder dans un terme de régularisation de l'information a priori sur les images naturelles est de travailler à l'échelle des patches. Les patches d'une image sont des sous-images de dimension réduite. Apprendre un modèle de régularisation à l'échelle des patches permet de réduire la complexité du problème liée à la grande dimensionnalité des images. Les modèles de régularisation sur les patches dépendent de modèles paramétriques comme les mélanges de gaussiennes [28, 9, 21], mais ces modèles sont trop simples pour représenter de manière précise la complexité de l'espace des patches.

1.3 Contributions et aperçu

Dans ce travail, précédemment présenté dans [16], nous proposons de combiner les avantages liés aux méthodes de régularisation par patches avec les atouts des méthodes de régularisation basées sur des réseaux convolutifs.
Ainsi, nous introduisons une fonction de régularisation explicite, non-convexe, et paramétrée par un réseau convolutif, qui agit de manière locale. Cette fonction peut ainsi être appliquée sur des images de n'importe quelle taille. Dans un premier temps, la fonction de régularisation est entraînée comme une critique entre deux distributions de patches propres et dégradés en utilisant un critère d'apprentissage se basant sur la théorie des Wasserstein-GAN [1], comme proposé dans [13]. La fonction de régularisation locale est dans un second temps intégrée dans un modèle de régularisation par patches pour réaliser la restauration, dans une formulation inspirée de [28]. Dans la section 2 nous décrivons la stratégie pour apprendre une fonction de régularisation locale, puis nous détaillons dans la section 3 comment est résolu le problème variationnel. Dans la section 4 nous donnons les détails d'implémentation et d'entraînement de notre méthode, puis, dans la section 5 nous présentons des exemples d'applications de notre méthode sur des problèmes de débruitage et de défloutage.

2 Apprentissage de la fonction de régularisation

2.1 Régularisation locale

Pour réduire la difficulté d'apprendre une fonction de régularisation pour des images de grandes dimensions, une stratégie efficace est de réduire la complexité du problème en apprenant une fonction de régularisation locale r sur des patches de dimensions réduites [28]. Une fois que l'on dispose d'une fonction de régularisation locale r, on peut alors simplement définir une fonction de régularisation globale $R(x)$ comme la valeur moyenne de la fonction locale sur l'ensemble des patches de l'image x :

$$R(x) = \frac{1}{K} \sum_{u_k \in \Omega_x} r(u_k), \tag{3}$$

où $\Omega_x = \{u_1, ..., u_K\}$ correspond à l'ensemble des K patches de dimension spatiale $p \times p$ contenus dans l'image x. Ici nous travaillons avec tous les patches de x, de telle sorte que les patches de Ω_x peuvent se superposer. On a donc en pratique un nombre de patches égal à $K = (H-p)(W-p)$ pour une image x de dimension $H \times W$.

2.2 Régularisation adversarielle

Dans cette partie nous détaillons comment est définie et entraînée la fonction de régularisation locale. La fonction de régularisation locale r_θ est conçue comme un réseau convolutif paramétré par ses poids θ. La taille du champ réceptif du réseau correspond à la taille de patch considérée pour notre modèle.

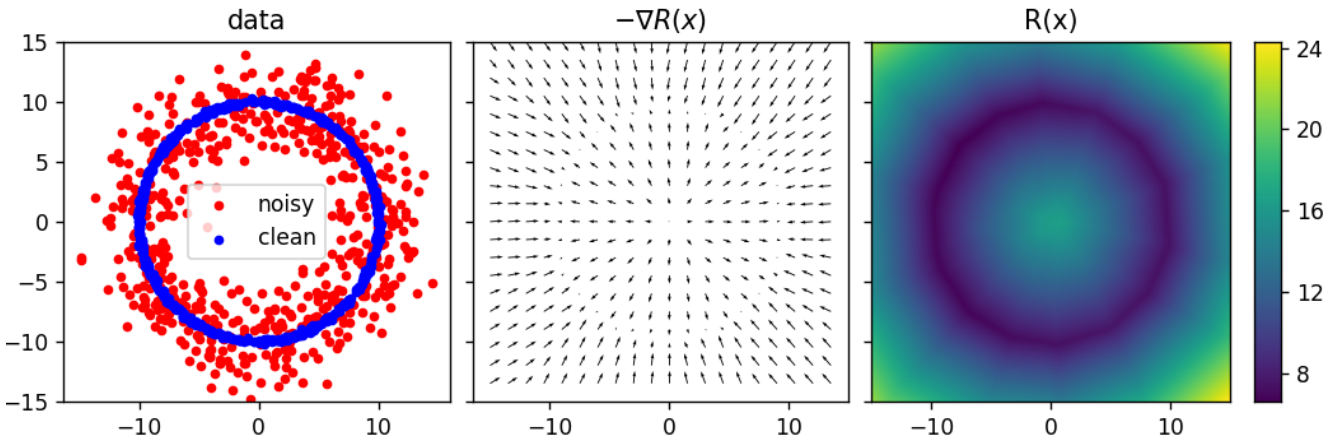


Figure 1: Exemple de l'apprentissage d'une fonction de régularisation sur des données synthétiques en 2d. A gauche les données propres (en bleue) réparties sur une variété en forme de cercle, et bruités (en rouge). Au centre, le gradient de la fonction de régularisation apprise, est de norme 1 en dehors de la variété correspondante aux données (le cercle). A droite, la fonction de régularisation apprise, est proportionelle à la distance à la variété.

Le critère d'entraînement utilisé est une adaptation de la méthode de [13] pour des patches d'images naturelles. L'idée est d'entraîner la fonction de régularisation r_θ comme la critique d'un Wasserstein-GAN [8] entre une distribution de patches propres $\mathbb{P}_u$ et une distribution de patches dégradés $\mathbb{P}_v$. Ainsi nous cherchons les paramètres θ du réseau qui maximisent le critère :

$$\begin{aligned} D(\theta) =& \mathbb{E}_{z\sim\mathbb{P}_v}\left[r_\theta(z)\right] - \mathbb{E}_{z\sim\mathbb{P}_u}\left[r_\theta(z)\right] \\ &- \mu\mathbb{E}_{z\sim\mathbb{P}_i}[(||\nabla_z r_\theta(z)||_2 - 1)^2], \end{aligned} \tag{4}$$

où $\mathbb{P}_i$ est la distribution de toutes les lignes connectant des points échantillonnés dans $\mathbb{P}_u$ et $\mathbb{P}_v$. Autrement dit, le terme du bas de (4) est un terme de pénalité sur la norme du gradient, encourageant la norme du gradient à être égale à 1 dans l'enveloppe convexe de l'union des supports de $\mathbb{P}_u$ and $\mathbb{P}_v$. D'un point de vue pratique le critère d'entraînement (4) pousse la fonction de régularisation à prendre une valeur élevée sur les patches dégradés et une valeur faible sur les patches propres. De plus la pénalisation de la norme du gradient permet d'empêcher les problèmes d'explosion du gradient et de gradient nul qui pourrait entraver la résolution du problème variationnel (2) avec une méthode du type descente du gradient. Une illustration de l'effet de la pénalisation de la norme du gradient sur des données synthétiques est donnée dans la figure 1.
On peut par ailleurs établir un lien entre le critère d'entraînement (4) et la théorie du transport optimal. En effet, un réseau entraîné avec un tel critère est une approximation du potentiel de Kantorovich φ dans le cadre de la formulation duale d'un problème de transport optimal entre les distributions $\mathbb{P}_u$ et $\mathbb{P}_v$ [1]:

$$r_\theta^\star \in \arg\max_{\varphi\in\mathrm{Lip}_1} \mathbb{E}_{z\sim\mathbb{P}_v}\left[\varphi(z)\right] - \mathbb{E}_{z\sim\mathbb{P}_u}\left[\varphi(z)\right]. \tag{5}$$

Il est alors possible de montrer que, sous certaines conditions, la solution de l'équation (5) correspond à la distance à la variété des patches propres [13]. Ainsi, on peut interpréter chaque itération d'une descente du gradient pour résoudre (2) comme un rapprochement des patches de l'image à restaurer x vers la variété des patches propres.

3 Résolution du problème variationnel

Dans cette partie, nous détaillons comment résoudre le problème variationnel (2) une fois la fonction de régularisation locale r_θ entraînée.
Pour trouver un minimum local du problème variationnel, nous utilisons une descente de gradient explicite. Ainsi, en notant x^ℓ l'image à l'itération l, une itération de la descente est donnée par :

$$x^{\ell+1} = x^\ell - \eta\left(\frac{1}{\sigma^2}A^*(Ax^\ell - y) + \lambda\nabla R(x)\right), \tag{6}$$

où A^* est l'opérateur adjoint de l'opérateur de dégradation linéaire A, et $\eta > 0$ est un pas de temps.
D'un point de vue pratique, cette méthode demande de calculer à chaque itération la gradient de la fonction de régularisation globale $\nabla R(x)$. Cette opération peut être réalisé simplement grâce à l'architecture convolutive de la fonction de régularisation locale. En effet, comme illustré sur la figure 2, en appliquant la fonction de régularisation locale sur l'image entière x à la manière d'un réseau "fully-convolutionnal" [12], on obtient en sortie une carte en 2 dimesions dont la valeur de chaque pixel correspond à la valeur de la fonction de régularisation locale sur le patch correspondant de l'image d'entrée. Ainsi, il suffit de prendre la valeur moyenne de l'image en sortie du réseau pour obtenir la valeur de la fonction de régularisation globale. De même, on peut calculer efficacement la valeur du gradient de $R(x)$ en utilisant la différentiation automatique. Cette approche est considérablement plus efficace que si

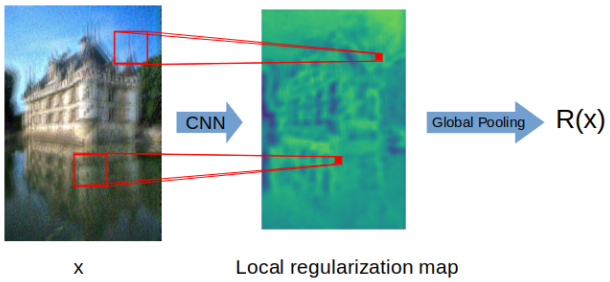


Figure 2: Calcul de la valeur de la fonction de régularisation globale. En appliquant le réseau local sur l'image entière on obtient la valeur de la fonction de régularisation locale sur chaque patches. La moyenne de ces valeurs donne la fonction de régularisation globale $R(x)$.

l'on devait traiter chaque patch de manière individuelle.

4 Considérations pratiques

Dans cette section, nous donnons les détails pratiques sur l'architecture et la procédure d'entraînement de la fonction de régularisation, ainsi que sur la résolution du problème variationnel.

4.1 Architecture de la fonction de régularisation

La fonction de régularisation est conçue comme un réseau de convolutions de 7 couches, chaque couche étant composée de convolution 3×3, suivie d'une activation non-linéaire ReLU [15]. La fonction de régularisation a ainsi un champ réceptif de taille 15×15, de telle sorte que quand le réseau prend un patch de taille 15×15 en entrée, il retourne un scalaire en sortie. Nous avons aussi entraîné des réseaux de 3 couches, correspondant à un champ réceptif de taille 7×7, mais nous avons observé de meilleurs résultats pour le réseau de taille 15×15. Par la suite, les résultats présentés correspondront aux résultats obtenus avec le réseau de taille 15×15.

4.2 Entraînement de la fonction de régularisation

Pour entraîner la fonction de régularisation, nous construisons une base de données de patches propres $\mathcal{D}_c$, contenant des patches de dimension 15×15 ayant été extraits à partir de 30000 images différentes. De même, nous construisons un dataset de patches bruités $\mathcal{D}_n$ avec des patches de la même taille et extraits à partir de 30 000 autres images. Les patches de $\mathcal{D}_n$ sont bruités avec un bruit blanc gaussien $\epsilon \sim \mathcal{N}(0, \sigma^2_{train} I)$. Les images proviennent du google landmarks dataset [23]. Le réseau est entraîné pour maximiser le critère (4) en utilisant l'algorithme d'optimisation Adam [11], avec les hyper-paramètres par défaut $\beta_1 = 0.9$ et $\beta_2 = 0.999$, et un pas d'apprentissage décroissant, égal à 10^{-3} à la première itération, et 10^{-4} à la dernière itération. La taille des batches est égale à 32 et nous réalisons 10^5 itérations. Enfin, le paramètre d'importance de la pénalité du gradient μ est égal à 5. L'algorithme d'entraînement est détaillé dans l'algorithme 1.

Algorithm 1 Apprentissage de la fonction de régularisation locale r_θ

Entrés: Bases de données de patches propres $\mathcal{D}_c$ et bruités $\mathcal{D}_n$; paramètre de pénalisation du gradient μ; taille de batche m, nombre d'itérations K
Sortie: fonction de régularisation locale r_θ
pour $k = 1$ **à** K **faire**
  Tirer m patches propres $\{z^c_j\}_{j=1}^m$ dans $\mathcal{D}_c$
  Tirer m patches bruités $\{z^n_j\}_{j=1}^m$ dans $\mathcal{D}_n$
  Tirer m nombres aléatoires $\{\alpha_j\}_{j=1}^m \in [0;1]^m$
  Définir les patches interpolés : $z^i_j = \alpha_j z^c_j + (1-\alpha_j) z^n_j$
  pour $j = 1$ **à** m **faire**
    $D_j(\theta) = r_\theta(z^n_j) - r_\theta(z^c_j) - \mu(||\nabla_z r_\theta(z^i_j)||_2 - 1)^2$
  fin pour
  $\theta \leftarrow Adam(\nabla_\theta \sum_{j=1}^m D_j(\theta))$
fin pour

En pratique, nous avons observé que prendre un niveau de bruit variable $\sigma \in [0.05; 0.3]$ dans la base de données des patches bruités $\mathcal{D}_n$ permet d'obtenir les meilleurs résultats pour la restauration d'image.La figure 3 illustre la capacité de la fonction de régularisation à distinguer les patches propres des patches bruités.

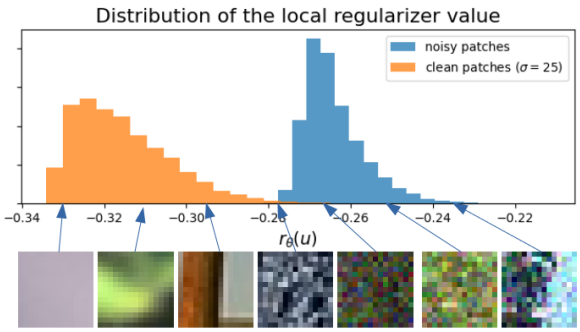


Figure 3: Distribution de la valeur de la fonction de régularisation locale sur des patches propres (en bleue) et bruités (orange). Des exemples de patches correspondants à chaque valeur sont donnés en dessous.

4.3 Résolution du problème variationnel

La restauration d'image est réalisée en résolvant (2). En pratique nous avons observé que l'utilisation d'un algorithme d'optimisation avec une méthode de momentum et un pas de descente adaptatif comme Adam permet d'obtenir de meilleurs résultats, tout en convergant plus rapidement qu'une simple descente de gradient. Ainsi, nous utilisons 50 itération d'Adam, avec les hyper-paramètres réglés à leurs valeurs par défaut ($\beta_1 = 0.9$ et $\beta_2 = 0.999$) et une décroissance exponentielle du pas de la descente, qui passe de 0.01 à 0.001 entre la première et

(a) noisy (20.35dB) (b) BM3D (25.48dB) (c) EPLL (25.48 dB) (d) ALR (26.96 dB) (e) clean

Figure 4: Comparaison visuelle du débruitage pour $\sigma = 0.1$.

la dernière itération. De plus le paramètre de régularisation λ est choisis empiriquement de manière à maximiser la qualité de la restauration pour chaque niveau de bruit.

5 Application

5.1 Débruitage

Dans cette partie nous évaluons l'efficacité de notre méthode dans le cas du débruitage, avec un bruit blanc gaussien $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Cela revient à résoudre le problème (2) avec comme opérateur de dégradation $A = I$.

Nous testons notre méthode sur les 68 images du dataset BSD68, nous évaluons le *peak-to-signal-ratio* (PSNR) ainsi que la similarité perceptuelle (LPIPS) [27] avec la vraie solution. Nous comparons notre méthode avec 2 méthodes populaires et similaires de débruitage : EPLL [28], qui définit aussi une fonction de régularisation sur les patches, mais avec une mélange de gaussiennes, et BM3D [5], une méthode exploitant la redondance des patches à l'intérieur d'une image. Une comparaison en terme de métriques est donnée dans la table 1, ainsi qu'une comparaison visuelle dans la figure 4. Nous nommons notre méthode ALR, pour *Adversarial Local Regularization*. La régularisation adversaire locale que nous proposons donne des résultats de qualité similaire à ces méthodes de débruitage, à la fois en terme de métriques et de qualité visuelle.

Table 1: Comparaison des performances de débruitage en termes de PSNR (gauche) et similarité perceptuelle (droite) pour 3 algorithmes de débruitage. Les résultats correspondent à une moyenne sur les 68 images de la base BSD68.

	PSNR			LPIPS		
σ	ALR	EPLL	BM3D	ALR	EPLL	BM3D
0.1	**28.85**	28.77	28.26	0.29	**0.28**	0.30
0.2	24.88	**24.92**	24.69	0.44	**0.42**	0.43
0.4	**21.58**	19.75	20.25	**0.57**	0.61	0.58

5.2 Défloutage

Notre fonction de régularisation locale peut aussi être incorporée dans des problèmes de restauration plus complexe, tel que le défloutage. Le problème de défloutage correspond à résoudre le problème inverse (1) dans le cas ou l'opérateur linéaire de dégradation A correspond à une convolution avec un noyau de floutage k :

$$y = k * x + \epsilon. \tag{7}$$

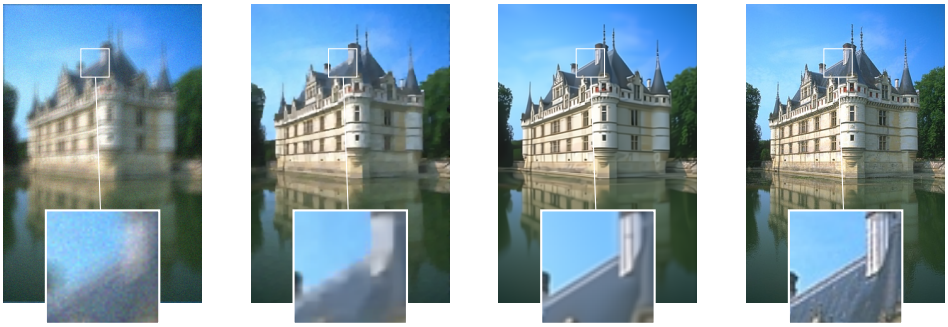
Dans la figure 5, nous illustrons la capacité de notre fonction de régularisation apprise comme une critique entre des patches propres et bruités à régulariser des problèmes de défloutage. Nous comparons notre méthode avec une méthode de type plug-and-play [25].

6 Conclusion

Dans ce travail, nous proposons une méthode pour résoudre des problèmes inverses en utilisant un réseau convolutif comme une fonction de régularisation locale. La fonction de régularisation est entraînée comme la critique d'un Wasserstein-GAN entre une distribution de patches bruités et une distribution de patches propres. Cette approche offre plusieurs avantages. Elle peut être adaptée à des images de n'importe quelle taille, l'entraînement de la fonction de régularisation ne nécessitant pas de données appariées. Elle peut également être utilisée pour des problèmes de restauration non-supervisés, et l'aspect local de la fonction de régularisation permet de réduire le besoin en données d'apprentissage. De plus, la fonction de régularisation apprise peut être utilisée pour résoudre différents types de problèmes inverses (débruitage, défloutage...).

Les résultats expérimentaux sur le débruitage montrent que, notre méthode peut rivaliser avec des algorithmes classiques de débruitage, illustrant le potentiel des réseaux convolutifs à être employés comme des fonctions de régularisation pour des problèmes inverses.

Pour de futurs travaux, nous pensons qu'améliorer le critère d'entraînement de la fonction de régularisation pourrait permettre d'améliorer les performances de la



(a) blurry (b) ALR (25.67 dB) (c) DPIR (28.28dB) (d) clean

Figure 5: Comparaison visuelle des méthodes de défloutage, dans le cadre d'un floutage lié au mouvement (motion blur).

restauration. En effet, le critère d'entraînement est associé à la distance 1-Wasserstein. Cela implique que, bien entraînée, la fonction de régularisation est une fonction linéaire de la distance à la variété des patches naturels. Le terme d'attache aux données évolue lui quadratiquement en fonction de la distance à la variété. Ainsi, définir un terme de régularisation qui croît lui aussi de manière quadratique en fonction de la distance à la variété pourrait permettre d'avoir un meilleur équilibre entre l'attache aux données et la régularisation, et donc permettre d'obtenir une meilleur restauration.

Remerciements Ce travail a été réalisé avec le support financier de l'Agence Nationale de la Recherche dans le cadre du projet PostProdLEAP (ANR-19-CE23-0027-01).

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[2] S. A. Bigdeli, M. Jin, P. Favaro, and M. Zwicker. Deep Mean-Shift Priors for Image Restoration. In *Adv. in Neural Information Proces. Systems 30*, pages 763–772, 2017.

[3] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. Compressed sensing using generative models. In *Int. Conference on Machine Learning*, volume 2, pages 537–546, 2017.

[4] R. R. Coifman and D. L. Donoho. Translation-Invariant De-Noising. In *Wavelets and Statistics (Lect. Notes in Statistics, vol 103)*, chapter 5, pages 125–150. Springer, 1995.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. on Image Processing*, 16:2080–95, 2007.

[6] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[7] M. González, A. Almansa, M. Delbracio, P. Musé, and P. Tan. Solving Inverse Problems by Joint Posterior Maximization with a VAE Prior. 2019.

[8] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.

[9] A. Houdard, C. Bouveyron, and J. Delon. High-dimensional mixture models for unsupervised image denoising (HDMI). *SIAM J. Imag. Sc.*, 11(4):2815–2846, 2018.

[10] U. S. Kamilov, H. Mansour, and B. Wohlberg. A plug-and-play priors approach for solving nonlinear imaging inverse problems. *IEEE Signal Processing Letters*, 24(12):1872–1876, 2017.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[13] S. Lunz, O. Öktem, and C.-B. Schönlieb. Adversarial regularizers in inverse problems. In *Adv. in Neural Information Processing Systems*, volume 31, pages 8507–8516, 2018.

[14] T. Meinhardt, M. Moller, C. Hazirbas, and D. Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *International Conference on Computer Vision*, pages 1781–1790, 2017.

[15] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 2010.

[16] J. Prost, A. Houdard, A. Almansa, and N. Papadakis. Learning local regularization for variational image restoration. In A. Elmoataz, J. Fadili, Y. Quéau, J. Rabin, and L. Simon, editors, *Scale Space and Variational Methods in Computer Vision*, pages 358–370, Cham, 2021. Springer International Publishing.

[17] E. T. Reehorst and P. Schniter. Regularization by Denoising: Clarifications and New Interpretations. *IEEE Trans. on Computational Imaging*, 5(1):52–67, mar 2019.

[18] Y. Romano, M. Elad, and P. Milanfar. The Little Engine That Could: Regularization by Denoising (RED). *SIAM J. on Imaging Sciences*, 10(4):1804–1844, 2017.

[19] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[20] E. K. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin. Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning*, pages 5546–5557, 2019.

[21] A. M. Teodoro, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Scene-Adapted Plug-and-Play Algorithm with Guaranteed Convergence: Applications to Data Fusion in Imaging, 2018.

[22] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *IEEE Global Conference on Signal and Information Processing*, pages 945–948, 2013.

[23] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval, 2020.

[24] G. Yu and G. Sapiro. DCT image denoising: a simple and effective image denoising algorithm. *Image Processing On Line*, 1, 2011.

[25] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte. Plug-and-play image restoration with deep denoiser prior. *arXiv preprint arXiv:2008.13751*, 2020.

[26] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning Deep CNN Denoiser Prior for Image Restoration. In *IEEE Conf. Comput. Vis. Pat. Recog.*, pages 2808–2817, 2017.

[27] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.

[28] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Int. Conference on Computer Vision*, pages 479–486, 2011.