

# Sélection d'hyperparamètres non supervisée par différentiation automatique, application à la segmentation de textures

Leo DAVY Nelly PUSTELNIK Patrice ABRY

ENSL, CNRS, Laboratoire de physique, F-69342 Lyon, France

Travail soutenu par l'ANR-19-CE48-0009 Multisc'In, la DGA/AID (bourse de thèse n° 01D22020572) et le Projet IADoc@UdL (Université de Lyon) « contrats doctoraux en intelligence artificielle 2020 » (ANR-20-THIA-0007-01).

**Résumé** – La segmentation de textures par approche variationnelle est confrontée à la difficulté récurrente de la sélection de l'hyperparamètre réglant l'importance relative entre le terme d'attache des données et celui de pénalisation. Dans cette contribution, nous nous concentrons sur une approche variationnelle permettant de combiner estimation de descripteurs locaux des propriétés statistiques (ici, invariance d'échelle et anisotropie) et segmentation, via une pénalisation favorisant les solutions constantes par morceaux. La sélection automatique, pilotée par les données, mais non supervisée, est ici abordée via le formalisme de SURE combiné au potentiel du déroulement d'un algorithme itératif d'optimisation et bénéficiant des procédures de différentiation automatique sous-jacentes.

**Abstract** – Texture segmentation by variational approach raises the long-lasting issue of the selection of the hyperparameter that governs the balance between data fidelity and regularizing constraints. In this contribution, we focus on a variational approach allowing to combine estimation of local descriptors of statistical properties (here, scale invariance and anisotropy) and segmentation, considering a penalization favoring piecewise constant solutions. The challenge of devising automated, data-driven, yet unsupervised hyperparameter selection strategies is here addressed by exploring SURE formalism and the benefits of unrolling optimization iterative algorithms combined with underlying automatic differentiation procedures.

## 1 Introduction

La notion de texture dans les images est fondamentale pour leur caractérisation et leur analyse. Dans ce travail, nous nous intéressons à la tâche spécifique de segmentation de textures et à l'estimation des hyperparamètres sous-jacents dont le choix est crucial pour obtenir une segmentation de qualité.

La tâche de segmentation de texture repose soit sur une formulation en deux étapes (extraction de descripteurs puis segmentation) [10] ou en une étape visant à combiner les deux précédentes [4, 11]. C'est dans cette deuxième classe de méthode que notre travail s'inscrit en poursuivant l'effort mené dans [12] où nous avons proposé d'estimer des descripteurs de régularité (et variance) locale isotropes avec à priori de constance par morceaux, tout en conservant une formulation variationnelle convexe, bien que non-lisse.

Dans [4], nous avons proposé une approche variationnelle permettant d'introduire des descripteurs anisotropes et nous nous concentrons ici sur la tâche complémentaire d'estimation de l'hyperparamètre pondérant attache au modèle de régression linéaire et constance par morceaux.

Dans les problèmes de segmentation de textures, mais également de façon plus générale dans les problèmes variationnels, ces hyperparamètres vont généralement quantifier des compromis type biais-variance et leur choix dépend fortement du contexte d'observation des données (par ex. niveau de bruit dans l'observation; difficulté du problème; sensibilité souhaitée dans la distinction des textures;...).

Il est donc nécessaire en pratique d'avoir un moyen d'estimer ces hyperparamètres. La méthode théoriquement idéale consiste à choisir les hyperparamètres qui donnent la solution la plus proche de la vraie solution, i.e. minimisant le (vrai) risque. La texture 'vraie' étant inaccessible, il est commun de

faire appel à l'estimateur non-biaisé du risque introduit par Stein [17] (SURE) qui a déjà été appliqué dans de nombreuses situations telles que la sélection de modèle [18], le débruitage d'images [6, 9] ou bien la segmentation de textures isotropes [12]. Cet estimateur est particulièrement adapté à la grande dimensionalité des images qui permet par concentration de la mesure d'avoir que la valeur empirique de SURE est très proche de sa valeur moyenne, le vrai risque.

**Contributions** – On présente ici une méthodologie simple à implémenter et généralisable à d'autres problèmes que la segmentation de texture qui permet la sélection rapide hyperparamètres par différentiation automatique, notamment dans un contexte non-supervisé en choisissant une fonction de coût appropriée.

**Plan** – Nous introduisons en section 2 la formulation variationnelle présentée dans [4] qui permet de réaliser de la segmentation de textures à l'aide d'un formalisme multi-échelle anisotrope. Nous décrivons en section 3 comment calculer SURE dans la situation considérée afin d'estimer le vrai risque. Suite à cela, nous présentons dans la section 4 comment minimiser le vrai risque ainsi que le risque estimé par SURE dans le contexte de la Différentiation Automatique (DA). Nous présentons en section 5 des résultats expérimentaux pour montrer l'efficacité de la méthode qui parvient en quelques dizaines d'itérations à obtenir un hyperparamètre optimal, sans nécessiter la connaissance a priori de la solution.

## 2 Segmentation de texture anisotrope par approche variationnelle

**Segmentation de texture** – On considère ici le problème de segmentation de textures basé sur des descripteurs de variance et de régularité locale notés  $(v, h)$  (cf. [11]). Plus pré-

En particulier, on suppose qu'il existe une partition de l'image  $\Omega = \bigcup_{m \in M} \Omega_m$  telle que sur chacune de ces parties ces descripteurs sont constants. Formellement on suppose

$$(\beta_k \geq \Omega_m) \quad z_k = (v_m, h_m). \quad (1)$$

et on s'intéresse au problème inverse de retrouver la partition  $\Omega = \bigcup_{m \in M} \Omega_m$  à partir de l'observation de l'image  $X$ .

**Modèle et analyse de textures homogènes** – On s'intéresse ici à des textures appelées Champs Browniens Fractionnaires Anisotropes (AFBF)  $X^f$ , qui ont été étudiés dans [2, 15]. Ces textures présentent l'intérêt d'être adaptées à la modélisation d'un grand nombre de champs étant caractérisés par leur régularité sans avoir à supposer d'isotropie [13].

Les (log-modules des) coefficients en ondelettes à une position  $k \geq \Omega$ , une échelle  $j \geq f_1, \dots, J_g$  et une bande  $b \geq f_1, \dots, B_g$  engendrés par un tel champ vérifient

$$l_{j,k}^{(b)} = \log |j h X^f|, \psi_{j,k}^{(b)} |j|^2 \quad (2)$$

$$v_k^{(b)} + j h_k^{(b)} + \zeta_{j,k}^{(b)}. \quad (3)$$

où  $\psi_{j,k}^{(b)}(x) := 2^{j-2} \psi(2^{-j}(x - k))$ . Ce modèle, en plus d'être simple est interprétable dans le sens où les  $h^{(b)}$  (resp.  $v^{(b)}$ ) peuvent être mis en correspondance avec une régularité (resp. variance) directionnelle locale.

La transformée en ondelettes choisie dans notre étude est la Dual-Tree Complex Wavelet Transform (DTCWT - [16]) non-décimée qui fournit une famille de  $B = 6$  ondelettes à support compact, dont chacune des orientations  $b$  est facilement interprétable et dont les modules des coefficients sont stables par translation.

Dans ce travail, nous proposons d'approcher  $\zeta = (\zeta_j)_j$  par un vecteur gaussien centré ayant pour covariance  $S$ , c-à-d  $\zeta \sim N(0, S)$ . Le contenu de  $S$  dépend à la fois du champ considéré  $X^f$  et des ondelettes. Comme décrit dans [12], il est nécessaire d'estimer  $S$  avec suffisamment de précision. Dans notre cadre d'étude, la covariance est à estimer localement à la fois entre les positions, les échelles et les directions. L'estimation de covariance est en général une tâche difficile pour laquelle de nombreuses stratégies peuvent être envisagées [14].

**Formulation variationnelle** – Nous proposons de formuler le problème de segmentation sous la forme d'un problème variationnel [4], qui, de façon conjointe, réalise un compromis entre respect du modèle (2) (via  $F = \sum_k \frac{1}{2} j l_k - \Phi \sum_j j^2$  où

$$\Phi = \begin{pmatrix} 1 & 0 & \dots & 0 & J & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & J & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & 0 & \dots & \dots & 0 & J \end{pmatrix} \quad (4)$$

est une matrice de taille  $B \times J = 2B$  - avec  $\mathbf{0} = (0, \dots, 0)^T \in \mathbb{R}^J$ ,  $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^J$  et  $\mathbf{J} = (1, \dots, J)^T \in \mathbb{R}^J$  - qui agit sur chaque position  $k$ ) et le fait qu'une segmentation est une description constante par morceaux (via la variation totale  $k D(v^>, h^>) > k_{2,1} = \sum_k \frac{1}{2} (j l_k - \Phi (v_k^>, h_k^>)^2 + j^2 + \lambda k D(v^>, h^>) > k_{2,1})$ ). On est donc amené à résoudre le problème variationnel

$$z = (l) = \operatorname{argmin}_{(v,h)} \frac{1}{2} \sum_k j l_k - \Phi (v_k^>, h_k^>)^2 + j^2 + \lambda k D(v^>, h^>) > k_{2,1} \quad (5)$$

dans lequel l'hyperparamètre  $\lambda$  permet de pondérer l'attachement aux données et la contrainte de constance par morceaux.

En particulier, dans le cas  $\lambda = 0$  aucune régularisation n'est appliquée, c'est à dire que l'on résout ponctuellement (2) indépendamment des points voisins, autrement dit on calcule la régression linéaire ordinaire qui explicitement se calcule comme  $z = \Phi^{-1} \Phi^T l$ . Tandis que pour  $\lambda \neq 0$ , n'importe quelle variation amènerait une pénalisation arbitrairement grande dans (5) et donc les solutions obtenues tendent à être constantes. Un cas particulier (isotrope) de (5) est présenté dans [12] où les  $(v, h)$  sont identiques à travers les directions ( $B = 1$ ).

**Résolution du problème variationnel** – Finalement, comme décrit dans [4, 11] on peut utiliser l'algorithme [3]

$$\begin{cases} z^{(t+1)} = \operatorname{prox}_{(v,h)}(z^{(t)} + \sigma^{(t)} \lambda D \tilde{u}^{(t)}) \\ u^{(t+1)} = \operatorname{prox}_{(v,h)}(u^{(t)} - \tau^{(t)} \lambda D z^{(t+1)}) \\ \theta^{(t)} = \frac{1}{1 + 2 \tau^{(t)}}, \\ \tau^{(t+1)} = \tau^{(t)} \theta^{(t)}, \quad \sigma^{(t+1)} = \sigma^{(t)} / \theta^{(t)} \\ \tilde{u}^{(t+1)} = u^{(t+1)} + \theta(u^{(t+1)} - u^{(t)}) \end{cases} \quad (6)$$

qui converge vers une solution de (5). L'un des intérêts de cet algorithme qui est développé ici est qu'à l'exception de la mise à jour de  $z_t$  qui est faite par une fonction non-linéaire qui agit pixel par pixel (pouvant être interprétée comme la fonction d'activation HardTanh [8]), toutes les autres opérations sont implémentables à l'aide d'applications linéaires. Cet algorithme est donc facile à implémenter sous la forme d'un réseau de neurones (c-à-d comme combinaison d'opérateurs linéaires et d'applications non-linéaires ponctuelles).

### 3 Fonctions de coût

**Introduction** – On dispose ainsi avec (6) d'un moyen itératif de résoudre le problème (5) pour un  $\lambda$  fixé. Cependant, cela ne suffit pas pour obtenir une solution pertinente du problème de segmentation, il faut en effet avoir un moyen de choisir un  $\lambda$  adapté à l'observation. Il faut donc introduire une fonction, appelée fonction de coût, qui permet de mesurer cette pertinence. On décrira donc dans un premier temps une fonction de coût si on connaît *a priori* la solution et dans un second temps on présentera une fonction de coût qui ne requiert pas la connaissance de la solution optimale (1)

Les deux méthodes ont les bonnes propriétés d'être relativement simples à implémenter et d'atteindre leurs minimas pour des valeurs de  $\lambda$  très proches.

**Estimation supervisée** – Dans un premier temps, on considère le problème d'estimation de  $\lambda$  supervisé, c'est à dire en considérant que l'on connaît à la fois le masque utilisé et les valeurs prises en moyenne par  $(v_k, h_k)$  en chaque point, valeurs qui ne dépendent que de la texture  $m$  à laquelle appartient  $k$ , l'ensemble de ces points étant noté  $\Omega_m$  (cf. (1)). On peut alors définir la fonction d'erreur qui nous servira de référence

$$\text{SSE}(\lambda) := \sum_j j z_j - \sum_j j^2. \quad (7)$$

**Estimation non-supervisée via SURE** – Dans des contextes analogues [9, 12] il a été observé que même si la vérité terrain  $z$  n'est pas connue, il est en fait possible à partir de l'estimateur non biaisé du risque de Stein [17] (SURE) d'obtenir de très bonnes approximations de (7). La fonction de coût n'est alors plus la SSE mais le critère de SURE dont l'expression est dans notre cadre d'étude [12] :

$$\text{SURE}(z) = \|A(z - \hat{z})\|_2^2 + \text{Tr}(SA @_j z) - \text{Tr}(ASA) \quad (8)$$

où  $A = (\Phi \Phi)^{-1} \Phi$ . L'estimateur étant non biaisé on a  $E \text{SURE}(\lambda)(\zeta) = \text{SSE}(\lambda)$ .

**SURE en pratique** – On utilise pour le calcul de  $\text{SURE}(\lambda)$  l'approximation<sup>1</sup> suivante pour  $\nu \neq 0$  et  $\epsilon \sim N(0, I)$  un vecteur Gaussien de même dimension que  $l$  :

$$jjz = z =_0jj^2 + \frac{2}{\nu}hz (1 + \nu\epsilon) = z (1), (\Phi \Phi)^{-1} \Phi S\epsilon i. \quad (9)$$

Cette approximation est non-biaisée dans le sens suivant :

$$\lim_{\nu \rightarrow 0} E ; (9) \quad \text{Tr}(ASA) = \text{SSE}(\lambda). \quad (10)$$

Afin de calculer numériquement l'expression (9), il est nécessaire de connaître la matrice de covariance  $S$ , ou au moins une bonne approximation de celle-ci. On utilise ainsi un calcul de la matrice de covariance empirique

$$\mathfrak{S}_{j_1, k_1; b_1}^{j_2, k_2; b_2} = \frac{1}{jKj} \times \begin{matrix} l_{j_1, k_1}^{(b_1)} l_{j_2, k_2}^{(b_2)} \\ h_{j_1}^{(b_1)} i \quad h_{j_2}^{(b_2)} i \end{matrix}. \quad (11)$$

Contrairement à [12] où une seule orientation est présente, ici il est important de prendre en compte les interactions entre orientations  $b$  ainsi qu'entre échelles  $j$  et différents incréments  $\delta = k_2 - k_1$ . L'un des intérêts de la DTCWT est de réduire les interactions entre échelles et orientations, cependant il reste nécessaire d'estimer  $\mathfrak{S}$  pour des incréments calculés sur des pas du même ordre de grandeur que l'ondelette analysante (c-à-d  $2^j$ ).

## 4 Estimation des hyperparamètres

**Principe de la minimisation par différentiation automatique** – Traditionnellement, minimiser (7) ou (9) se fait soit par une recherche coûteuse sur une grille, soit en calculant explicitement (à la main) des gradients puis en implémentant des schémas d'optimisation afin d'obtenir un hyperparamètre adapté au problème considéré [5].

Une méthode alternative qui est aujourd'hui particulièrement répandue suite au développement des réseaux de neurones et permise par l'augmentation des capacité de calcul des ordinateurs est la Différentiation Automatique (DA) [1, 7].

---

### Algorithme 1 : Optimisation sous PyTorch

---

```

1  $c :=$  fonction de coût
2  $\lambda := \text{torch.nn.Parameter}(\quad)$ 
3 pour  $i = 1, \dots, N_{\text{epochs}}$  faire
4   Calculer  $z$ 
5   Calculer  $c(z)$ 
6    $\lambda = \text{optim}(\text{Backprop}(c(z)))$ 
7 fin
Sorties :  $\lambda = \lambda_c^{\text{optim}}(N_{\text{epochs}})$ 

```

---

**Implémentation** – Dans ce contexte, on implémente directement l'algorithme proximal (6) dans un langage adapté, dans notre cadre d'étude PyTorch, capable de calculer automatiquement les gradients et de les rétropropager. Ce calcul de gradients est alors réalisé par l'algorithme de DA en utilisant le fait que les opérations réalisées dans un ordinateur se réduisent en des opérations élémentaires (type  $+$ ,  $-$ ,  $\dots$ ) pour

<sup>1</sup>Afin d'obtenir cette approximation on utilise pour obtenir le premier terme que  $z =_0(l) = (\Phi \Phi)^{-1} \Phi l$  est la solution ordinaire ( $=_0$ ) du problème de régression linéaire minimisant l'erreur quadratique. Pour le second terme on utilise d'une part  $\text{Tr}(M) = E \langle M ; \cdot \rangle$  et d'autre part une approximation par différence finie de pas  $h$  (cf. [12]). Cette approximation est notée  $\tilde{d}f(\cdot)$ .

lesquelles on connaît des expressions explicites des gradients correspondant. La DA calcule ainsi la *valeur* de ces gradients *élémentaires* et les conserve au travers de toutes les opérations réalisées, qui grâce à la "chain-rule" permet de calculer le gradient de  $c$  en l'itération courante. On peut alors chercher à minimiser  $c(z)$  à l'aide d'un algorithme d'optimisation opérant sur les gradients, les plus répandus étant (S)GD et ADAM. Cette méthode est résumée dans l'algorithme 1.

On peut alors appliquer l'algorithme 1 avec comme fonction de coût  $c$  la fonction définie en (7) ou bien celle définie (10) selon que l'apprentissage soit supervisé ou non-supervisé, afin de trouver une valeur adaptée de  $\lambda$ .

L'un des intérêts principaux de cette méthode est sa flexibilité car une fois la méthode mise en place, il n'y a que très peu de difficulté à changer  $\lambda \forall z$  ou bien la fonction de coût  $z \forall c(z)$  ou même la méthode d'optimisation de  $\lambda$ , ce qui n'est pas le cas pour des méthodes plus traditionnelles qui requièrent de nouveaux calculs et implémentations pour n'importe laquelle de ces modifications.

## 5 Résultats expérimentaux

On applique la méthodologie décrite ci-dessus sur des images de taille  $j\Omega_j = 256 \times 256$ , désignées par Texture A et Texture B (cf. figures 1a-2a), chacune combinant deux textures homogènes et dont les transitions sont gérées par combinaison convexe de deux AFBFs (cf. figures 1b-2b).

La Texture A (resp. B) est caractérisée par  $v_1 = v_2$  (resp.  $v_1 = v_2 + \delta$  avec  $jj\delta jj \ll 1$ ) et  $h_1 = h_2 + 0.3 = 0.7$  (resp.  $h_1 = h_2 + 0.1 = 0.7$ ) avec les notations de (1).

**Comparaison SSE-SURE** – Dans les figures 1e (resp. 2e) nous affichons l'évolution de la SSE et de SURE en fonction de  $\lambda$  pour la Texture A (resp. B). Bien que les courbes SSE et SURE soient distinctes d'un terme additif (d'après (10)), mais aussi par l'approximation (11), leurs minimas coïncident, c-à-d  $\lambda_{\text{SSE}} = \lambda_{\text{SURE}}$ .

**Résultats de l'estimation du minimum de SSE ou SURE par grid-search et ADAM/SGD** – Afin de comparer les résultats entre méthodes supervisée et non-supervisée, on peut définir la méthode d'optimisation `optim` parmi `GS` (grid-search<sup>2</sup>); `ADAM` et (S)GD. Chacune de ces méthodes pouvant être utilisée pour minimiser la fonction de coût pouvant prendre la forme de la SSE dans le cas supervisé ou bien de SURE dans le cas non-supervisé.

On peut alors définir l'hyperparamètre  $\lambda_c^{\text{optim}}(N)$  obtenu après  $N$  itérations de l'algorithme (1). Les images associées sont présentées Figure 1 (resp. 2) pour la Texture A (resp. B).

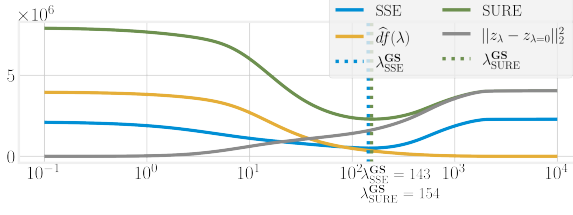
Les figures 1f et 2f permettent d'illustrer que les 3 approches (GS, (S)GD et ADAM) convergent vers des valeurs de critères proches à la fois pour SSE et pour SURE. Nous observons également que la stratégie de recherche de  $\lambda$  basée sur (S)GD permet dans les deux configurations d'atteindre une valeur de SSE et SURE plus faible que par grid-search, fournissant ainsi une valeur de  $\lambda$  plus proche de la valeur optimale.

Les figures 1g et 2g montrent que, quelle que soit la fonction de coût ou bien la méthode d'optimisation, les trois méthodes convergent vers des valeurs de lambda proches que ce soit en terme de SSE ou de SURE.

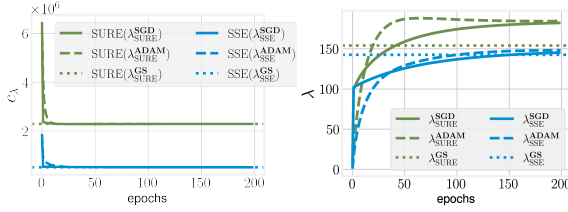
<sup>2</sup>Dans le cas de l'optimisation par GS le nombre d'itérations correspond à  $N := |\Lambda|$ , dans le cas où le pas de la grille est "on a  $|\Lambda| \sim \epsilon^{-P}$  où  $P$  est le nombre d'hyperparamètres à sélectionner.



(a) Texture A (b) Masque (c)  $Z_{\text{SGD}}^{\text{SSE}}$  (d)  $Z_{\text{SGD}}^{\text{SURE}}$



(e)  $\text{SSE}(\cdot)$  et  $\text{SURE}(\cdot)$



(f)  $c(Z(N_{\text{epochs}}))$

(g)  $(N_{\text{epochs}})$

FIGURE 1 : (1ère ligne) Texture A, Masque et résultats de segmentation avec  $\lambda$  trouvé par SGD en minimisant SSE ou SURE. (2ème ligne) Evolution des critères SSE et SURE en fonction de  $\lambda$ . (3ème ligne, gauche (resp. droite)) Evolution de la fonction de coût (resp.  $\|z_\lambda - z_{\lambda=0}\|_2^2$  estimé par minimisation de) SSE ou SURE avec différentes stratégies de recherche de paramètres optimaux.

## 6 Conclusion et Perspectives

Nous avons ici présenté des résultats montrant l'efficacité de la différentiation automatique pour la sélection automatique d'hyperparamètres dans le contexte de la segmentation de textures anisotropes. Bien que la méthode de résolution, et le problème posé, soient non lisses la méthodologie présentée converge rapidement vers des valeurs minimales des critères considérés. On a également montré l'efficacité de la méthode pour un critère tel que SURE qui permet ainsi de faire de l'estimation non supervisée.

L'un des intérêts de cette méthode reposant sur sa flexibilité, nous pourrions étudier par la suite le comportement de la fonctionnelle (5) pour un plus grand nombre d'hyperparamètres. Par exemple, nous pourrions considérer un hyperparamètre par direction ( $\lambda \geq \mathbb{R}^B$ ) ou bien par coefficient de régression linéaire ( $\lambda \geq \mathbb{R}^{2B}$ ). En effet, dans ce type de situation implémenter des méthodes type grid-search semble intractable, le nombre d'éléments de la grille étant exponentiel en la dimension de l'espace des hyperparamètres.

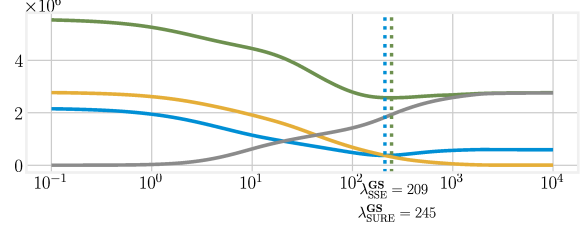
De plus, d'autres questions qui n'ont pas été traitées ici seront étudiées par la suite telles que l'estimation de  $S$ , la dépendance par rapport au nombre d'itérations de (6) ou bien l'utilisation de métriques plus adaptées à la segmentation que l'erreur quadratique.

## Références

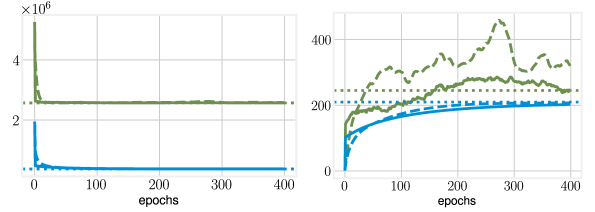
- [1] Q. BERTRAND et AL. : Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *J. Mach. Learn.*, 23(149):1–43, avril 2022.
- [2] A. BONAMI et A. ESTRADE : Anisotropic analysis of some gaussian models. *J. Fourier Anal. Appl.*, 9(3):215–236, 2003.
- [3] A. CHAMBOLLE et T. POCK : A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.*, 40(1):120–145, 2011.



(a) Texture B (b) Masque (c)  $Z_{\text{ADAM}}^{\text{SSE}}$  (d)  $Z_{\text{ADAM}}^{\text{SURE}}$



(e)  $\text{SSE}(\cdot)$  et  $\text{SURE}(\cdot)$



(f)  $c(Z(N_{\text{epochs}}))$

(g)  $(N_{\text{epochs}})$

FIGURE 2 : Résultats sur Texture B

- [4] L. DAVY, N. PUSTELNIK et P. ABRYS : Combining dual-tree wavelet analysis and proximal optimization for anisotropic scale-free texture segmentation. *In IEEE ICASSP 2023*, Rhodes Island, Greece, 2023.
- [5] C-A. DELEDALLE, S. VAITER, J. FADILI et G. PEYRÉ : Stein unbiased gradient estimator of the risk (sugar) for multiple parameter selection. *SIAM J. Imaging Sci.*, 7(4):2448–2487, 2014.
- [6] D. L. DONOHO et I. M. JOHNSTONE : Minimax estimation via wavelet shrinkage. *Ann. Stat.*, 26(3):879 – 921, 1998.
- [7] A. GRIEWANK : On automatic differentiation. *Mathematical Programming : recent developments and applications*, 6(6):83–107, 1989.
- [8] H.T.V. LE, N. PUSTELNIK et M. FOARE : The faster proximal algorithm, the better unfolded deep learning architecture ? the study case of image denoising. *In Proc. EUSIPCO*, pages 947–951, 2022.
- [9] C. LOUCHET et L. MOISAN : Posterior expectation of the total variation model : properties and experiments. *SIAM J. Imaging Sci.*, 6(4):2640–2684, 2013.
- [10] J. D. B. NELSON et N. C. KINGSBURY : Dual-Tree wavelets for estimation of locally varying and anisotropic fractal dimension. *In IEEE ICIP*, Hong Kong, Sept. 26-29, 2010.
- [11] B. PASCAL, N. PUSTELNIK et P. ABRYS : Strongly convex optimization for joint fractal feature estimation and texture segmentation. *Appl. Comp. Harm. Analysis*, 54:303–322, 2021.
- [12] B. PASCAL, S. VAITER, N. PUSTELNIK et P. ABRYS : Automated data-driven selection of the hyperparameters for total-variation-based texture segmentation. *J. Math. Imag. Vis.*, 63(7):923–952, 2021.
- [13] K. POLISANO, M. CLAUSEL, V. PERRIER et L. CONDAT : Texture modeling by gaussian fields with prescribed local orientation. *In IEEE ICIP*, pages 6091–6095, Paris, France, 2014.
- [14] M. POURAHMADI : *High-Dimensional Covariance Estimation : With High-Dimensional Data*. Wiley Series in Probability and Statistics. Wiley, 2013.
- [15] F.J.P. RICHARD : Anisotropy of Hölder Gaussian random fields : characterization, estimation, and application to image textures. *Statist. Comput.*, 28(6):1155–1168, 2018.
- [16] I.W. SELESNICK, R.G. BARANIUK et N.C. KINGSBURY : The Dual-Tree Complex wavelet transform. *IEEE Signal Process. Mag.*, 22(6):123–151, 2005.
- [17] C. M. STEIN : Estimation of the mean of a multivariate normal distribution. *Ann. Stat.*, 9(6):1135–1151, 1981.
- [18] H. ZOU, T. HASTIE et R. TIBSHIRANI : On the “degrees of freedom” of the lasso. *Ann. Stat.*, 35(5):2173 – 2192, 2007.