

# EchoCem: évaluation de la qualité du ciment par segmentation d'images ultrasoniques par SLB

Félix Fourreau Mathis Wauquiez

École des Ponts ParisTech (ENPC)

ENS Paris-Saclay

{felix.fourreau, mathis.wauquiez}@eleves.enpc.fr

Mars 2025

## Abstract

*We address the challenge of semantic segmentation in ultrasonic imaging for the detection of tubage and Third Interface Echo (TIE) in oil wells. While semantic segmentation is a well-studied machine learning problem, numerous approaches exist, each with varying suitability depending on the data characteristics. Our work focused on designing a method tailored to the physical nature of our images and benchmarking it against alternative approaches. We first implemented deep learning models, such as U-Nets, before exploring physics-based models and various preprocessing and postprocessing techniques to enhance segmentation performance.*

## 1. Introduction

Ce rapport décrit notre travail, réalisé dans le cadre du challenge "EchoCem: évaluation de la qualité du ciment par segmentation d'images ultrasoniques par SLB", dans le contexte du cours "Génération de données en IA par transport et débruitage" du Professeur Stéphane Mallat. Nous allons d'abord mener une analyse des données à notre disposition, avant d'expliquer les différentes méthodologies que nous avons pu explorer, puis donner un point de vue critique sur nos méthodes et les résultats que nous avons obtenus. L'entièreté de notre code est disponible sur notre page GitHub: <https://github.com/mathis-wauquiez/EchoCemDataChallenge>.

### Contexte et objectifs

**Contexte** SLB est une entreprise mondiale, présente dans plus de 100 pays, axée sur l'innovation énergétique, notamment dans les secteurs du pétrole et du gaz. Dans ce con-

texte, SLB a besoin d'évaluer l'état physique de plusieurs puits de pétrole. Plus précisément, un puit de pétrole est constitué de plusieurs tuyaux métalliques, chacun séparé par une couche de ciment. Ces tuyaux métalliques sont soumis à de fortes contraintes mécaniques. Celles-ci se décomposent en une pression externe, exercée par la roche environnante et les fluides des formations géologiques, des charges axiales, dues au poids des tubes, et des contraintes de cisaillement. Les tuyaux peuvent également être soumis à des contraintes thermiques, notamment à de grandes profondeurs. Ils peuvent aussi souffrir de corrosion, de sulfidation et d'érosion.

**Objectifs** Détecter ces deux surfaces est un enjeu crucial, afin d'évaluer la qualité du ciment, pour surveiller la qualité du puits pendant le processus de forage. Un tube défaillant peut en effet entraîner des problèmes de stabilité du puits ou encore des risques environnementaux dus à des fuites.

## 2. Description des données

**Acquisition** Les données à notre disposition sont acquises par un scanner d'isolation. Celui-ci est un capteur par ultrason actif, qui fonctionne par écho d'impulsion: un émetteur envoie une impulsion sous forme d'onde flexurale, qui va être partiellement réfléchi à chaque changement de milieu. Ainsi, elle sera d'abord réfléchi par le tubage, puis par la formation. La réponse du capteur est donc constituée de deux "pics" d'intensité, correspondant à ces deux réflexions. Ces mesures unidimensionnelles sont prises à des intervalles de profondeurs réguliers, et sur diverses sections du puits.

**Description de la base de données** La base de données avec laquelle nous travaillons est constituée d'images ultrasoniques de 11 puits, ainsi que de leurs masques de seg-

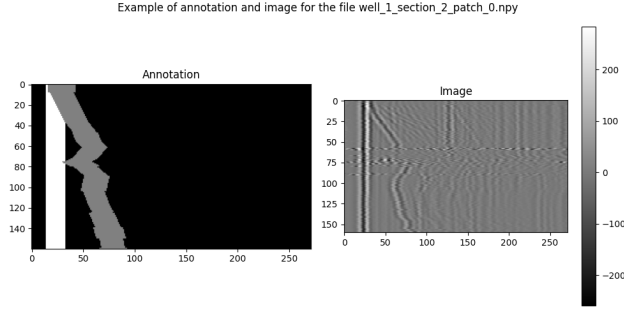


Figure 1. Exemple d'un patch et de son annotation

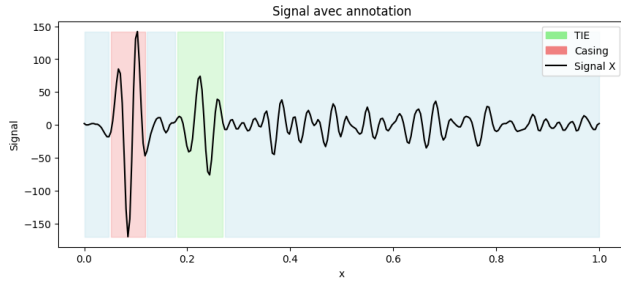


Figure 2. Un signal et son annotation

mentation. Les masques de segmentation nous indiquent où dans l'image sont situés le tubage et le TIE, que nous cherchons à segmenter. Ces données sont par la suite découpées en patches de taille  $160 \times 160$  et  $160 \times 272$ . Nous avons donc à notre disposition un total de **4410 patches**, stockés sous l'extension `.npz`, et nommés selon le format suivant : `well_n_section_mpatch_l.npz`.

Nous disposons également de **1980 patches** sans annotations, pouvant servir à des fins d'apprentissage non supervisé, ainsi qu'une base de données de test, constituée de **972 patches**.

**Structure d'un patch** Les patches que nous considérons sont un empilement vertical de signaux captés par le scanner d'isolation. Ils reflètent donc la nature physique de l'acquisition du signal. Notamment, ils reflètent comme les ondes flexurales envoyées par l'émetteur vont se propager dans le puits puis retourner au capteur. Ces informations sont d'une importance cruciale pour notre problème: en analysant finement les informations renvoyées par le capteur, nous sommes capables de localiser le tubage et le TIE dans nos images. Les signaux que nous obtenons sont la somme de diverses réflexions, comme illustré dans la figure 1, dans lequel nous observons d'abord la réflexion par le tubage, puis par le TIE, ainsi que beaucoup de bruit et des résonances. De plus, de manière aléatoire, le signal est parfois légèrement décalé. 1 2

**Prétraitement des données** La première chose que nous avons faite est de changer la façon dont sont découpés les patches. En effet, certaines méthodes, notamment de deep learning, pourraient mal généraliser, du fait que les patches utilisés sont toujours découpés de la même façon et que nous ne disposons pas d'assez de données. Nous avons donc agrégé les patches en grandes images, avec une image par puits et par section. Puis, nous avons créé une classe de jeu de données Pytorch, composée de chaque découpage de taille  $160 \times 272$  possible à partir de ces images, ce qui diminue la possibilité d'un biais lors de l'ajustement de nos modèles. Pour ce qui est des patches de taille  $160 \times 160$ , nous avons simplement prolongé les annotations et les données en les continuant par des zéros. Les signaux furent ensuite normalisés par la formule suivante:

$$\tilde{s}[t] = \frac{s[t] - \mu}{\sigma}$$

Avec  $\mu$  et  $\sigma$  respectivement la moyenne et l'écart-type des signaux. Nous ne ferons plus la distinction entre  $x[t]$  et  $\tilde{x}[t]$  dans la suite de ce rapport.

### 3. Approches

#### 3.1. Première approche: Deep Learning

Notre première approche pour segmenter nos données a été d'utiliser un modèle de type U-Net [10], une architecture encodeur-décodeur largement reconnue en segmentation d'images. Ce modèle combine les informations des niveaux d'encodage et de décodage pour capturer à la fois le contexte global et les détails fins. Nous avons ajouté des connexions résiduelles afin de limiter les problèmes de gradients explosants ou disparaissants.

De plus, nous avons intégré des blocs d'attention inspirés de [9] dans la couche la plus profonde du réseau. Ce bloc d'attention permet au modèle de mieux se concentrer sur les régions pertinentes à segmenter.

Le modèle U-Net est structuré comme suit :

- Bloc de convolution (ConvBlock) : deux convolutions 2D (kernel=(3,3), biais=True) séparées par une normalisation par batch (BatchNorm2D) et une activation ReLU.
- Bloc de convolution ascendante (UpConvBlock) : une opération d'upsampling (facteur=2), suivie d'une convolution 2D (kernel=(3,3)), d'une normalisation par batch (BatchNorm2D) et d'une activation ReLU.
- Bloc d'attention (AttBlock) : combine les caractéristiques du signal d'entrée et du signal de référence par des convolutions 2D (kernel=1), une addition suivie d'une activation ReLU, puis une convolution supplémentaire avec une activation sigmoïde. Ce résultat est ensuite multiplié par l'entrée pour effectuer un filtrage adaptatif. Cette architecture est directement tirée de cet article [9].

Notre U-Net a une profondeur de 5 niveaux avec un facteur de croissance des filtres égal à 2, en partant de 16 filtres de base. L'entraînement du modèle est réalisé avec un optimiseur Adam [7]. Les classes n'étant pas balancées, nous avons utilisé comme fonction de perte un mélange entre une `dice loss` [11] et la `Lovász-Softmax loss` [2]. Les résultats de ce réseau sont décrits dans la section 4.

### 3.2. Deuxième approche: Modélisation physique

En s'intéressant aux données à notre disposition, on se rend rapidement compte que:

- **Le problème est relativement simple.** Les signaux sont facilement identifiables, et les ondes retournées par les parois prennent toujours la même forme, les rendant facilement détectable.
- **Il y a un écart de domaine entre les puits d'entraînement et ceux de test.** En effet, il semblerait que les distributions des données d'entraînement et de test soient relativement différentes, notamment sur les puits 10. Celui-ci possède des données beaucoup plus régulières, et des profondeurs pour lesquelles nous n'observons pas de TIE.

Un modèle de deep learning n'est donc peut être pas idéal pour résoudre ce problème. En effet, un modèle trop compliqué risquerait de faire du surapprentissage: il pourrait performer très bien sur les puits du jeu d'entraînement, mais mal sur les données de test. Cela motive donc l'utilisation d'algorithmes nécessitant moins de paramètres. Du fait de la nature physique des signaux à notre disposition, nous proposons une modélisation mathématique explicite des signaux reçus.

Plus précisément, nous définissons notre modèle de la façon suivante:

$$s(x, t) = A(x)R(t - t_0(x)) + \sum_{i=1}^n B_i(x)R(t - t_0 - i\Delta t(x))$$

Dans cette expression,

- $x$  est la profondeur
- $t$  est l'axe horizontal de nos données
- $R(t)$  est une onde réfléchie
- $A(x)$  est l'amplitude de l'onde réfléchie par le tubage à la profondeur  $x$
- $B_i(x)$  est l'amplitude de la  $i$ -ème réflexion de l'onde due à la transition du ciment à la formation
- $t_0(x)$  est le temps que l'onde réfléchie par le tubage met pour arriver au capteur à la profondeur  $x$
- $\Delta t(x)$  est le temps que prend l'onde réfléchie par la formation met pour arriver au tubage à la profondeur  $x$

Ce modèle permet de modéliser les réflexions successives des ondes. Il nécessite cependant de trouver la forme  $R(t)$  qu'une onde réfléchie prend, en supposant qu'elle soit la même pour chaque réflexion. Il nécessite également de

déterminer, une fois  $R(t)$  connu, l'amplitude des réflexions  $A(x)$  et  $B_i(x)$  ainsi que  $t_0(x)$  et  $\Delta t(x)$ .

Notre travail a alors consisté à trouver des méthodes pour déterminer numériquement ces valeurs.

**Première étape: trouver  $R(x)$**  Nous proposons une méthode permettant de déterminer la forme que prend la réflexion d'une onde,  $R(x)$ . Celle-ci est basée sur l'apprentissage de dictionnaire convolutif [5]. Plus précisément, nous essayons de résumer nos signaux  $s(x, t)$  à une représentation éparse, en trouvant une fonction  $R(t)$  tel que le signal puisse s'écrire comme une somme discrète de versions décalées et pondérées de cette fonction. Nous étendons notre modélisation à un nombre  $K$  de fonctions. Notre modélisation est donc la suivante:  $\tilde{s}(x, t) = \sum_{k=1}^K (\mathbf{z}_k(x) * \mathbf{d}_k)(t)$ , avec  $\{z_k\}_{k \in [1, K]}$  les activations de nos **atomes**  $\{d_k\}_{k \in [1, K]}$ . Nous déterminons alors nos atomes et nos activations en résolvant le problème suivant:

$$\begin{aligned} \min_{(\mathbf{d}_k), (\mathbf{z}_k)} \quad & \frac{1}{2} \sum_{x=1}^N \left\| \mathbf{s}(x) - \sum_{k=1}^K \mathbf{z}_k(x) * \mathbf{d}_k \right\|^2 \\ & + \lambda \sum_{x=1}^N \sum_{k=1}^K \|\mathbf{z}_k(x)\|_1 \\ \text{s.c.} \quad & \|\mathbf{d}_k\|^2 \leq 1 \end{aligned}$$

Où  $\lambda > 0$  est notre coefficient de pénalité, et les normes sont temporelles. Ce problème n'est pas convexe par rapport au couple  $(\mathbf{d}_k)_k, (\mathbf{z}_k)_k$  mais convexe quand l'on considère chaque sous-problème individuellement (on reconnaît un problème des moindres carrés linéaire quand on minimise par rapport à  $d_k$ , et une régression Lasso quand on minimise par rapport à  $z_k$ ). Cela mène naturellement à un algorithme d'optimisation alternée, utilisé par [6], dans lequel les auteurs implémentent cet algorithme et le distribuent en tant que module python, que nous avons utilisé.

L'atome que nous avons déterminé 3 ressemble bien aux réponses impulsionnelles identifiées dans la figure 2. Nous observons que notre atome ressemble très fortement à une ondelette de chapeau mexicain (au signe près), très courante en traitement du signal.

### Deuxième partie: Apprendre les paramètres du modèle

La seconde tâche consiste à estimer les paramètres  $A(x)$ ,  $B_i(x)$ ,  $t_0(x)$  et  $\Delta t(x)$  du modèle. Pour cela, nous résolvons un problème de moindres carrés régularisé, formulé comme suit :

$$\begin{aligned} \min_{A(x), B_i(x), t_0(x), \Delta t(x)} \quad & \|\tilde{s}_{A, B_i, t_0, \Delta t}(x, t) - s(x, t)\|_2^2 + \\ & \lambda_1 \text{TV}_x(\Delta t) + \lambda_2 \text{TV}_x(t_0) \end{aligned}$$

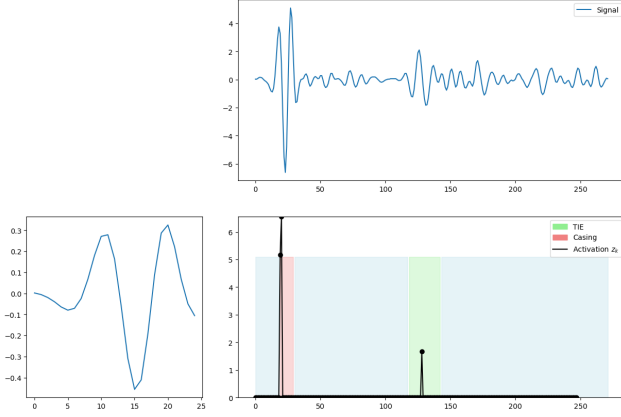


Figure 3. Atome calculé ainsi que son activation

où  $TV(f) = \int \left| \frac{df}{dt} \right| dt$  représente la variation totale de la fonction  $f$ .

L'utilisation des variations totales dans ce modèle a pour objectif d'intégrer un à priori dans nos solutions. En effet, nous savons que l'origine temporelle de nos signaux est parfois décalée. Or, les paramètres  $t_0(x)$  et  $\Delta t(x)$  sont censées refléter la nature majoritairement continue du mur. Nous encodons donc cette nature continue en pénalisant les changements importants de ces valeurs, ce qui devrait rendre nos prédictions résilientes aux décalages d'origine temporelle aléatoires.

Plusieurs approches sont envisageables pour résoudre ce problème. Nous avons opté pour une minimisation alternée : à  $\Delta t$  et  $t_0$  fixés, l'optimisation de  $A$  et des  $B_i$  se ramène à un problème classique de régression Lasso.

Cependant, bien que cette première étape soit relativement simple, l'optimisation de  $t_0(x)$  et  $\Delta t(x)$  pour des valeurs fixées de  $A(x)$  et  $B_i(x)$  est nettement plus complexe. Nous avons initialement tenté une descente de gradient, en utilisant les activations de l'atome pour initialiser  $t_0$  et  $\Delta t$ . Cette approche s'est avérée inefficace, car cette initialisation correspondait déjà à un minimum local du modèle. Une tentative d'initialisation aléatoire n'a pas été plus concluante : les coefficients optimaux  $A$  et  $B_i$  obtenus étaient proches de zéro, ce qui menait à des activations nulles lors de la minimisation alternée. En conséquence, nous n'avons pas réussi à surmonter cet obstacle, limitant notre exploration de cette approche.

Nous restons toutefois persuadés qu'une modélisation physique du problème pourrait mener à des résultats plus généralisables et fiables qu'une approche par réseaux de neurones. Cependant, nous devons avouer qu'en continuant sur cette piste, nous aurions rencontré quelques problèmes:

1. Notre modèle nous donne une estimation ponctuelle de la position du tubage et du TIE. Il nous aurait fallu trouver une règle pour déterminer la largeur de ceux-ci.

2. La régularisation par variations totales pourrait, dans les zones où la position du tubage et du TIE varient fortement, dégrader les résultats obtenus.

## 4. Combinaison des deux méthodes

### 4.1. Modification du prétraitement des données

Pour éviter le surapprentissage observé dans l'entraînement du UNet nous avons légèrement modifié l'obtention des données. Afin de standardiser les patches, ceux de taille initiale  $160 \times 160$  ont été transformés en  $160 \times 272$  par padding gaussien sur l'image normalisée (bruit gaussien de moyenne 0 et d'écart-type égal à la std de l'image). Les annotations initialement remplies avec des valeurs  $-1$  ont été remplacées par 0. Cette approche préserve la distribution statistique d'origine et évite de polluer l'apprentissage avec des valeurs artificielles.

Nous avons de la même manière reconstruit les images originales en recollant les patches afin de pouvoir ensuite redécouper librement ces images complètes. Dans ce cas là, nous avons choisi de produire des patches carrés de taille  $272 \times 272$ , plus pratiques pour certaines analyses fréquentielles.

Afin d'éviter le surapprentissage nous n'avons pas extraits tous les patches possibles de l'image entière mais seulement  $\left\lfloor \frac{H}{\text{taille}_{\text{patch}}} \times N \right\rfloor$  patches par image (avec  $\text{taille}_{\text{patch}} = 272$  et  $N$  un coefficient multiplicateur). Les patches sont extraits séquentiellement de haut en bas avec un incrément de  $\frac{H}{N \times \text{taille}_{\text{patch}}}$  entre chaque extraction.

**data augmentation :** Afin d'augmenter la diversité des données, les patches sont soumis à une symétrie horizontale avec une probabilité de 50 % et à l'ajout d'un léger bruit gaussien durant l'entraînement. L'objectif est de varier l'orientation et la texture des patches, tout en conservant des caractéristiques globales (par exemple, la position du début du coffrage à gauche de l'image).

### 4.2. Pre processing fréquentiel :

Pour mieux exploiter la structure locale des données, nous avons testé plusieurs approches fréquentielles. L'idée n'est plus de faire la classification à partir d'une approche fréquentielle analytique seule, mais plutôt de se servir de ces méthodes comme *pré-traitement* pour ensuite utiliser le modèle de *deep learning* (U-Net). Nous avons expérimenté trois méthodes différentes : la transformée de Fourier 2D, la transformée en ondelettes et les filtres de Gabor.

#### Transformée de Fourier

Notre première approche consistait à utiliser la décomposition fréquentielle de Fourier pour essayer de récupérer les détails intéressants de l'image, en particulier les vaguelettes correspondant aux parois du coffrage du puits. La transformée de Fourier discrète en 2D s'écrit :

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j 2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)}.$$

L'idée est d'arriver à sélectionner une bande de fréquence idéale qui permette de se concentrer uniquement sur les ondes intéressantes. En appliquant une décomposition en 8 bandes sur l'intervalle de fréquence de la décomposition nous obtenons :

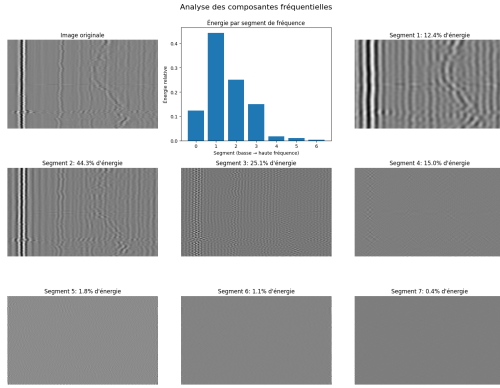


Figure 4. Enter Caption

Nous ne sommes jamais vraiment parvenu à obtenir une bande de fréquence qui ne conservait uniquement que les ondes associées aux réflexions sur les parois. Plusieurs raisons expliquent cet échec :

- **Perte de localisation spatiale** : la Transformée de Fourier est globale et mélange l'information spatiale, ce qui rend difficile la localisation précise.
- **Non-périodicité des données** : les patches ne sont pas périodiques et présentent des variations abruptes, générant des artefacts aux bords.
- **Signature fréquentielle complexe** : les vaguelettes du coffrage ne sont pas uniformes et varient localement, rendant un filtrage global peu efficace.

#### Transformée en ondelettes (Wavelets)

Contrairement à la TF qui opère globalement, la transformée en ondelettes fournit une décomposition à plusieurs échelles tout en conservant une localisation spatiale. L'idée générale est d'exprimer le signal (ou l'image) comme une somme de fonctions mères translatées et dilatées [3, 8] :

$$f(x) = \sum_k a_{j_0,k} \phi_{j_0,k}(x) + \sum_{j=j_0}^J \sum_k d_{j,k} \psi_{j,k}(x),$$

où  $\phi$  est la fonction d'échelle et  $\psi$  la fonction d'ondelette.

**Sélection de la meilleure ondelette.** Pour une image, à chaque niveau de décomposition, on sépare les basses fréquences (approximation) et les hautes fréquences

(détails) selon deux directions (horizontale et verticale). On obtient ainsi quatre sous-bandes :

$$\begin{cases} \text{BF-BF} & (\text{basses fréquences horizontales et verticales}), \\ \text{BF-HF} & (\text{basses fréquences horizontales, hautes verticales}), \\ \text{HF-BF} & (\text{hautes fréquences horizontales, basses verticales}), \\ \text{HF-HF} & (\text{hautes fréquences horizontales et verticales}). \end{cases}$$

En itérant la décomposition sur la bande BF-BF, on construit un arbre multi-niveaux permettant de capturer les détails à différentes échelles.

Nous avons testé la reconstruction sur 50 images à l'aide de différentes familles d'ondelettes, notamment Daubechies, Symlets, Coiflets, Biorthogonal, Reverse Biorthogonal et Discrete Meyer. La liste exacte des ondelettes testées inclut db1, db2, ..., db20, sym2, ..., sym20, coif1, ..., coif5, bior1.1, ..., bior6.8, rbior1.1, ..., rbior6.8, dmey, etc. (voir [1]).

Dans 45 cas sur 50, la meilleure qualité de reconstruction a été obtenue avec `bior2.2`. Cette ondelette correspond à la famille biorthogonale, elle utilise des filtres d'analyse et de synthèse distincts, et est optimisée pour réduire la distorsion et mieux préserver certaines discontinuités. La `bior2.2` est définie par deux paires de filtres (décomposition et reconstruction).

On obtient alors sur le même exemple que pour Fourier :

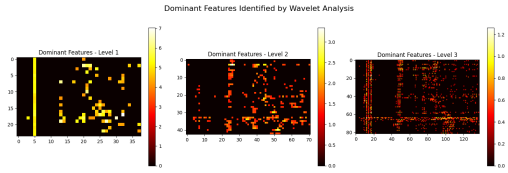


Figure 5

De manière générale, les résultats sont visuellement bien meilleurs que ceux obtenus par la TF simple, car la localisation spatiale est préservée. Et d'autre part l'ondelette utilisée ressemble beaucoup à la manière dont on observe les vaguelettes sur l'image du scan, en effet la structure principale de l'ondelette correspond à l'enchaînement des valeurs  $[-1, 2, -1]$ , comme on l'a vu dans la partie précédente. La figure 6 illustre cette ressemblance, en superposant l'atome déterminé dans la section précédente et l'ondelette que nous venons de choisir. Cela crée un lien fort entre les deux parties: on peut considérer la seconde comme une extension multiéchelle du travail que nous avons pu mener dans la partie précédente, en se restreignant toutefois à une classe d'ondelettes (on n'apprend pas l'ondelette).

#### Filtre de Gabor

Nous avons également expérimenté le filtre de Gabor [4], déjà utilisé en computer vision pour l'extraction de motifs



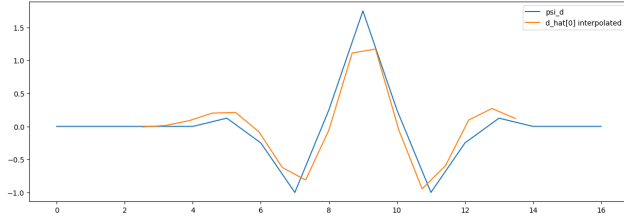


Figure 6. En bleu l'ondelette utilisée, en orange l'atome déterminé dans la partie précédente (à l'amplitude et l'échelle près)

(par exemple en imagerie médicale). Contrairement à la TF purement globale, il permet une représentation locale. Il est défini mathématiquement par :

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right),$$

où  $(x', y')$  est un système de coordonnées orienté de l'angle  $\theta$ . Ce filtre combine une gaussienne (pour la localisation spatiale) et une sinusoïde (pour la sélectivité fréquentielle).

**Choix des paramètres.** Nous avons utilisé l'implémentation de `scikit-image` [12] (`gabor_kernel`), avec les paramètres suivants :

- **fréquences** :  $\{0.08, 0.1, 0.12\}$  (obtenues expérimentalement),
- **angles** :  $\theta \in \{\frac{\pi}{8}k \mid k \in [1 : 7]\}$ , en excluant 0 et  $\pi$  pour ignorer les détails purement horizontaux,
- **sigma** = 3 (issu d'une approximation de  $\lambda/\pi$  pour la gaussienne).

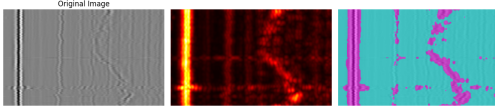


Figure 7. Exemple d'application de plusieurs filtres de Gabor sur un patch.

Sur la Figure 7, l'image du milieu correspond à la combinaison de toutes les orientations et fréquences, et l'image de droite montre une sélection de composantes dont le poids normalisé est supérieur à 0.15. On remarque que le résultat semble excellent pour l'exemple présenté, mais il est très sensible au choix du *threshold*. En pratique, cette méthode seule n'est pas suffisante pour une segmentation robuste : il reste beaucoup de bruit et la sensibilité aux hyperparamètres est importante.

### 4.3. UNET Enrichi

Au vu des résultats, l'approche retenue consiste à conserver l'architecture U-Net introduite plus tôt, mais en modifiant les canaux d'entrée. Concrètement, au lieu d'alimenter le réseau avec l'image brute, on a :

- La transformée de Gabor avec les hyperparamètres donnés avant
- Les coefficients de la transformée en ondelettes (`bior2.2` à 3 niveaux).

Le modèle de segmentation apprend alors à exploiter ces canaux pour mieux détecter le coffrage.

L'architecture d'un U-Net est d'autant plus justifiée qu'elle a la capacité de modéliser par elle-même les bases d'ondelettes et d'atomes que nous avons explorées grâce aux convolutions et représentations multi-échelles intrinsèques à ce type de modèle, et de filtrer dans cette base, grâce aux non-linéarités ReLU. En cela, la classe de fonctions de U-Nets est plus riche que les classes que nous avons explorées au cours des deux dernières méthodes, bien que l'on n'ait aucune garantie de converger vers un minimum satisfaisant. En donnant comme informations les données de la transformée de Gabor et les coefficients de la transformée en ondelettes, nous espérons obtenir un modèle qui converge vers un meilleur minima local, basé sur des caractéristiques du signal plus informatives et généralisables.

## 5. Post Processing

Une fois le modèle entraîné, nous l'appliquons sur les données de test en reproduisant le même pipeline de pré-traitement. Nous reconstruisons l'image totale du puits à partir des patches, puis effectuons l'inférence du modèle sur tous les patches possibles, de haut en bas. Pour chaque pixel, nous agrégeons (somme) les probabilités prédites sur les différentes occurrences de patches qui le contiennent. Nous normalisons ensuite ligne par ligne, car les pixels situés en haut ou en bas de l'image reçoivent moins de prédictions. Enfin, nous effectuons un post-traitement *spécifique à chaque classe* pour corriger les erreurs fréquentes.

### 5.1. Post-traitement de la classe 2 (paroi du casing)

La classe 2 correspond à la paroi du casing en contact avec le ciment. Sur nos données, elle apparaît globalement comme une ligne verticale positionnée à gauche de l'image. Elle est parfois incomplète en haut ou en bas, mais demeure continue. Les corrections suivantes ont été implémentées :

- **Position latérale attendue** : Les points associés à la classe 2 en dehors de l'intervalle position moyenne  $\pm 3$  écarts-types sur la ligne sont supprimés.
- **Seuil de probabilité** : Si la probabilité associée à ces points (après normalisation) est inférieure à 10%, ils sont supprimés.
- **Continuité verticale** : Si l'écart entre deux groupes de points de classe 2 sur la même colonne est inférieur à  $H/100$ , ils sont connectés (pour combler les petits trous).

### 5.2. Post-traitement de la classe 1

La classe 1 (souvent associée à des zigzags ou zones d'intérêt plus diffuses) ne présente pas de contrainte de con-

tinuité ni de position latérale stricte. Les corrections sont plus limitées :

- **Conflit entre deux composantes** : Si deux ensembles de pixels (connexes entre eux mais discontinus entre eux) coexistent sur une même ligne, on ne conserve que celui dont la somme de probabilité normalisée est la plus élevée.
- **Seuil de probabilité** : Si la probabilité associée à la classe 1 est inférieure à 1%, ces points sont supprimés.
- **Conflit avec la classe 2** : Si des pixels de classe 1 coexistent au même endroit que la classe 2, la classe 2 est privilégiée (suppression des +1).

### 5.3. Prédictions finales

La prédiction finale est obtenue en combinant les classes 0 (fond ou zone neutre), 1 (zones d'intérêt) et 2 (paroi). Après les suppressions et fusions, les classes 1 et 2 sont normalement disjointes.

## 6. Resultats

Sur les données de test, la comparaison entre le U-Net de base et le U-Net enrichi (avec canaux issus de Gabor et d'ondelettes) donne les scores suivants :

UNet (basique)	UNet enrichi
0.57	0.62

Table 1. Comparaison IoU moyenne sur le jeu de test.

Remarquons cependant que les résultats fonctionnent mieux sur certains puits que sur d'autres. Si le puits 8 semble bien fonctionner, le puits 10 en revanche donne des annotations parcellaires cela doit être du au fait que le contraste pour l'image du puits 10 est beaucoup plus faible comparé à tous ceux de l'entraînement. D'autres part le puits 10 possède une classe 1 discontinue, l'enjeu est de déterminer correctement où elle l'est réellement et où s'il s'agit simplement du faible contraste [8, 9]

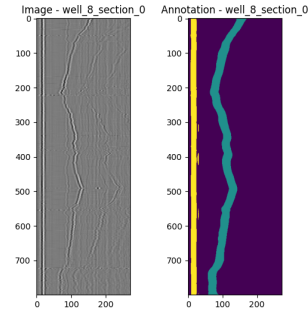


Figure 8. Prediction sur le puit 8

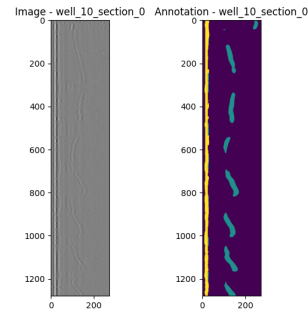


Figure 9. Prediction sur le puit 10

## 7. Conclusion

Dans ce travail, nous avons exploré différentes approches pour la segmentation d'images ultrasoniques dans le contexte de l'évaluation de la qualité du ciment dans les puits pétroliers. Cette étude nous a permis de mettre en évidence plusieurs points importants.

Premièrement, nous avons constaté que la nature physique des données joue un rôle crucial dans le choix des méthodes appropriées. Notre approche initiale par apprentissage profond avec un U-Net a fourni des résultats acceptables, mais présentait des limites en termes de généralisation entre les différents puits. Cela illustre un problème classique en apprentissage automatique : les modèles complexes peuvent suroptimiser sur les données d'entraînement sans capturer les invariants physiques sous-jacents.

Notre tentative de modélisation physique explicite, bien que théoriquement prometteuse, s'est heurtée à des obstacles d'optimisation. L'extraction d'un atome représentatif par apprentissage de dictionnaire convolutif a néanmoins révélé une structure remarquablement similaire à une ondelette de chapeau mexicain, confirmant ainsi notre intuition sur la nature ondulatoire des signaux.

La combinaison d'approches - prétraitement par transformées en ondelettes et filtres de Gabor alimentant un U-Net - s'est révélée être notre solution la plus efficace, avec une amélioration significative de l'IoU moyen (de 0,57 à 0,62). Ce résultat souligne l'importance d'incorporer des connaissances du domaine dans les données à disposition de

nos modèles. En fournissant au réseau des représentations physiquement pertinentes, nous guidons l'optimisation vers des solutions plus généralisables.

Les techniques de post-traitement spécifiques à chaque classe ont également joué un rôle important dans l'amélioration des résultats finaux, en intégrant des contraintes physiques comme la continuité verticale pour la classe 2. Cependant, les différences de performance entre les puits (notamment le puits 10 avec son faible contraste) suggèrent qu'il reste une marge d'amélioration pour traiter la variabilité inter-puits.

Pour les travaux futurs, plusieurs pistes semblent prometteuses :

1. L'exploration de méthodes d'apprentissage de représentation non supervisées pour mieux gérer la variabilité entre puits.
2. Le raffinement du modèle physique avec des techniques d'optimisation plus robustes, et sa possible intégration dans un modèle d'apprentissage profond.

## References

- [1] Pywavelets – wavelet transforms in python. Online. <https://pywavelets.readthedocs.io>. 5
- [2] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks, 2018. 3
- [3] Ingrid Daubechies. Ten lectures on wavelets. Society for Industrial and Applied Mathematics (SIAM), 1992. 5
- [4] Dennis Gabor. Theory of communication. In: Journal of the IEE, vol. 93, no. 26, pp. 429–457, 1946. 5
- [5] Cristina Garcia-Cardona and Brendt Wohlberg. Convolutional dictionary learning: A comparative review and new algorithms. *IEEE Transactions on Computational Imaging*, 4(3):366–381, 2018. 3
- [6] Toni Giorgino. Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software*, 31(7):1–24, 2009. 3
- [7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 3
- [8] Stephane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989. 5
- [9] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas, 2018. 2
- [10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2
- [11] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. *Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations*, page 240–248. Springer International Publishing, 2017. 3
- [12] S. van der Walt, J. L. Schönberger, et al. scikit-image: image processing in python. In: *PeerJ*, vol. 2, e453, 2014. 6