

Application au contrôle non destructif ultrasonore d'un réseau  
de neurones convolutif pour identifier, localiser et  
dimensionner des défauts

par

Thibault LATÊTE

MÉMOIRE PAR ARTICLE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE  
SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA  
MAÎTRISE AVEC MÉMOIRE EN GÉNIE MÉCANIQUE  
M. Sc. A.

MONTRÉAL, LE 10 SEPTEMBRE 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC



Thibault Latête, 2020



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

**PRÉSENTATION DU JURY**

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

Pr. Pierre Bélanger, directeur de mémoire  
Département de génie mécanique à l'École de technologie supérieure

Pr. Ricardo Zednik, président du jury  
Département de génie mécanique à l'École de technologie supérieure

M. Guillaume Painchaud-April, examinateur externe  
Olympus OSSA

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 4 SEPTEMBRE 2020

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE



## **REMERCIEMENTS**

Je souhaite tout d'abord remercier le professeur Pierre Bélanger pour son accompagnement tout au long du projet. Rien n'aurait été possible sans ses conseils avisés, son suivi régulier et sa confiance. Travailler avec lui était un plaisir de chaque instant.

Je remercie ensuite le groupe Olympus OSSA pour leur financement ainsi que tous les employés qui ont contribué au projet. Une pensée spéciale pour Guillaume et Alain qui m'ont apporté toute leur expertise.

Mes salutations chaleureuses envers tous mes collègues, amis et même colocataires du PulÉTS ayant contribué à mon épanouissement au sein du laboratoire et pour leurs conseils éclairés, j'adresse toute ma gratitude à : Aubin, Nicolas, Damien, Aurélien, Geoffrey, Frédéric, Daniel, Jorge, Cherif, Guillaume, Victor, Anthony. Une pensée toute particulière envers Baptiste notamment pour son support technique sept jours sur sept, à toute heure du jour et de la nuit, ainsi qu'un remerciement singulier à Arthur et Jérôme pour leur initiation précieuse aux réseaux de neurones.

Merci également au partenariat entre l'École de Technologie Supérieure de Montréal et l'école d'ingénieurs française des Arts et Métiers. Merci à toutes les équipes pédagogiques et administratives qui soutiennent ce formidable cursus de double diplôme.

Un remerciement final à mes chers parents pour leur inépuisable soutien.



# **Application au contrôle non destructif ultrasonore d'un réseau de neurones convolutif pour identifier, localiser et dimensionner des défauts**

Thibault LATÊTE

## **RÉSUMÉ**

Les algorithmes d'apprentissage machine sont largement utilisés dans la reconnaissance d'images. Dans le domaine du contrôle non destructif par ultrasons multi-éléments, les images sont généralement formées par la superposition constructive et destructive de signaux réfléchis par des défauts ou des caractéristiques géométriques. Cependant, les images ainsi générées restent difficiles à interpréter. Dans cette étude, le Faster R-CNN, un réseau neuronal convolutif, a été utilisé pour identifier, localiser et dimensionner des trous à fond plat (FBH) et des trous à perçage latéral (SDH) dans un bloc immergé en utilisant une unique insonification à onde plane. L'entraînement a été effectué sur des données segmentées et classifiées, générées par des simulations par éléments finis accélérées par GPU. Des SDH et des FBH de différents diamètres, profondeurs et positions latérales ont été inclus dans l'ensemble de l'entraînement. L'épaisseur du bloc percé était également variable. Une sonde ultrason de 64 éléments a été simulée. Tous les éléments de la sonde ont émis simultanément et les traces temporelles de chaque élément ont été enregistrées. Les traces temporelles ont été concaténées pour former une matrice. Ces matrices de traces temporelles ont servi à l'entraînement du réseau neuronal. Cette méthode d'inspection permet une acquisition rapide des données au détriment de la faible résolution latérale de l'image résultante. Le réseau neuronal entraîné a été testé dans un premier temps à l'aide de simulations par éléments finis. Les résultats ont été évalués sur le critère de l'intersection de l'union (IoU) entre le véritable encadrement du bloc ou de son défaut et l'encadrement prédit. En ce qui concerne les cas simulés, l'épaisseur de l'échantillon a été détectée dans tous les cas. Un seuil d'IoU de 40 % a conduit à la détection de 87 % des FBH et seulement 20 % des SDH. Le plus petit FBH simulé était de 0,56 longueur d'onde de profondeur et de 1,04 longueur d'onde d'étendue latérale. En traçant un encadrement utilisant la méthode de la chute à -6 dB autour du FBH, l'IoU était toujours inférieure à 15 %. L'étendue latérale du FBH selon la méthode de la chute à -6 dB était en moyenne trois fois plus grande que le diamètre prévu par la méthode proposée. L'entraînement fut reconduit et terminé avec une petite base de données expérimentales (équivalente à 3% de la base de données simulées). En ce qui concerne les cas expérimentaux, les résultats montrent que l'échantillon d'essai a été correctement identifié dans tous les cas. Cette fois, un seuil d'IoU de 40 % a permis de détecter 70 % des FBH. Le plus petit FBH détecté expérimentalement avait une profondeur de 2 longueurs d'onde.

**Mots-clés:** Contrôle non destructif, Ultrason, Ondes planes, Réseau de neurones convolutif, Reconnaissance d'image, Faster-RCNN



# **Towards using convolutional neural network to locate, identify and size defects in phased array ultrasonic testing**

Thibault LATÊTE

## **ABSTRACT**

Machine learning algorithms are widely used in image recognition. In phased array ultrasonic testing, images are typically formed through constructive and destructive superpositions of signals backscattered from flaws or geometrical features. However, images generated in phased array ultrasonic testing remain difficult to interpret. In this study, the Faster R-CNN was used to identify, locate and size flat bottom holes (FBH) and side-drilled holes (SDH) in an immersed test specimen using a single plane wave insonification. The training was performed on segmented and classified data generated using GPU-accelerated finite element simulations. SDH and FBH of different diameters, depths and lateral positions were included in the training set. The thickness of the test specimen was also variable. An ultrasonic phased array probe of 64 elements was simulated. All elements of the phased array probe were fired at the same time and the time traces from each element were recorded. The individual time traces were concatenated to form a matrix, which was then used in the training. This inspection scenario enables fast acquisition of data at the expense of poor lateral resolution in the resulting image. The trained neural network was initially tested using finite element simulations. Results were assessed in terms of the intersection of the union (IoU) between the ground truth geometry and the predicted geometry. With the simulated cases, the thickness of the test specimen was detected in all cases. When using a 40% IoU threshold, the detection rate of the FBH was 87% while only 20% for the SDH. The smallest detected FBH had a 0.56 wavelength depth and a lateral extent of 1.04 wavelength. Drawing a box using the  $-6$  dB drop method around the FBH always led to an IoU under 15%. On average, the lateral extent of the FBH using the  $-6$  dB drop method was three times larger than the diameter predicted by the proposed method. Then, the training was continued with a small augmented experimental dataset (equivalent to 3% of the simulated dataset). In experiments, the results show that the test specimen was always correctly identified. When using a 40% IoU threshold the experimental detection rate of the FBH was 70%. The smallest detected defect in experiments had a depth of 2 wavelengths.

**Keywords:** Non Destructive Testing, Ultrasound, Plane Waves, Image recognition, Convolutional Neural Network, Faster-RCNN



## TABLE DES MATIÈRES

|                                                                                                                                           | Page |
|-------------------------------------------------------------------------------------------------------------------------------------------|------|
| INTRODUCTION .....                                                                                                                        | 1    |
| CHAPITRE 1 REVUE DE LITTÉRATURE .....                                                                                                     | 3    |
| 1.1 Introduction à la propagation d'ondes ultrasonores .....                                                                              | 3    |
| 1.1.1 Les équations .....                                                                                                                 | 3    |
| 1.1.2 Comportement d'un ultrason aux interfaces .....                                                                                     | 7    |
| 1.1.3 L'atténuation .....                                                                                                                 | 8    |
| 1.2 Modélisation par éléments finis des inspections ultrasonores .....                                                                    | 9    |
| 1.2.1 Équation élastodynamique discrète .....                                                                                             | 9    |
| 1.2.2 Frontières absorbantes .....                                                                                                        | 10   |
| 1.2.3 Taille des éléments et incrément temporel .....                                                                                     | 11   |
| 1.3 Méthodes de contrôle non destructif par ultrasons .....                                                                               | 12   |
| 1.3.1 Insonification par ondes planes .....                                                                                               | 12   |
| 1.3.2 Techniques d'imagerie ultrasonore .....                                                                                             | 14   |
| 1.3.2.1 A-Scan .....                                                                                                                      | 14   |
| 1.3.2.2 B-Scan .....                                                                                                                      | 14   |
| 1.3.2.3 Méthode de focalisation en tout point .....                                                                                       | 15   |
| 1.3.3 Chute à moins six décibels .....                                                                                                    | 17   |
| 1.4 Les réseaux de neurones .....                                                                                                         | 18   |
| 1.4.1 Le modèle du perceptron .....                                                                                                       | 18   |
| 1.4.2 Les réseaux de neurones convolutifs .....                                                                                           | 21   |
| 1.4.2.1 Principe de base .....                                                                                                            | 22   |
| 1.4.2.2 Les différents types de CNN .....                                                                                                 | 25   |
| 1.4.2.3 Le Faster R-CNN .....                                                                                                             | 27   |
| 1.4.2.4 RPN .....                                                                                                                         | 29   |
| 1.4.2.5 Réseau de détection .....                                                                                                         | 30   |
| 1.4.2.6 Entraînement, validation et test .....                                                                                            | 32   |
| 1.5 Bilan .....                                                                                                                           | 32   |
| CHAPITRE 2 TOWARDS USING CONVOLUTIONAL NEURAL NETWORK<br>TO LOCATE, IDENTIFY AND SIZE DEFECTS IN PHASED<br>ARRAY ULTRASONIC TESTING ..... | 35   |
| 2.1 Abstract .....                                                                                                                        | 35   |
| 2.2 Introduction .....                                                                                                                    | 36   |
| 2.3 Materials and methods .....                                                                                                           | 38   |
| 2.3.1 Finite Element simulated dataset .....                                                                                              | 38   |
| 2.3.2 Faster R-CNN adaptation and development .....                                                                                       | 41   |
| 2.3.3 Training the neural network with simulations .....                                                                                  | 43   |
| 2.3.4 Experimental setup .....                                                                                                            | 44   |
| 2.3.5 Additional training with experiments .....                                                                                          | 45   |

|                                         |                                               |    |
|-----------------------------------------|-----------------------------------------------|----|
| 2.3.6                                   | Performance Metrics .....                     | 45 |
| 2.4                                     | Results .....                                 | 47 |
| 2.4.1                                   | Tests of the simulation-only training .....   | 47 |
| 2.4.2                                   | Tests on experimental acquisitions .....      | 49 |
| 2.4.3                                   | Comparison with the $-6$ dB drop method ..... | 50 |
| 2.5                                     | Discussion .....                              | 51 |
| 2.6                                     | Conclusion .....                              | 54 |
| 2.7                                     | Acknowledgements .....                        | 54 |
| CONCLUSION ET RECOMMANDATIONS .....     |                                               | 55 |
| ANNEXE I COMPLÉMENTS DE RÉSULTATS ..... |                                               | 57 |
| BIBLIOGRAPHIE .....                     |                                               | 59 |

## LISTE DES TABLEAUX

|                                             | Page |
|---------------------------------------------|------|
| Tableau 2.1    Materials Properties .....   | 39   |
| Tableau 2.2    Model dimension ranges ..... | 41   |



## LISTE DES FIGURES

|             | Page                                                                                                                                                                                        |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Figure 1.1  | Les deux modes de propagation d'une onde acoustique de volume. .... 6                                                                                                                       |
| Figure 1.2  | Schéma de propagation d'une onde aux interfaces. (A) la propagation d'un milieu liquide vers un milieu solide et (B) la propagation d'un milieu liquide vers un autre milieu solide. .... 7 |
| Figure 1.3  | Émission et transmission d'une onde plane dans une pièce immergée. .... 13                                                                                                                  |
| Figure 1.4  | Schéma de l'acquisition d'un A-Scan (à gauche) et sa représentation (à droite). .... 14                                                                                                     |
| Figure 1.5  | Schéma de l'acquisition d'un B-Scan (à gauche) et sa représentation (à droite). .... 15                                                                                                     |
| Figure 1.6  | Schéma de l'acquisition d'une FMC. .... 16                                                                                                                                                  |
| Figure 1.7  | Schéma de la méthode de la chute à $-6$ dB. .... 17                                                                                                                                         |
| Figure 1.8  | Schéma du concept de perceptron à trois entrées. .... 19                                                                                                                                    |
| Figure 1.9  | Réseau de neurones entièrement connectés. .... 20                                                                                                                                           |
| Figure 1.10 | Différentes possibilités de prédictions d'un CNN. .... 21                                                                                                                                   |
| Figure 1.11 | Schéma d'une opération de convolution. .... 23                                                                                                                                              |
| Figure 1.12 | Schéma d'une opération de sous-échantillonnage. .... 23                                                                                                                                     |
| Figure 1.13 | Schéma d'architecture d'un VGG16. .... 25                                                                                                                                                   |
| Figure 1.14 | Exemple de prédiction issue d'un Faster R-CNN. .... 27                                                                                                                                      |
| Figure 1.15 | Schéma des étapes de fonctionnement d'un Faster R-CNN par le regroupement d'un CNN, d'un RPN et d'un réseau de détection. .... 28                                                           |
| Figure 1.16 | Exemple de ROIP sur deux fenêtres d'ancrage différentes. .... 31                                                                                                                            |
| Figure 2.1  | Experimental setup (A) and a schematic of FE model (B). .... 40                                                                                                                             |
| Figure 2.2  | Example of a neural network prediction from a raw TTM. The green boxes associated with their label, classify, locate and size the objects. .... 42                                          |

|             |                                                                                                                                                                       |    |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 2.3  | CNN architecture.....                                                                                                                                                 | 43 |
| Figure 2.4  | Learning curve and loss function as a function of epoch, starting with FE simulated dataset and following with experimental dataset. ....                             | 44 |
| Figure 2.5  | Example of two boxes (red and green) IoU. In this example the, IoU is 25% .....                                                                                       | 46 |
| Figure 2.6  | Neural Network test results over FE simulations. FBH and SDH testing results in function of SNR.....                                                                  | 47 |
| Figure 2.7  | ROC Curve of neural network test results over FBH FE simulations.....                                                                                                 | 48 |
| Figure 2.8  | Neural Network test results over FE simulations. FBH and SDH testing results in function of IoU Threshold with SNR fixed to 7 dB. ....                                | 49 |
| Figure 2.9  | Faster R-CNN predictions on an experimental plane-wave insonification TTM of an immersed aluminium block with an FBH. ....                                            | 50 |
| Figure 2.10 | Comparison between Faster R-CNN predictions and $-6$ dB drop method on an FE simulated plane-wave insonification TTM of an immersed aluminium block with an FBH ..... | 51 |

## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

|        |                                                                                              |
|--------|----------------------------------------------------------------------------------------------|
| CND    | Contrôle Non Destructif                                                                      |
| FBH    | Trou à fond plat ( <i>Flat-Bottom Hole</i> )                                                 |
| SDH    | Trou percé latéral ( <i>Side-Drilled Hole</i> )                                              |
| IoU    | Intersection de l'union ( <i>Intersection of Union</i> )                                     |
| GPU    | Processeur graphique ( <i>Graphics Processing Unit</i> )                                     |
| 3D     | Tridimensionnel                                                                              |
| 2D     | Bidimensionnel                                                                               |
| SNR    | Ratio signal sur bruit ( <i>Signal-to-Noise Ratio</i> )                                      |
| TFM    | Méthode de focalisation en tout point ( <i>Total Focusing Method</i> )                       |
| FMC    | Capture complète de la matrice ( <i>Full Matrix Capture</i> )                                |
| ReLU   | Unité de rectification linéaire ( <i>Rectified Linear Unit</i> )                             |
| CNN    | Réseau neuronal convolutif ( <i>Convolutional Neural Network</i> )                           |
| R-CNN  | Régions avec réseau neuronal convolutif ( <i>Regions with Convolutional Neural Network</i> ) |
| RPN    | Réseau de propositions de régions ( <i>Region Proposal Network</i> )                         |
| NMS    | Suppression du non-maximum ( <i>Non-Maximum Supression</i> )                                 |
| ROI    | Région d'intérêt ( <i>Region Of Interest</i> )                                               |
| ROIP   | Regroupement des régions d'intérêt ( <i>Region Of Interest Pooling</i> )                     |
| UT     | Contrôle Ultrasonore ( <i>Ultrasonic Testing</i> )                                           |
| TTM    | Matrice des traces temporelles ( <i>Time Trace Matrix</i> )                                  |
| FE     | Éléments finis ( <i>Finite Element</i> )                                                     |
| RGB    | Rouge-Vert-Bleu ( <i>Red-Green-Blue</i> )                                                    |
| ROC    | Fonction d'efficacité du récepteur ( <i>Receiver Operating Characteristic</i> )              |
| AUC    | Aire sous la courbe ( <i>Area Under the Curve</i> )                                          |
| Conv2D | Convolution bidimensionnelle ( <i>2 Dimensions Convolution</i> )                             |



## LISTE DES SYMBOLES ET UNITÉS DE MESURE

|                    |                                               |
|--------------------|-----------------------------------------------|
| $\text{dB}$        | Décibel                                       |
| $\text{Hz}$        | Hertz                                         |
| $\text{Pa}$        | Pascal                                        |
| $\text{m}$         | Mètre                                         |
| $\text{m s}^{-1}$  | Mètre par seconde                             |
| $\text{kg m}^{-3}$ | Kilogramme par mètre cube                     |
| $E$                | Module d'Young                                |
| $\nu$              | Coefficient de Poisson                        |
| $\lambda$          | Longueur d'onde                               |
| $c$                | Vitesse de propagation de l'onde              |
| $\rho$             | Densité                                       |
| $\sigma$           | Fonction d'activation                         |
| $w$                | Poids du perceptron ou du réseau de neurones  |
| $b$                | Biais du perceptron                           |
| $s$                | Sortie du perceptron ou du réseau de neurones |



## INTRODUCTION

Les techniques de contrôle non destructif (CND) sont utilisées aujourd'hui pour vérifier l'intégrité d'un matériau et identifier d'éventuels défauts dans des pièces ou des structures de toutes dimensions, sans pratiquer d'altération. Allant d'une pièce de composite aéronautique au silo de stockage pétrolier, la technique utilisée doit s'adapter au type de défaut recherché, à la géométrie de la pièce, au matériau, ou encore aux contraintes du terrain d'inspection. Le CND est utilisé en fin de production pour garantir une haute qualité mais aussi en tant que maintenance préventive, afin d'anticiper des réparations ou des remplacements et, par conséquent, de mieux maîtriser les coûts.

Le CND par ultrasons est une technique majeure d'inspection dans l'industrie. Elle se distingue des autres méthodes, comme les rayons X ou les courants de Foucault, par sa simplicité conceptuelle et la versatilité de ses instruments de mesure ainsi que la possibilité d'aller sur le terrain ou de résister à l'environnement difficile et exigeant d'une usine. Les grands axes de la recherche sur le contrôle ultrasonore sont l'amélioration des techniques d'imagerie ou le développement de capteurs plus performants.

Ces deux recherches de l'excellence ne sont pas sans conséquence. D'une part, une prise de mesure longue de plus en plus exigeante qui doit être réalisée au cas par cas par du personnel qualifié et d'autre part, des coûts de capteurs qui augmentent. L'objectif est l'obtention de la meilleure qualité d'image qui soit après avoir procédé à des calculs mathématiques lourds. Il apparaît donc rare ou très cher d'inspecter les pièces à forte valeur ajoutée issues d'une chaîne de production à haute cadence.

L'objectif de cette étude est double. Tout d'abord, la possibilité d'augmenter la cadence d'inspection par une technique d'examen simple et une imagerie sans reconstruction. Ensuite, la faculté d'automatiser l'identification du défaut à l'aide d'un réseau de neurones aussi performant qu'une inspection humaine mais plus rapide. En effet, aujourd'hui, les réseaux de neurones

atteignent des performances humaines, voire meilleures, dans le domaine de la vision par ordinateur pour distinguer les différents éléments d'un environnement. L'étude cherche donc à reproduire cette capacité pour les images issues du CND ultrasonore.

Afin de mener à bien ces objectifs, une large base de données d'inspections classifiées sera générée à partir de simulations par éléments finis. Un réseau de neurones capable d'être entraîné à partir de ces simulations sera ensuite développé. Une fois l'entraînement validé et le réseau de neurones testé, l'entraînement continuera sur une petite base de données expérimentales. Enfin, le réseau de neurones sera testé expérimentalement.

Le premier chapitre expose les acquis nécessaires en contrôle ultrasonore et en apprentissage machine à la réalisation de cette étude. Le second chapitre est constitué d'un article soumis au journal *Ultrasonics* décrivant les méthodes déployées pour atteindre les objectifs du sujet ainsi que les résultats obtenus. Finalement des conclusions sont tirées et des perspectives de travaux futurs sont exposés.

## CHAPITRE 1

### REVUE DE LITTÉRATURE

#### 1.1 Introduction à la propagation d'ondes ultrasonores

Les ondes ultrasonores sont étudiées depuis le 18<sup>e</sup> siècle. Elles font partie de la famille des ondes mécaniques. Elles concernent la propagation d'une oscillation de pression ou de contrainte dans un milieu non-vide (gazeux, liquide, solide). La fréquence de ces ondes est supérieure à la limite humaine audible de 20 kHz. De nombreux types d'ondes ultrasonores sont connues, parmi lesquelles, par exemple, les ondes de volume, les ondes guidées et les ondes de surface.

##### 1.1.1 Les équations

La première mise en équation de la propagation des ondes est réalisée au 18<sup>e</sup> siècle par d'Alembert, à partir des lois de Hooke et de Newton ainsi que de l'hypothèse selon laquelle les liaisons entre les particules d'un solide sont modélisables par un système masse-ressort (Cheeke, 2002). Ainsi, dans un milieu unidimensionnel, isotrope et homogène, l'équation de propagation est la suivante :

$$\frac{\partial^2 u(x, t)}{\partial t^2} = c^2 \frac{\partial^2 u(x, t)}{\partial x^2} . \quad (1.1)$$

Dans cette équation,  $u$  représente le déplacement de la particule,  $c$  la vitesse de propagation de l'onde,  $x$  la position, et  $t$  le temps. Cette équation possède une solution générale dans un milieu matériel :

$$u(x, t) = f(x - c \cdot t) + g(x + c \cdot t) \quad (1.2)$$

où  $f$  est une fonction de propagation arbitraire dans le sens des  $x$  positifs et  $g$  une autre fonction de propagation arbitraire dans le sens des  $x$  négatifs. Ramené au cas où la perturbation mécanique est sinusoïdale dans le sens des  $x$  positifs, l'équation de propagation devient :

$$u(x, t) = A \cos \left( \omega \left( t - \frac{x}{c} \right) \right) \quad (1.3)$$

avec  $A$  l'amplitude de l'onde et  $\omega$  sa pulsation. Euler étend alors cette équation dans le cas d'une onde plane, dans un milieu tridimensionnel (3D) solide et isotrope sans force extérieure appliquée (Shull, 2002) :

$$\rho \frac{\partial^2 u_j}{\partial t^2} = \sum_j \frac{\partial T_{ij}}{\partial x_j} \quad (1.4)$$

avec  $\rho$  la densité et  $[T]$  le tenseur de contrainte de coefficients  $T_{ij}$  issu de la loi de Hooke :

$$T_{ij} = C_{ijkl} \cdot S_{kl} \quad (1.5)$$

où  $[C]$  est le tenseur d'élasticité de coefficients  $C_{ijkl}$  et  $[S]$  le tenseur des déformations de coefficients  $S_{kl}$ . Dans un matériau homogène et isotrope, les coefficients  $T_{ij}$  se simplifient sous la forme suivante :

$$T_{ij} = \lambda S_{kk} \delta_{ij} + 2\mu S_{ij} \quad (1.6)$$

avec  $\lambda$  et  $\mu$  les paramètres de Lamé qui peuvent être définis par le module d'Young ( $E$ ) et le coefficient de poisson ( $\nu$ ) ou des coefficients de  $[C]$  :

$$\left\{ \begin{array}{l} E = \mu \frac{3\lambda+2\mu}{\lambda+\mu} \\ \nu = \frac{\lambda}{2(\lambda+\mu)} \\ C_{12} = \lambda \\ C_{44} = \mu = \frac{C_{11}-C_{12}}{2} \end{array} \right. . \quad (1.7)$$

En utilisant l'équation 1.6 dans l'équation 1.5 on obtient :

$$\rho \frac{\partial^2 u_i}{\partial t^2} = (\lambda + \mu) \frac{\partial S_{kk}}{\partial x_i} + \mu \nabla^2 u_i , \quad (1.8)$$

ce qui donne en notation vectorielle :

$$\rho \frac{\partial^2 \vec{u}}{\partial t^2} = (\lambda + \mu) \vec{\nabla} (\vec{\nabla} \cdot \vec{u}) + \mu \nabla^2 \vec{u} . \quad (1.9)$$

Grâce à la décomposition de Helmholtz, le vecteur  $\vec{u}$  se découple en un champ vectoriel  $\vec{\Psi}$  et un champ scalaire  $\Phi$  :

$$\vec{u} = \Phi \vec{\nabla} + \vec{\nabla} \times \vec{\Psi} = \vec{u}_L + \vec{u}_T \quad (1.10)$$

où  $\vec{u}_L$  est la composante de  $\vec{u}$  dans le plan de propagation (appelée composante longitudinale) et  $\vec{u}_T$  est la composante de  $\vec{u}$  normale au plan de propagation (appelée composante transversale). S'en déduisent alors deux équations indépendantes :

$$\left\{ \begin{array}{l} \frac{\partial^2 \Phi}{\partial t^2} = c_L^2 \nabla^2 \Phi \\ \frac{\partial^2 \vec{\Psi}}{\partial t^2} = c_T^2 \nabla^2 \vec{\Psi} \end{array} \right. . \quad (1.11)$$

Les équations des vitesses de propagations longitudinale ( $c_L$ ) et transversale ( $c_T$ ) s'expriment alors grâce aux équations suivantes :

$$\begin{cases} c_L = \sqrt{\frac{\lambda+2\mu}{\rho}} = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}} \\ c_T = \sqrt{\frac{\mu}{\rho}} = \sqrt{\frac{E}{2\rho(1+\nu)}} \end{cases} . \quad (1.12)$$

On peut remarquer que ces deux modes de propagations ne s'effectuent pas à la même vitesse. Ces équations démontrent ainsi l'indépendance de chacun de ces deux modes. Toute autre propagation d'ondes dans un matériau solide dérive de ceux-là. La Figure 1.1 illustre chacun de ces modes de propagation.

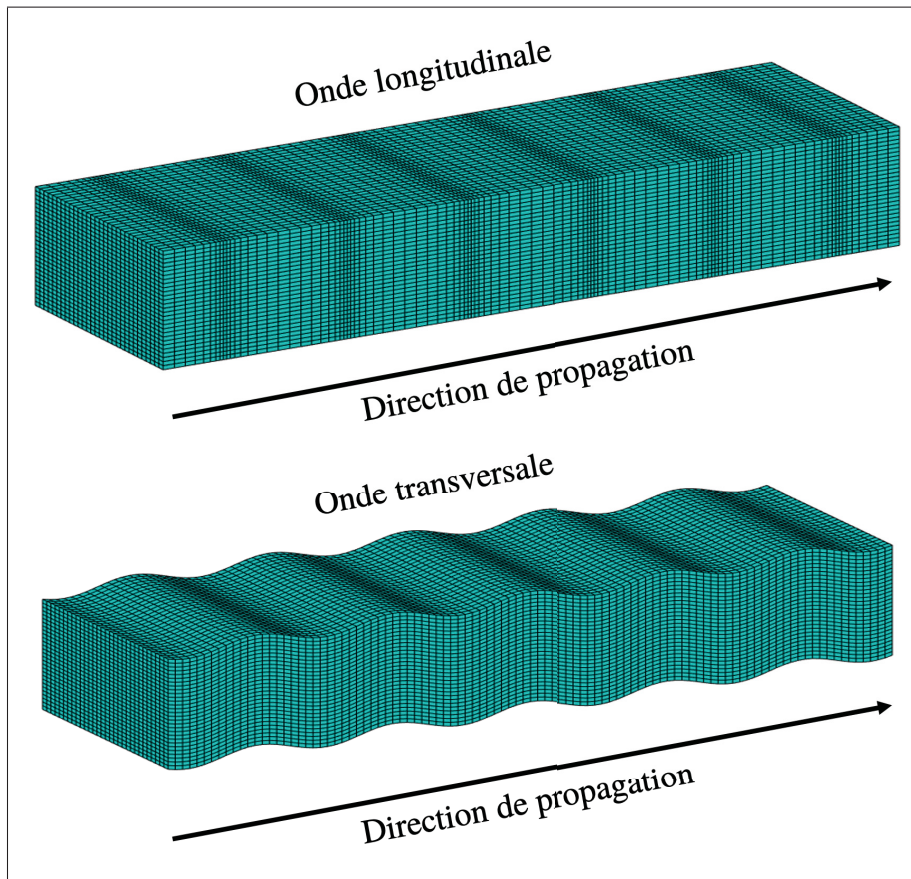


Figure 1.1 Les deux modes de propagation d'une onde acoustique de volume.

### 1.1.2 Comportement d'un ultrason aux interfaces

Quand une onde mécanique rencontre une frontière entre deux matériaux isotropes aux propriétés physiques différentes, deux phénomènes majeurs se produisent. Une partie de l'énergie est transmise d'un matériau à l'autre tandis que le reste de l'énergie est réfléchi. En fonction de l'angle d'incidence de l'onde principale et des propriétés physiques de chacun des deux matériaux, la loi de Snell-Descartes théorise le phénomène de propagation de l'onde entre les deux milieux. En considérant les exemples présentés Figure 1.11, la loi de Snell-Descartes peut s'exprimer ainsi :

$$\frac{\sin \theta_i}{c_i} = \frac{\sin \theta_{tL}}{c_{tL}} = \frac{\sin \theta_{tT}}{c_{tT}} = \frac{\sin \theta_{rL}}{c_{rL}} = \frac{\sin \theta_{rT}}{c_{rT}} . \quad (1.13)$$

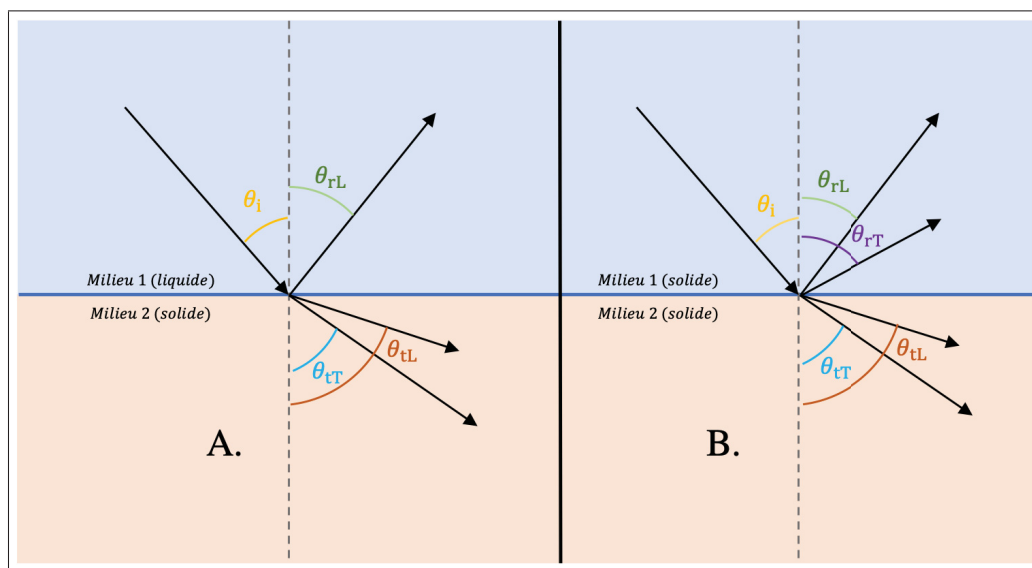


Figure 1.2 Schéma de propagation d'une onde aux interfaces. (A) la propagation d'un milieu liquide vers un milieu solide et (B) la propagation d'un milieu liquide vers un autre milieu solide.

$\theta_i$  est l'angle d'incidence de l'onde.  $\theta_{rL}$  est l'angle de l'onde longitudinale réfléchi.  $\theta_{rT}$  est l'angle de l'onde transversale réfléchi.  $\theta_{tL}$  est l'angle de l'onde longitudinale transmise.  $\theta_{tT}$  est l'angle de l'onde transversale transmise. La même notation est utilisée pour les vitesses  $c_i$ ,

$c_{tL}$ ,  $c_{tT}$ ,  $c_{tL}$  et  $c_{tT}$ . Quand le milieu incident est un milieu liquide non ou peu visqueux, les ondes transversales s'atténuent jusqu'à suivre un régime de dissipation pure, annulant ainsi leurs vitesses et, par voie de conséquence, leurs propagations. La loi de Snell-Descartes ne s'applique alors pas.

Pour chaque milieu  $n$ , peut être calculée une impédance acoustique  $Z_n$  selon l'équation 1.14, avec  $\rho_n$  sa densité et  $c_n$  la vitesse de propagation de l'onde dans ce milieu.

$$Z_n = \rho_n \cdot c_n . \quad (1.14)$$

Dans le cas d'une incidence normale, les fractions d'énergie transmise ( $T$ ) et réfléchie ( $R$ ) peuvent être calculées selon les équations suivantes :

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad T = \sqrt{1 - R^2} . \quad (1.15)$$

### 1.1.3 L'atténuation

L'atténuation d'une onde ultrasonore est le phénomène selon lequel l'énergie de l'onde diminue au fur et à mesure de sa propagation spatiale dans un matériau. Cela se manifeste par une diminution exponentielle de son amplitude ( $A(x)$ ) selon la formule suivante :

$$A(x) = A_0 \cdot e^{-\alpha x} \quad (1.16)$$

avec  $A_0$  l'amplitude initiale de l'onde,  $x$  la distance de propagation et  $\alpha$  le coefficient d'atténuation. Ce dernier dépend de nombreux phénomènes. Deux catégories d'atténuations sont proposées par Cheeke (2002) : une catégorie d'atténuation externe causée par les qualités microscopiques et macroscopiques des matériaux comme, par exemple, les fissures, les dislocations ou les joints de grains, et une catégorie d'atténuation intrinsèque, aussi appelée absorption, causée, par exemple,

par les effets thermiques. Cheeke (2002) montre également que l'atténuation est proportionnelle au carré de la fréquence de l'onde.

Lors d'une acquisition, l'atténuation, comme décrite ci-dessus, pourrait être compensée par un gain pur appliqué à la mesure du signal mais c'est sans compter sur le bruit. En effet, toute acquisition expérimentale est soumise au bruit, qu'il soit électronique ou dû aux imperfections du matériau ou aux mêmes sources que l'atténuation externe. Le signal brut est donc moins différentiable du bruit. On caractérise alors ce phénomène par le rapport signal sur bruit (SNR) exprimé en décibel (dB). La lisibilité du signal augmente avec le SNR. Afin de pouvoir augmenter l'amplitude du signal sans saturation de bruit, il est important de mettre en place des dispositifs de filtrage du bruit.

## **1.2 Modélisation par éléments finis des inspections ultrasonores**

Il existe aujourd'hui de nombreux moyens de simuler des inspections ultrasonores. L'un des moyens les plus communs est la simulation par éléments finis. Pour cela, il existe différents logiciels de modélisation comme, par exemple Abaqus, Pogo (Huthwaite, 2014) ou encore Ansys. Pour des raisons de modélisation de larges bases de données, nécessaires à cette étude, le logiciel Pogo a été choisi pour l'ensemble des modélisations. Son interface en ligne de commande associée à Matlab permet de modéliser efficacement des modèles différents par itérations sur plusieurs paramètres ; par exemple, pour modéliser 1000 fois le même modèle avec une dimension de défaut différente à chaque itération. De plus, Pogo est un logiciel de modélisation utilisant les processeurs graphiques. Cela permet de paralléliser des calculs davantage qu'avec un processeur central multi-cœur.

### **1.2.1 Équation élastodynamique discrète**

La modélisation par éléments finis utilise l'équation élastodynamique discrétisée dans le domaine spatial suivante :

$$\mathbf{M} \frac{d^2 \vec{u}}{dt^2} + \mathbf{C} \frac{d \vec{u}}{dt} + \mathbf{K} \vec{u} = \vec{F} \quad (1.17)$$

avec  $\vec{u}$  le vecteur déplacement.  $\mathbf{M}$  est la matrice diagonale de masse dont les valeurs dépendent de  $\rho$ .  $\mathbf{K}$  est la matrice de rigidité statique dont les valeurs dépendent de  $E$  et de  $\nu$ .  $\mathbf{C}$  correspond à la matrice d'amortissement visqueux dépendant directement de l'amortissement de Rayleigh. Enfin  $\vec{F}$  est la force extérieure appliquée. La propagation d'ondes est initialisée par une perturbation de l'état d'équilibre engendrée par l'application d'une force ou d'une contrainte de déplacement sur les noeuds du maillage.

La résolution de cette équation est effectuée par incrément temporel en calculant à chaque fois les valeurs des noeuds pour l'incrément suivant. Huthwaite (2014) détaille toutes les étapes de cette résolution. Plusieurs choix sont possibles parmi les types d'éléments et selon le nombre de dimensions du modèle. Ainsi, ils peuvent être tétraédriques ou cubiques pour un modèle 3D et triangulaires ou carrés pour un modèle bidimensionnel (2D). Pour les matériaux élastodynamiques, chaque noeud possède autant de degrés de liberté que le nombre de dimensions du modèle. Pour former un maillage, chaque noeud est attaché aux noeuds adjacents.

Même si l'objectif est de modéliser des inspections aussi proches du monde physique que possible, il n'est pas rare de préférer une modélisation 2D plutôt qu'une modélisation 3D. En effet, lorsqu'un modèle 3D, loin des effets de bords, n'est qu'une extrusion d'un modèle 2D, la différence de résultats est suffisamment minimale dans certains cas pour justifier un allègement drastique de la charge de calcul. Il s'agit de l'hypothèse retenue dans cette étude après avoir observé une différence minime entre les simulations 2D et 3D.

### 1.2.2 Frontières absorbantes

Par défaut, les limites du modèle sont des frontières libres, c'est à dire qu'elles sont comme connectées à un matériau à impédance acoustique très faible. Cela génère alors des interférences et des réflexions venant perturber l'acquisition. Prenons l'exemple d'une acquisition expérimentale :

les limites latérales de la pièce inspectée sont suffisamment éloignées de la sonde pour ne produire aucune réflexion significative. Cependant, afin d'alléger la modélisation, il est nécessaire de réaliser des modèles plus réduits. Afin d'absorber la totalité des ondes se propageant jusqu'aux frontières et de rendre ces modèles insensibles aux réflexions non désirées, des frontières absorbantes sont ajoutées au modèle. Les régions absorbantes modélisées dans cette étude sont divisées en multiples couches aux coefficients d'absorption croissant dans le sens de propagation de l'onde. La gradation du coefficient d'absorption doit être assez fine pour minimiser toute réflexion. De plus, l'épaisseur de chaque couche doit être suffisamment importante pour absorber correctement. L'étude de Rajagopal *et al.* (2012) recommande une épaisseur de 3 à 5 longueurs d'onde.

### 1.2.3 Taille des éléments et incrément temporel

Afin de réduire les erreurs de calculs lors de la modélisation, celles-ci doivent respecter 2 critères essentiels.

Premièrement, la taille des éléments doit être bien inférieure à la longueur d'onde. D'après Drozd *et al.* (2006) un minimum de 15 éléments par longueur d'onde est nécessaire. Sans cela, différents modes d'excitation de différentes longueurs d'onde pourraient être obtenus. Dans un modèle utilisant plusieurs matériaux, il est primordial de prendre pour étalon la longueur d'onde la plus petite. Dans cette étude, un rapport de 20 éléments par longueur d'onde dans l'eau a été utilisé.

En second lieu, l'incrément temporel de calcul  $\Delta t$  doit être inférieur au temps nécessaire pris par l'onde la plus rapide pour traverser un élément (Cook *et al.*, 2001). L'équation définie est la suivante :

$$\Delta t = 0,8 \frac{\Delta e_{\min}}{c_{\max}} \quad (1.18)$$

avec  $\Delta e_{\min}$  la taille de l'élément le plus petit  $e_{\min}$ ,  $c_{\max}$  la vitesse de propagation d'onde la plus élevée et 0,8 un facteur de sécurité.

### 1.3 Méthodes de contrôle non destructif par ultrasons

Le CND ultrasonore a bénéficié de nombreux développements au cours de ces dernières années pour des opérations d'inspection sur ligne de production. La méthode utilisée doit être adaptée à la fois à la précision d'inspection requise et au temps qui peut lui être alloué ; tout en sachant qu'une inspection dont résulte une plus grande précision d'inspection requiert plus de temps d'opération.

#### 1.3.1 Insonification par ondes planes

Pour générer une insonification par ondes planes, il est nécessaire d'utiliser une sonde multi-éléments. Il faut ensuite synchroniser l'émission de chaque élément selon une loi de retard précise, dépendant de l'angle d'inclinaison souhaité pour le front d'onde. Appliqué au CND d'une pièce immergée, la loi de retard doit prendre en compte la loi de Snell-Descartes (équation 1.13) afin que l'onde d'angle d'incidence  $\theta_i$  se réfracte selon l'angle de transmission longitudinale  $\theta_{tL}$ , dans le cas simple d'une surface plane. La Figure 1.3 schématise ce phénomène.

En considérant un capteur en immersion de  $N$  éléments d'abscisse  $x_e$  et un angle d'incidence dans l'eau  $\theta_i$ , le retard  $\Delta T_e$  d'émission de l'élément  $e$  est donné par :

$$\Delta T_e = \frac{x_e}{c_{L_e}} \sin \theta_i \quad (1.19)$$

avec  $c_{L_e}$  la vitesse de propagation de l'onde longitudinale dans l'eau. L'équation peut ensuite être exprimée en fonction de l'angle souhaité dans la pièce immergée  $\theta_{tL}$  par combinaison avec la loi de Snell-Descartes (équation 1.13).

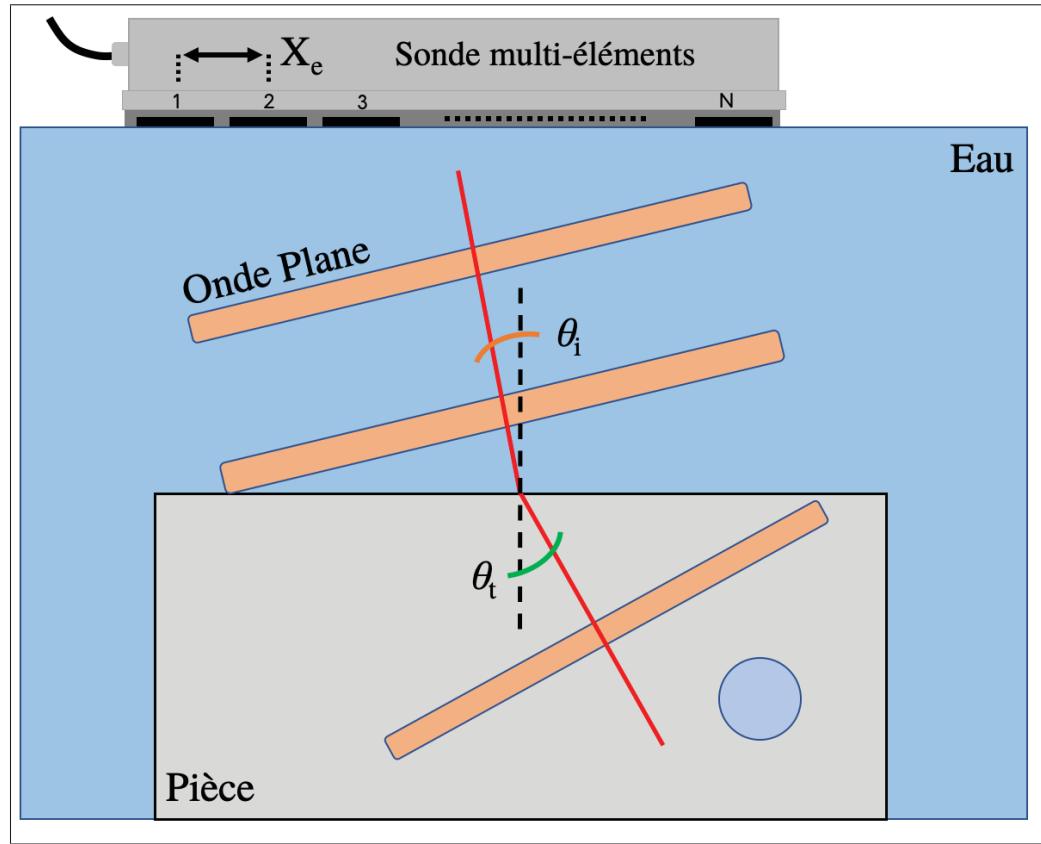


Figure 1.3 Émission et transmission d'une onde plane dans une pièce immergée.

$$\Delta T_e = \frac{x_e}{c_{L_p}} \sin \theta_{iL} \quad (1.20)$$

où  $c_{L_e}$  est la vitesse de propagation de l'onde longitudinale dans la pièce immergée. Ainsi, cela démontre -t-il que tout se déroule comme si l'émission était faite directement dans la pièce. Par simplification directe, si l'onde plane a un angle d'incidence nul, tous les éléments doivent émettre en même temps, c'est-à-dire avec un retard nul.

### 1.3.2 Techniques d'imagerie ultrasonore

#### 1.3.2.1 A-Scan

Un A-Scan est la représentation de l'amplitude ondulatoire d'un élément d'une sonde (selon l'axe des ordonnées) au fur et à mesure du temps (selon l'axe des abscisses) (Mountford & Wells, 1972). Les A-Scans sont les moyens les plus simples de représenter les données brutes de réception. Ils fournissent des informations sur l'amplitude des échos ainsi que sur la distance des défauts par rapport à la surface de la pièce. La Figure 1.4 illustre l'acquisition et la représentation d'un A-Scan.

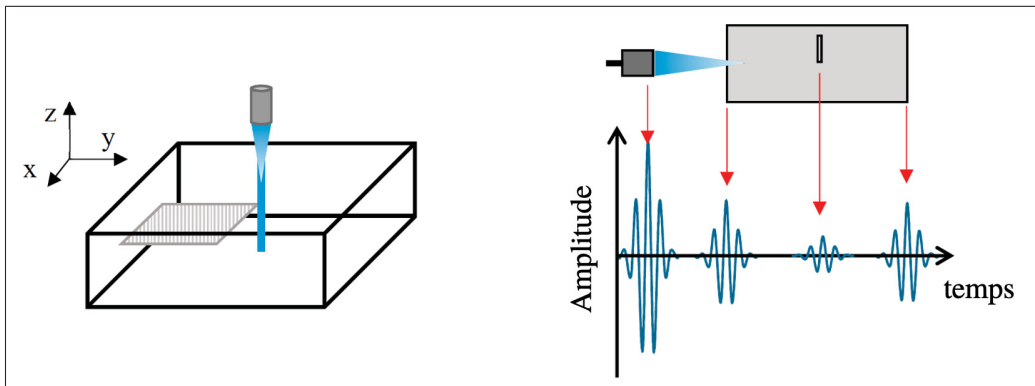


Figure 1.4 Schéma de l'acquisition d'un A-Scan (à gauche) et sa représentation (à droite).

#### 1.3.2.2 B-Scan

Les B-Scans sont les combinaisons des A-Scans selon la position des éléments de la sonde à la surface (Fatemi & Kak, 1980). L'amplitude de chaque A-Scan est généralement représentée suivant une échelle de couleur et occupe alors une colonne de pixels du B-Scan complet. Les B-Scans permettent de représenter la profondeur et la forme grossière des défauts d'une pièce. La Figure 1.5 représente l'acquisition d'un B-Scan selon la direction y ainsi que sa représentation.

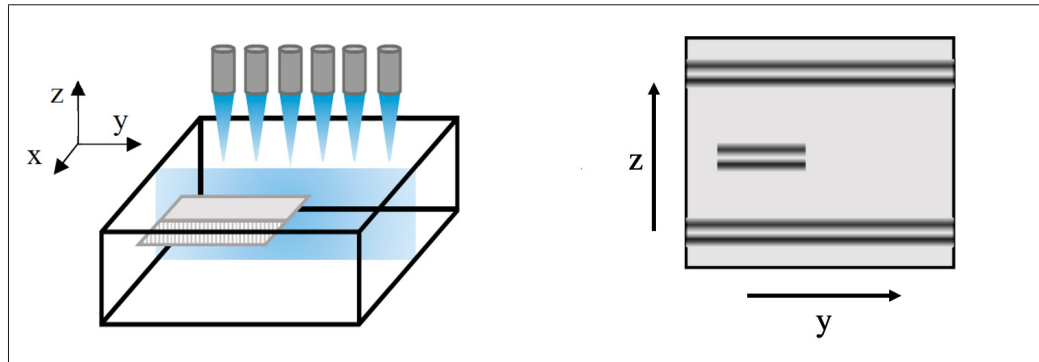


Figure 1.5 Schéma de l'acquisition d'un B-Scan (à gauche) et sa représentation (à droite).

### 1.3.2.3 Méthode de focalisation en tout point

Les techniques fondamentales détaillées précédemment ne permettent pas d'extraire de nombreuses données si elles sont utilisées comme telles. Pour obtenir plus d'informations comme la taille, la forme ou l'orientation d'un défaut, il est nécessaire de combiner une multitude de signaux et d'en effectuer une analyse mathématique appelée reconstruction.

La méthode d'imagerie la plus utilisée aujourd'hui est la focalisation en tout point (TFM, de l'anglais "*Total Focusing Method*"). Pour construire une TFM, il est généralement fait l'acquisition d'une FMC (de l'anglais "*Full Matrix Capture*") (Schmerr, 2015). La FMC requiert l'utilisation d'une sonde multiéléments, généralement un minimum de 16 éléments qui vont émettre successivement, alors que tous les éléments vont être utilisés pour recevoir les échos après chaque émission. Pour une sonde de  $n$  éléments, chaque émission donne alors une matrice de traces temporelles (TTM, de l'anglais "*Time Trace Matrix*") unique qui est la concaténation des traces temporelles d'amplitudes acquises dans l'ordre des éléments et dont la représentation est identique aux B-Scans. Chaque TTM est composée de  $n$  A-Scans et il est fait l'acquisition de  $n$  TTM. Une FMC est alors composée de  $n^2$  A-Scans. La Figure 1.6 illustre l'acquisition d'une FMC.

Une fois les données de FMC acquises, une zone de reconstruction incluse dans les TTM est choisie. Cette zone constitue une matrice des distances entre chacun de ses points et chaque

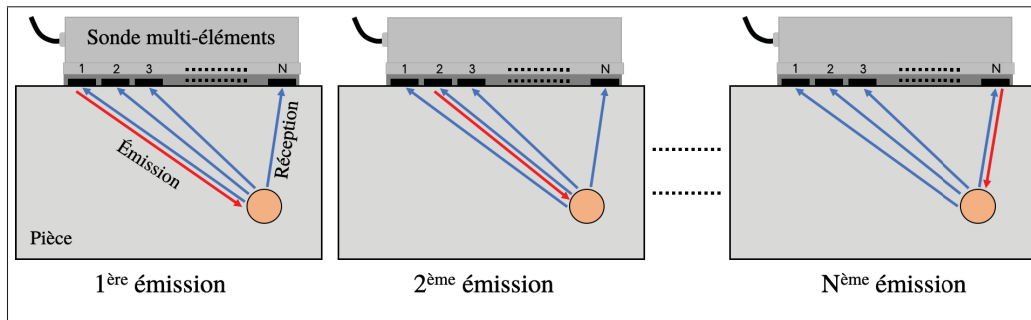


Figure 1.6 Schéma de l'acquisition d'une FMC.

élément. Connaissant la vitesse de l'onde dans le matériau de propagation, il est possible de sommer les amplitudes de chaque A-Scans dont les délais couvrent chaque point de la zone de reconstruction.

Cette sommation précise donne lieu à des interférences constructives et destructives qui aboutissent à une TFM et définissent ainsi les contours des défauts rencontrés. En effet, proches des défauts, les interférences sont majoritairement constructives ; les amplitudes crêtes de tous les A-Scans s'additionnent. Alors qu'éloignées des défauts, les interférences sont majoritairement destructives ; les amplitudes crêtes de tous les A-Scans s'additionnent pour donner un résultat plus faible.

Cette méthode n'est cependant pas parfaite. D'une part, le nombre d'émissions différées et le débit de données à acquérir et à traiter limitent la capacité d'affichage en temps réel de la reconstruction. D'autre part, en raison, notamment, de la présence d'artefacts de reconstruction ainsi que de la déformation des défauts, l'analyse humaine du résultat final nécessite une certaine expérience et une main d'oeuvre certifiées. Pallier ces défauts constitue aujourd'hui le grand enjeu de la TFM mais ils constituent encore un frein à l'utilisation de cette méthode sur des chaînes de production à haute cadence.

### 1.3.3 Chute à moins six décibels

Dans le domaine des ultrasons, la chute à  $-6$  dB est largement utilisée pour mesurer la longueur ou la largeur d'un réflecteur.

La Figure 1.7 illustre les différentes étapes de cette méthode. La sonde est balayée de gauche à droite. Quand l'émission est parfaitement au-dessus du défaut, l'amplitude de l'écho est à son maximum (Figure 1.7 B). Quand l'émission se réalise au bord du défaut (deux positions possibles, chacune de part et d'autre du défaut) la sonde émet directement vers le réflecteur et directement vers le fond du bac d'immersion. L'amplitude de l'écho est alors réduite de moitié par rapport au cas précédent ; ce qui correspond à une chute de  $-6$  dB (Figure 1.7 A et C). La distance entre les deux positions de chute à  $-6$  dB équivaut à une estimation de la largeur du défaut ( $x_2 - x_1$ ). Plus le défaut possède une surface plane, plus l'estimation est précise.

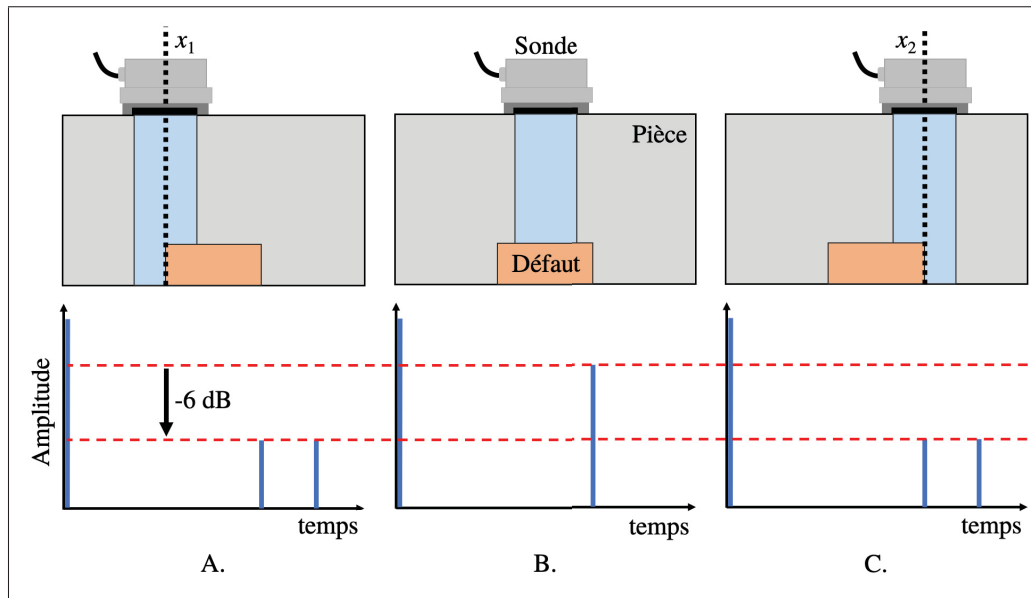


Figure 1.7 Schéma de la méthode de la chute à  $-6$  dB.

Cette méthode simple et rapide à mettre en place sera utilisée comme point de comparaison dans cette étude.

## 1.4 Les réseaux de neurones

Le concept de réseau de neurones est né il y a presque 70 ans, mais, trop gourmand en calculs pour les capacités informatiques de l'époque, il n'a pas connu de succès particulier de développement. Aujourd'hui, après les progrès fulgurants en électronique pour développer des processeurs de plus en plus performants, les réseaux de neurones sont autant déployés dans le domaine grand public que dans le milieu professionnel. La recherche dans cette spécialité est très active, en développant de nouveaux algorithmes ou en procédant à l'optimisation de plus anciens ; le champ d'applications augmente et les limites opérationnelles sont sans cesse repoussées.

### 1.4.1 Le modèle du perceptron

Le perceptron (1.8) qui est l'élément de base du réseau de neurones, a été conceptualisé dans une étude de Rosenblatt (1958). Il s'agit d'un opérateur mathématique à plusieurs entrées  $x_k$  mais sortie unique  $s$ . Le calcul de la sortie découle de la formule suivante :

$$s = \sigma \left( \sum_{k=1}^n w_k x_k + b \right) . \quad (1.21)$$

Les réels  $w_k$  sont appelés les poids. Ils donnent plus ou moins d'importance à leur entrée respective. Le réel  $b$  est appelé le biais.  $n$  est le nombre total d'entrées (équivalent au nombre total de poids).  $\sigma$  est la fonction d'activation. La fonction d'activation du premier modèle de perceptron de Rosenblatt (1958) est la fonction d'Heaviside. Le calcul de la sortie suit donc la formule suivante :

$$s = \begin{cases} 1 & \text{si } \sum_{k=1}^n w_k x_k + b > 0 , \\ 0 & \text{autrement.} \end{cases} \quad (1.22)$$

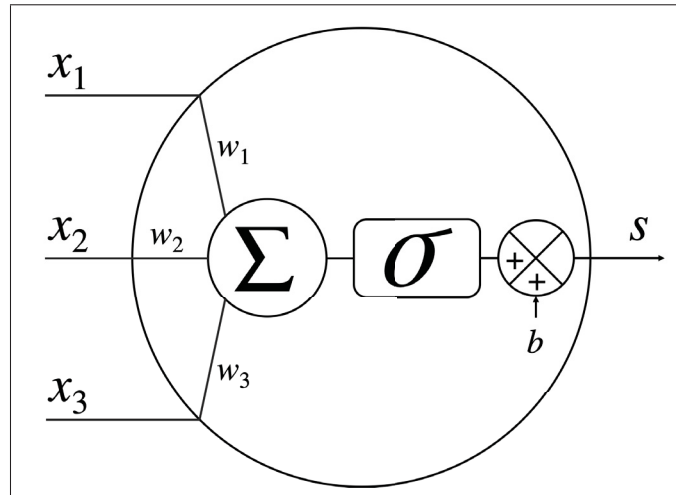


Figure 1.8 Schéma du concept de perceptron à trois entrées.

Les perceptrons se sont aujourd'hui complexifiées avec la diversification de la fonction d'activation. Par exemple, la fonction d'activation la plus utilisée aujourd'hui est la fonction ReLU  $\sigma_{\text{ReLU}}$  (de l'anglais "*Rectified Linear Unit*"). Avec  $z$  un réel, la formule est la suivante :

$$\sigma_{\text{ReLU}}(z) = \begin{cases} z & \text{si } z > 0, \\ 0 & \text{autrement.} \end{cases} \quad (1.23)$$

Comme son nom l'indique, les réseaux de neurones sont constitués de plusieurs neurones et par analogie, de plusieurs perceptrons interconnectés. La frontière définissant l'état de la sortie devient donc plus complexe et il peut y avoir plusieurs sorties au réseau de neurones. La Figure 1.9 illustre un réseau de neurones multicouches entièrement connectés. Le nombre d'interconnexions ainsi que le nombre de couches alourdissent directement l'apprentissage, nécessitant ainsi de plus grandes capacités de calculs. Il est donc rare d'utiliser des réseaux de neurones entièrement connectés comme dans la Figure 1.9.

Les poids sont initialisés de façon aléatoire. La mise à jour itérative des poids  $w$  du réseau de neurones constitue ce que l'on nomme l'apprentissage. L'objectif est de diminuer l'écart entre la

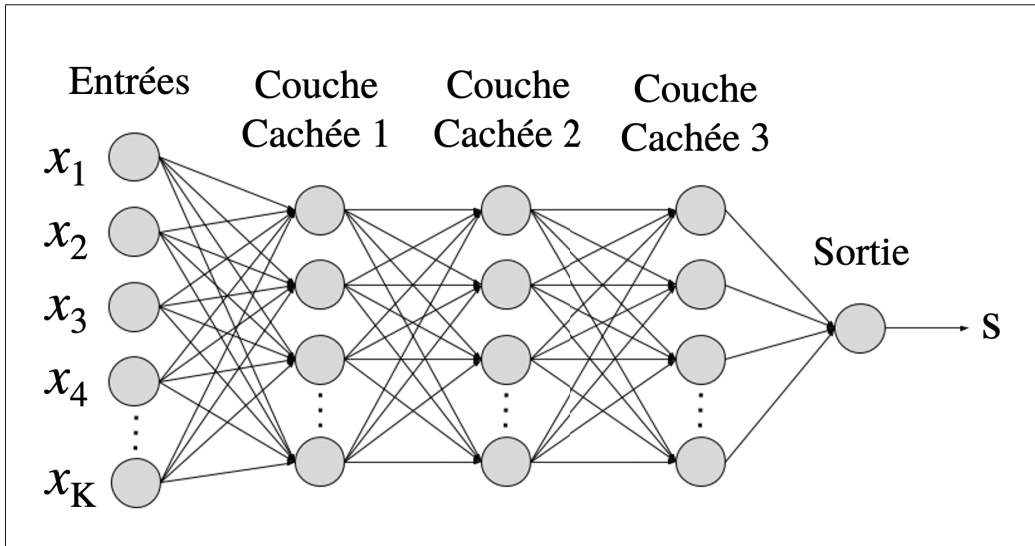


Figure 1.9 Réseau de neurones entièrement connectés.

sortie  $s$  et la sortie désirée  $r$  pour chaque perceptron. Ce calcul de l'écart est appelé la fonction de coût  $C$ . Un exemple de fonction de coût est le calcul de l'erreur quadratique moyenne. Pour un entraînement avec  $N$  exemples, en notant  $s_i$  les sorties successives de l'entraînement et  $r_i$  les sorties successives désirées, la fonction de coût est la suivante :

$$C = \frac{1}{N} \sum_{i=1}^N (s_i(w) - r_i)^2. \quad (1.24)$$

L'objectif est de diminuer la fonction de coût au fur et à mesure des incréments de calculs  $\partial t$  et de trouver un minimum en résolvant :

$$\frac{\partial C}{\partial t} = \frac{\partial C}{\partial w} \frac{\partial w}{\partial t} \leq 0. \quad (1.25)$$

La minimisation de la fonction de coût à chaque itération est appelée la descente du gradient. Par rétropropagation, les poids sont mis successivement à jour dans les couches de neurones. La méthode de la descente du gradient est complexe en  $N$ . Avec l'arrivée de larges bases de données d'entraînement, une nouvelle méthode, le gradient stochastique, beaucoup plus économique en

calculs, a été développée et son utilisation est aujourd'hui largement répandue. Cette méthode est plus rapide car la mise à jour des poids ne se fait plus après l'évaluation de l'ensemble de la base de données d'entraînement mais après chaque itération. Le défaut de cette méthode est de limiter la probabilité de trouver un minimum global au cas d'étude.

#### 1.4.2 Les réseaux de neurones convolutifs

Les premières architectures de réseaux de neurones convolutifs (CNN de l'anglais "*Convolutional Neural Network*") apparurent à partir des années 1990 (Fukushima, 1980; LeCun *et al.*, 1999). L'utilisation de tels réseaux de neurones s'est popularisée récemment après l'apparition d'une nouvelle génération de carte graphique en 2014. Ils peuvent être capables, par exemple, de nommer les différents éléments présents dans une image (la classification ou l'étiquetage) ou même capables de localiser avec précision l'emplacement de ces éléments dans l'image (1.10).

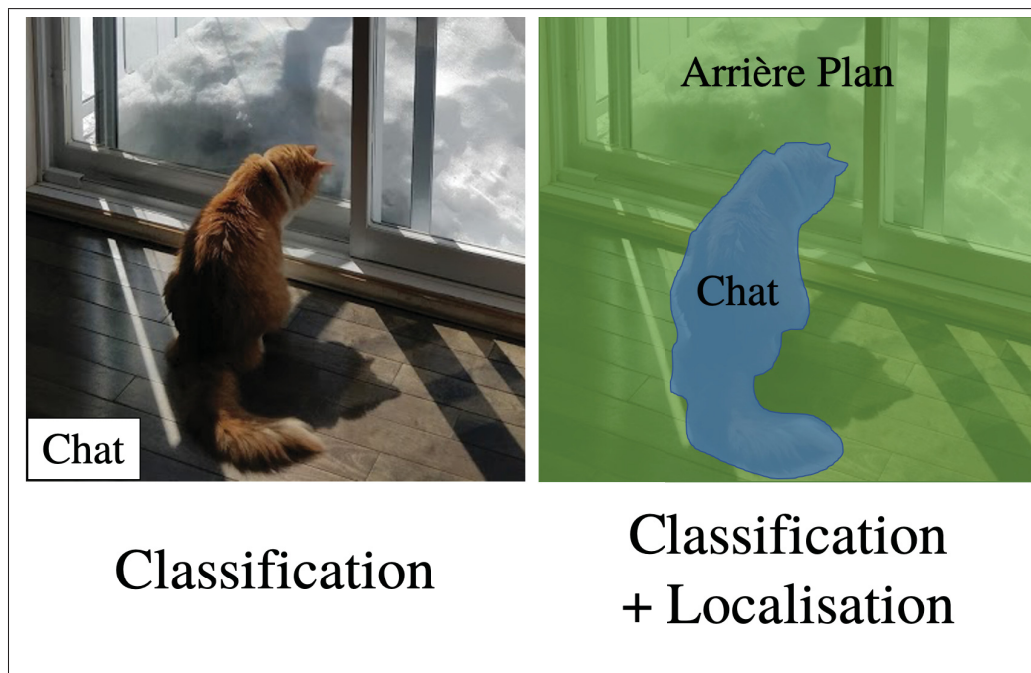


Figure 1.10 Différentes possibilités de prédictions d'un CNN.

### 1.4.2.1 Principe de base

Les CNN ont été développés afin d'éviter de calculer tous les poids d'un réseau de neurones. En effet, ce calcul n'apparaît pas ou si peu nécessaire, et, par conséquent, n'est pas véritablement efficace. Préférons davantage la prise en compte de quelques poids dupliqués à l'ensemble de la couche neuronale. Cette méthode tente d'imiter le fonctionnement du cortex visuel des animaux, c'est pourquoi les CNN sont largement utilisés dans le domaine de la vision par ordinateur.

Ce traitement se rapproche de la méthode des filtres de Sobel (Kanopoulos *et al.*, 1988) utilisée en imagerie pour la détection des contours. Les algorithmes de CNN sont plus sophistiqués car ils mettent à jour les poids de pondération des filtres à chaque itération de l'entraînement.

En prenant comme exemple de donnée d'entrée une image, cette dernière peut être considérée comme une matrice de pixels. La Figure 1.11 simplifie chaque pixel en une valeur binaire. En considérant une matrice de poids, appelée "noyau" (de l'anglais "*kernel*"), l'opération de convolution consiste à faire glisser cette matrice de poids au dessus de l'image originale et de réaliser un produit matriciel de Hadamard pour chaque position désirée. Afin de conserver la dimension de la matrice d'origine après la convolution, des pixels nuls sont ajoutés en bordure et le produit de Hadamard est réalisé pour chaque position possible non centrée sur un pixel nul. Le "pas" (de l'anglais "*stride*") est alors de  $(1 \times 1)$ . Le résultat est appelé "carte de redescription" (de l'anglais "*feature map*"). S'en suit alors l'application de la fonction d'activation.

En pratique, un CNN "apprend" les valeurs de ces poids au fur et à mesure de l'entraînement, afin d'arriver au résultat souhaité. Il faut également définir le nombre de noyaux (c'est à dire le nombre de matrices de poids qui vont réaliser cette opération). Chaque noyau sera unique et pour chaque noyau sera calculée une carte de redescription. La matrice en 3D qui est le regroupement de ces noyaux 2D est appelé "filtre"

Entre chaque regroupement d'opérations de convolution, se produit une opération de sous-échantillonnage (aussi appelée en anglais "*pooling operation*"). Il peut y avoir différents types d'opération de ce genre comme le maximum, le minimum, la somme, la moyenne, ... Il s'agit de

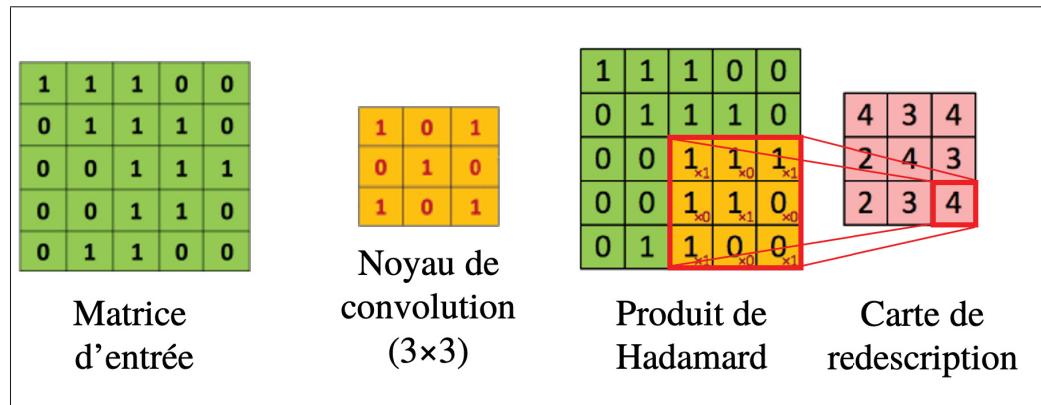


Figure 1.11 Schéma d'une opération de convolution.

faire glisser un noyau de dimension et de pas fixe sur la carte de redescription puis d'appliquer l'opération voulue. La Figure 1.12 illustre l'extraction du maximum (*max pooling*). Selon le pas choisi, la matrice sera plus ou moins sous-échantillonnée.

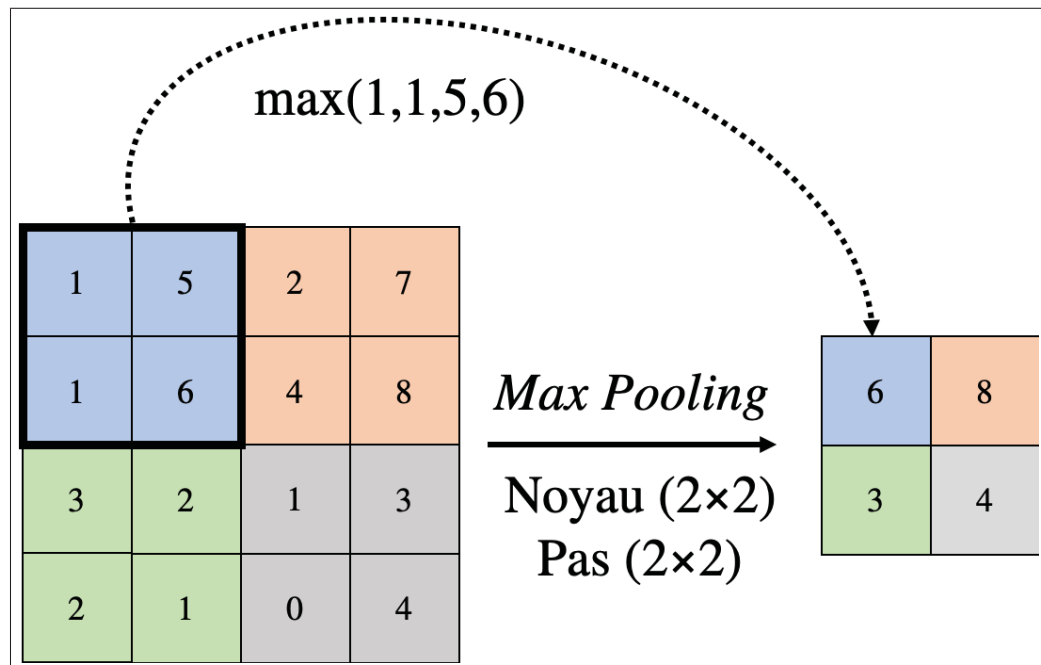


Figure 1.12 Schéma d'une opération de sous-échantillonnage.

Les intérêts du sous-échantillonnage sont nombreux. Les données d'entrées étant de plus en plus légères, la charge de calculs diminue. Cela rend les données plus maniables mais également

moins propices au surapprentissage (détailé par la suite). Le réseau de neurones est également moins sensible aux petites variations, permettant alors d'obtenir le même résultat pour deux données similaires mais avec de légères différences insignifiantes. Enfin, cela permet par exemple de détecter un objet - peu importe où cet objet est localisé car la représentation obtenue est équivariante.

Pour finir, les dernières couches du CNN sont généralement des couches de neurones entièrement connectées dont l'objectif est de transformer les matrices que sont les cartes de redescription en vecteurs de classification. Par exemple, si l'objectif est de classer une image d'oiseau en lui attribuant un label parmi 4 disponibles (*chien, chat, oiseau, ours*), le résultat doit être un vecteur "proche" du vecteur (0, 0, 1, 0) pour la classer comme une image représentant un oiseau. Pour cela, la dernière couche utilise la fonction d'activation *softmax* ( $\sigma_S$ ) par exemple. Elle permet de convertir un vecteur de valeurs réelles  $y_i$  en vecteur de probabilités  $p_i$  selon la formule suivante :

$$\sigma_S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} = p_i . \quad (1.26)$$

Les architectures de CNN sont classées selon le nombre de couches de convolution et le nombre de couches entièrement connectées. Par exemple, le modèle de CNN utilisé comme base dans cette étude est le VGG16 (Simonyan & Zisserman, 2014). Il s'agit d'un modèle assez léger car il ne compte que 13 couches de convolution et 3 couches entièrement connectées, réparties selon la Figure 1.13. L'étude de Simonyan & Zisserman (2014) aboutit à 92,7 % de réussite lors du test sur la base de données ImageNet (Deng *et al.*, 2009). Cette base de données est aujourd'hui composée de 14 millions d'images, réparties en plus de 21000 classes.

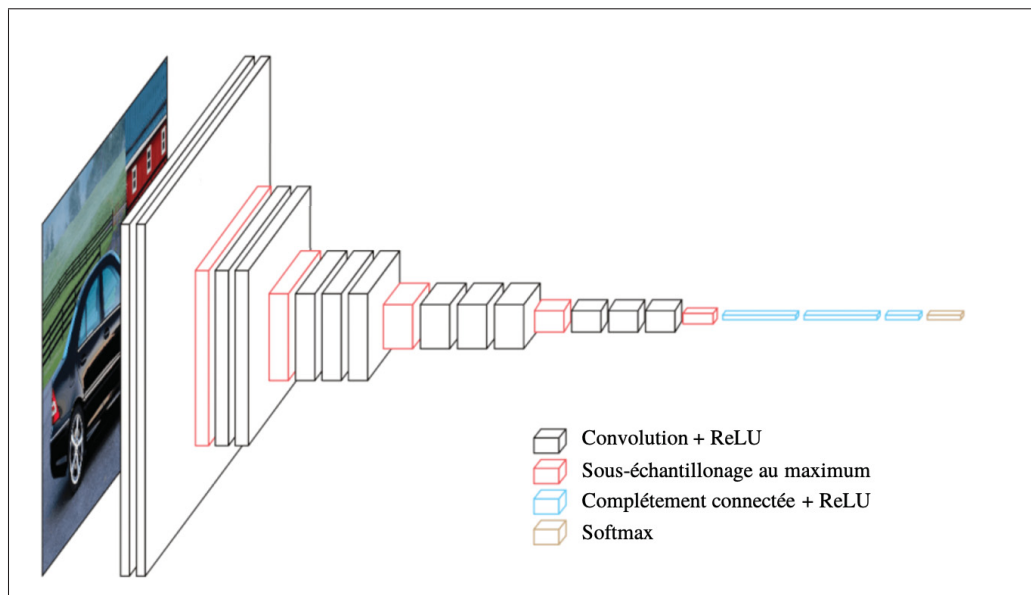


Figure 1.13 Schéma d'architecture d'un VGG16.

#### 1.4.2.2 Les différents types de CNN

Différentes méthodes d'apprentissages ont été développées afin de s'adapter à la base de données d'entraînement disponible. L'apprentissage non-supervisé a lieu lorsque les données d'entraînement ne sont pas du tout étiquetées. Le réseau de neurones doit alors automatiquement identifier les similitudes d'une même classe.

L'apprentissage est dit supervisé lorsque l'intégralité des données d'entraînement sont étiquetées. Étiqueter de larges bases de données étant fréquemment une opération manuelle, celle-ci est longue et coûteuse. L'apprentissage semi-supervisé (Weston *et al.*, 2008) utilise, pendant l'entraînement, un faible pourcentage de données étiquetées (entre 0,1 % et 0,5 %). Il est aussi possible d'effectuer un apprentissage dit "renforcé" (Weston *et al.*, 2008) en indiquant uniquement les classes présentes sur l'image mais sans les localiser. Le réseau de neurones devra tout de même parvenir à les localiser après l'entraînement. Il s'agit d'un apprentissage non-supervisé (Durand, 2017) lorsqu'aucune classe ni localisation ne sont données. Le réseau de neurones devra automatiquement rassembler les points d'une classe ensemble grâce à des techniques de regroupement.

Dans le cas du CND où les inspections peuvent être simulées assez fidèlement, il est possible d'envisager d'utiliser un apprentissage supervisé. En effet, une large base de données d'inspection classifiées n'existe pas mais les simulations numériques peuvent être automatiquement classifiées au moment de la modélisation car elles sont entièrement paramétrées.

Attention cependant ! L'apprentissage d'un réseau de neurones peut-être trop fin. Un tel cas et le réseau de neurones est alors incapable de transférer le domaine de simulation au domaine expérimental. Le transfert d'apprentissage et de domaines est un enjeu majeur de l'apprentissage machine. Il est intéressant dans le domaine du CND par ultrasons car les mesures expérimentales sont plus coûteuses à réaliser et plus fastidieuses à étiqueter que les simulations. Pour réaliser ce transfert, il est possible de réaliser un double apprentissage. Dans cette étude, le premier apprentissage est réalisé grâce à la large base de données simulées (base source). Il consiste surtout à apprendre au réseau de neurones la détection des contours. Ensuite, le second apprentissage est réalisé à partir de la base de données expérimentales (base cible) pour ajuster le réseau de neurones à ce domaine. La taille de la base cible ne peut comparativement représenter que quelque pour-cent de la taille de la base source (Shin *et al.*, 2016; Barbedo, 2018). Bien qu'avec cette méthode, le nombre d'échantillons dans la base de données expérimentales ne vaut que 4 % du nombre d'échantillons dans la base de données simulées, il peut être initialement beaucoup plus faible.

En effet, il est possible de réaliser une augmentation de données. Il s'agit de déformer, modifier, transformer les données originales pour en obtenir de nouvelles. Il faut cependant faire attention à la cohérence et à la pertinence des données additionnelles. Dans le domaine du CND, il est possible, à partir d'un B-Scan par exemple, de réaliser des symétries axiales ou centrales, de recadrer ou encore d'illuminer un défaut pour en augmenter la taille. Cette dernière technique a été utilisée par Virkkunen *et al.* (2019).

### 1.4.2.3 Le Faster R-CNN

Le Faster R-CNN est un réseau de neurones bien connu pour la détection d'objets dans des images. A partir d'une image d'entrée, son objectif est de fournir un encadrement des objets appartenant à l'une des classes prédéfinies et d'identifier cette classe. La Figure 1.14 illustre un exemple de prédiction désirée avec une image naturelle.

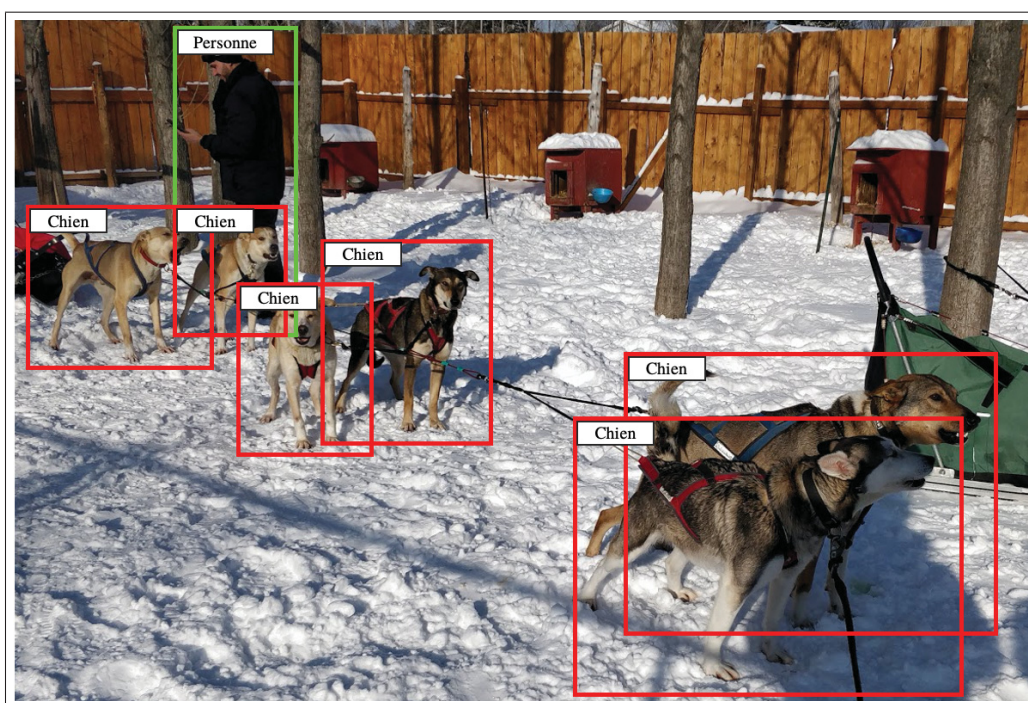


Figure 1.14 Exemple de prédiction issue d'un Faster R-CNN.

Pour atteindre cet objectif, il est nécessaire de fournir en entraînement des images accompagnées des coordonnées de chacun des encadrements des objets à classer ainsi que leur classe correspondante.

Le Faster R-CNN est un regroupement de trois réseaux de neurones. Le premier est un réseau de neurones VGG (détaillé précédemment) dénué de ses dernières couches entièrement connectées. Le deuxième est un réseau de propositions de régions (RPN, de l'anglais "*Region Proposal Network*"). Le troisième est un réseau de détection. La Figure 1.15 illustre la combinaison de ces 3 réseaux.

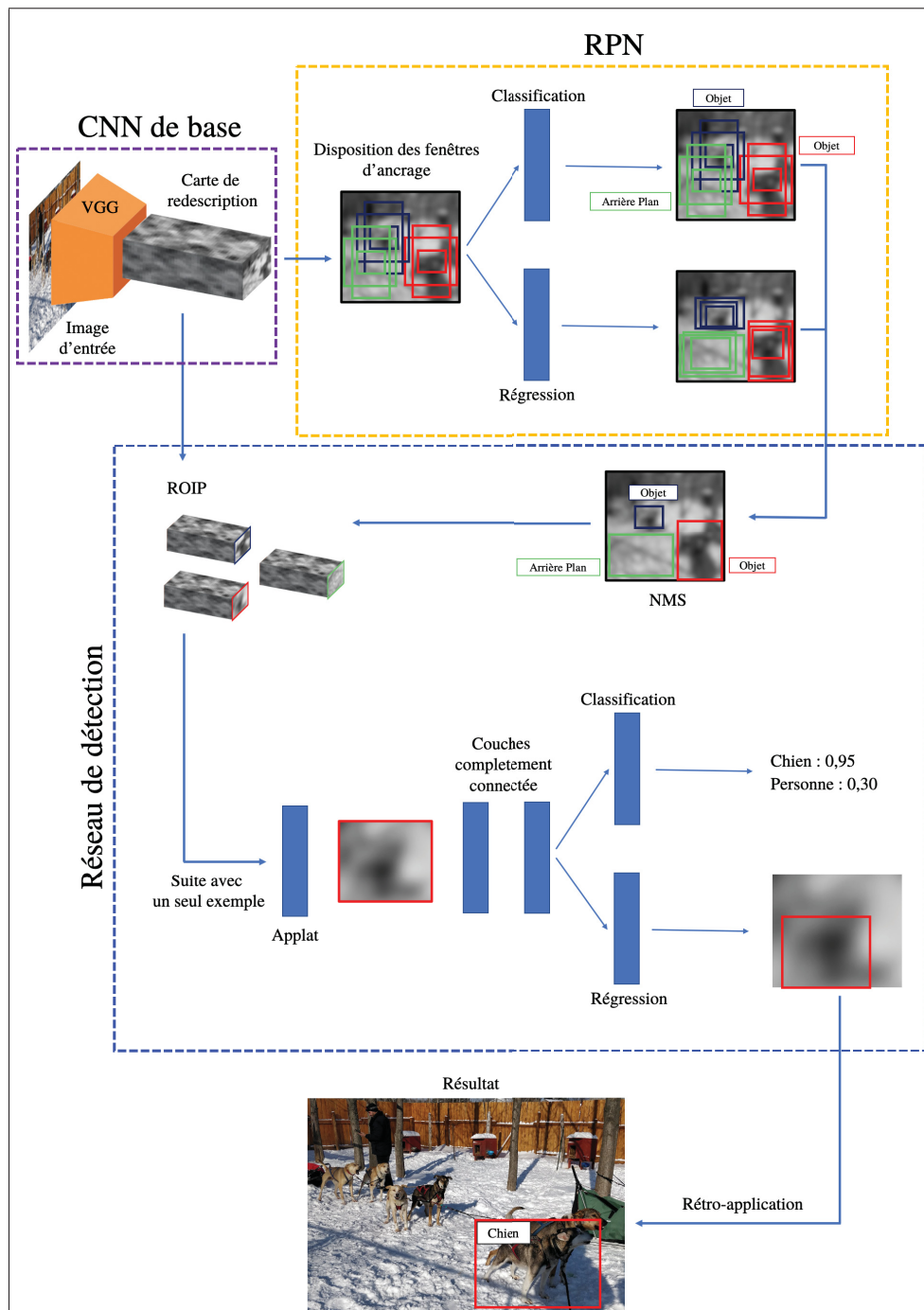


Figure 1.15 Schéma des étapes de fonctionnement d'un Faster R-CNN par le regroupement d'un CNN, d'un RPN et d'un réseau de détection.

#### 1.4.2.4 RPN

Pour chaque "pixel" de la carte de redescription, le RPN doit apprendre s'il y a un objet aux alentours de ce point puis en estimer la position et la taille. Pour cela, à chaque pixel (appelé aussi "ancree") est associé un nombre  $k$  prédéfini de fenêtres d'ancrage. Les fenêtres d'ancrage sont des fenêtres rectangulaires de différentes tailles et de différents rapports de longueur sur largeur.

Tout d'abord, la carte de redescription résultante du CNN subit une nouvelle couche de convolution qui en conserve les dimensions de hauteur et de largeur.

Après cela, la carte de redescription subit parallèlement 2 couches de convolution de noyaux ( $1 \times 1$ ). L'une des couches de convolution est la couche de classification. Le pas de convolution est de 1, le nombre de noyaux est de  $2k$  et les valeurs de sorties sont des probabilités (aussi appelées scores) de 0 à 1. Pour chaque fenêtre d'ancrage, il y a donc le score de probabilité que celle-ci encadre une zone avec présence d'un objet et le score de probabilité que celle-ci encadre une zone en dehors de toute présence d'objet. Autrement dit, cette couche permet d'apprendre pour chaque ancre si cette dernière "possède" ou pas un objet parmi chacune de ses fenêtres d'ancrage. L'autre couche est la couche de régression. Le pas de convolution est de 1, le nombre de noyaux est de  $4k$  et les valeurs de sorties sont des coefficients de régression. Il s'agit des 4 coefficients  $\Delta x$ ,  $\Delta y$ ,  $\Delta w$ ,  $\Delta h$  pour corriger les coordonnées du point d'ancrage ( $\Delta x$ ,  $\Delta y$ ) et les dimensions de la fenêtre d'ancrage ( $\Delta w$ ,  $\Delta h$ ). Cette couche permet d'apprendre à améliorer la localisation et la taille de chaque fenêtre d'ancrage de chaque ancre. Ces deux sorties constituent conjointement le groupe de propositions des fenêtres d'ancrage.

Dans ce groupe, de nombreuses fenêtres se superposent. La mesure de la superposition s'effectue par le calcul du rapport de l'intersection sur l'union (IoU, de l'anglais "*Intersection over Union*") d'une paire de fenêtres. Une simple couche algorithmique appelée "suppression du non-maximum" (NMS, de l'anglais "*Non-Maximum Suppression*") permet d'en retirer une large majorité redondante. Cet algorithme retire du groupe de propositions les fenêtres d'ancrage qui

se superposent au-delà d'un certain seuil d'IoU, avec d'autres fenêtres qui possèdent un score de classification plus élevé.

Lors de l'entraînement, le RPN doit apprendre à donner de meilleurs groupes de propositions. En ce qui concerne la couche de classification, afin de calculer les valeurs de sorties  $s$  à atteindre pour effectuer le calcul de la fonction de perte, les coordonnées des fenêtres d'ancrage sont comparées aux vraies coordonnées des objets dans l'image. Les 128 meilleures superpositions sont identifiées comme étant des régions d'intérêt d'objet. On a alors  $s = 1$ . Les 128 pires superpositions sont identifiées comme étant des régions d'intérêt d'arrière-plan. On obtient alors  $s = 0$ . Ces 256 régions proposées constituent le groupe final de propositions. Enfin, les autres propositions "inutiles" impliquent  $s = -1$ . Les coefficients de régression de sorties sont calculés de sorte que la fenêtre d'ancrage se superpose au véritable encadrement de l'objet le plus proche. Ce n'est qu'après le calcul de la fonction de perte pour chacun des réseaux de classification et de régression, que la rétropropagation s'effectue.

Lors du test du réseau de neurones, le groupe de propositions ne passe pas cette dernière étape mais est envoyé directement au réseau de détection.

#### 1.4.2.5 Réseau de détection

La première étape du réseau de détection est le redimensionnement des régions proposées (ROIP, de l'anglais *"Region Of Interest Pooling"*). En effet, les régions proposées fournies par le RPN sont de différentes tailles. Le ROIP ne classe pas par la suite que des ROI de tailles identiques, ce qui est beaucoup plus efficace. Le ROIP prend comme entrée la carte de redescription résultant du CNN ainsi que les coordonnées des régions d'intérêt du RPN. La Figure 1.16 détaille les étapes du ROIP sur un exemple simple. La sortie du ROIP est une "petite" carte de redescription (aussi appelée "proposition") de taille fixe pour chaque région proposée par le RPN.

Chacune de ces propositions est "aplatie" par une couche neuronale. Il s'agit de transformer les matrices tridimensionnelles que sont les petites cartes de redescription en simples vecteurs colonnes. Ensuite, chaque vecteur passe par deux couches entièrement connectées (dont la



Figure 1.16 Exemple de ROIP sur deux fenêtres d'ancrage différentes.

fonction d'activation habituellement utilisée est la fonction ReLu) avant de passer par deux couches parallèles : une couche de classification et une couche de régression. Ces couches sont différentes de celles employées pour le RPN. Ce sont des couches entièrement connectées et non pas des couches de convolution. La taille de la couche de classification est égale au nombre de classes identifiables  $C$  additionné à la classe d'arrière-plan. Elle fonctionne comme la classification d'un CNN précédemment évoqué. La couche de régression est de taille  $4C$  (les 4 coefficients de régression  $\Delta x$ ,  $\Delta y$ ,  $\Delta w$ ,  $\Delta h$ , pour chacune des classes). La fonction d'activation utilisée pour la couche de classification est la fonction *softmax* ; pour la couche de régression, il s'agit de la fonction de regression linéaire.

Pour l'entraînement, la sortie de la couche de classification est calculée afin d'être identique à la classification du véritable encadrement dont l'IoU avec la proposition est supérieure à 0.5 (valeur usuelle). La sortie de la couche de régression est calculée comme étant le décalage entre la proposition et le véritable encadrement précédemment cité.

Pour le test, la classification prise en compte est celle de plus haute probabilité et la régression prise en compte est celle correspondant à cette classification. Enfin, une nouvelle fois, une NMS est appliquée afin d'espérer ne conserver qu'une seule fenêtre d'ancrage par objet.

#### **1.4.2.6 Entraînement, validation et test**

L'entraînement d'un réseau de neurones est réalisé à partir d'une base de données aussi large que possible. La phase d'entraînement complète est divisée en périodes. Ici, la descente du gradient stochastique est utilisée, donc chaque période est un entraînement sur la base de données au complet. Au fur et à mesure des différentes périodes, les échantillons sont donc utilisés plusieurs fois. L'évolution de la fonction de perte n'est donc pas complètement "objective" car le réseau de neurones "connaît" déjà les échantillons. Afin d'obtenir une mesure objective de cette fonction de perte, une même base de données indépendante de validation est utilisée entre chaque période pour calculer "objectivement" cette fonction. Les poids ne sont pas mis à jour pendant les phases de validation. Le test s'effectue alors à partir d'une troisième base de données dont les échantillons n'ont jamais été utilisés lors de l'entraînement.

### **1.5 Bilan**

Ce premier chapitre a présenté un tour d'horizon des fondamentaux des ondes ultrasonores, du contrôle non destructif par ultrason et de sa modélisation par éléments finis ainsi que les bases des réseaux de neurones convolutifs, en particulier l'architecture du Faster R-CNN.

En résumé, en émettant une unique insonification par onde plane, il est possible de réaliser des inspections adaptées aux hautes cadences industrielles. Grâce aux logiciels Pogo et Matlab, il est possible d'exploiter la modélisation par éléments finis pour générer itérativement un grand nombre de blocs, avec et sans défaut, puis de simuler une insonification par onde plane. Cette base de données étant automatiquement classifiée lors de la modélisation, il est possible de mettre en place un réseau de neurones par apprentissage supervisé. L'architecture Faster R-CNN semble particulièrement adaptée pour localiser et dimensionner des FBH et des SDH. Ce réseau

de neurones peut d'abord être entraîné avec la base de données simulées. Ensuite, une petite série de mesures expérimentales peut être augmentée afin de constituer une base de données supplémentaire suffisante pour continuer l'entraînement du réseau de neurones en utilisant le principe de transfert d'apprentissage.



## CHAPITRE 2

### **TOWARDS USING CONVOLUTIONAL NEURAL NETWORK TO LOCATE, IDENTIFY AND SIZE DEFECTS IN PHASED ARRAY ULTRASONIC TESTING**

Thibault Latête, Pierre Bélanger

Département de Génie Mécanique, École de Technologie Supérieure,  
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article soumis à la revue « Ultrasonics » en juillet 2020.

#### **2.1 Abstract**

Machine learning algorithms are widely used in image recognition. In phased array ultrasonic testing, images are typically formed through constructive and destructive superpositions of signals backscattered from flaws or geometrical features. However, images generated in phased array ultrasonic testing remain difficult to interpret. In this study, the Faster R-CNN was used to identify, locate and size flat bottom holes (FBH) and side-drilled holes (SDH) in an immersed test specimen using a single plane wave insonification. The training was performed on segmented and classified data generated using GPU-accelerated finite element simulations. SDH and FBH of different diameters, depths and lateral positions were included in the training set. The thickness of the test specimen was also variable. An ultrasonic phased array probe of 64 elements was simulated. All elements of the phased array probe were fired at the same time and the time traces from each element were recorded. The individual time traces were concatenated to form a matrix, which was then used in the training. This inspection scenario enables fast acquisition of data at the expense of poor lateral resolution in the resulting image. The trained neural network was initially tested using finite element simulations. Results were assessed in terms of the intersection of the union (IoU) between the ground truth geometry and the predicted geometry. With the simulated cases, the thickness of the test specimen was detected in all cases. When using a 40% IoU threshold, the detection rate of the FBH was 87% while only 20% for the SDH. The smallest detected FBH had a 0.56 wavelength depth and a lateral extent of 1.04 wavelength. Drawing a box using the  $-6$  dB drop method around the FBH always led to an IoU under 15%.

On average, the lateral extent of the FBH using the  $-6$  dB method was three times larger than the diameter predicted by the proposed method. Then, the training was continued with a small augmented dataset of experiments (equivalent to 3% of the simulated dataset). In experiments, the results show that the test specimen was always correctly identified. When using a 40% IoU threshold the experimental detection rate of the FBH was 70%. The smallest detected defect in experiments had a depth of 2 wavelengths.

## 2.2 Introduction

Defect detection is critical in high value-added industries. To ensure customer satisfaction and/or safety, produced parts must be controlled pristine. Ultrasonic testing (UT) is one of the most common inspection method used to detect flaws such as cracks, voids or inclusions. Nevertheless, highly trained inspectors are required to identify flaws in UT scans. Ultrasonic phased array probes, combined with heavy signal and image processing, are the key to successful inspections. However, in high-speed manufacturing, the acquisition process may become too slow because of the numerous transmissions required.

A recent paper by Pavlov *et al.* (2019) used a generative adversarial network to achieve a pixel-wise speed of sound reconstruction from a plane wave ultrasonic insonification. The time taken by their acquisition process was shorter than the full matrix capture (FMC) required for the total focusing method (TFM). Moreover, their method was used to generate an image equivalent to a TFM reconstruction without the heavy calculations typically associated with the image formation. However, their method was not able to automatically locate and size defects. This paper presents a method using convolutional neural networks (CNN) to achieve high-speed inspection and flaw identification from a single plane wave insonification.

It has been shown that CNN can be used to detect the presence or to state on the absence of flaws in A-scans, even in noisy situations (Virupakshappa *et al.*, 2018). Munir *et al.* (2019) also demonstrated that CNN can classify defects in welds, still using only A-scans. For such detections, a large set of classified experimental acquisitions is required. However, generating

large sets of experimental measurements is difficult since, for every inspection, the sample must be manufactured, and the scan classified. To overcome this challenge, data augmentation may be used. Rotation, translation, shearing and other geometric modifications were implemented by Virkkunen *et al.* (2019) in B-Scans to train a neural network capable of detecting the presence or absence of defects in pipes with better accuracy than humans.

In this paper, the challenge is not only to detect a defect using a Time Trace Matrix (TTM), which is a juxtaposition of amplitude time traces acquired after firing a single plane wave and arranged with the probe elements order, but to size it as well. CNN are commonly used today in computer vision to spot and classify the different elements of an image or a video. Ren *et al.* (2015) were the first to present Faster R-CNN in a bid to achieve the preceding objectives as quickly as possible. This network is a faster version of Fast R-CNN, which was already a faster version of R-CNN. R-CNN extracts region proposals before applying the CNN, and then classifying regions. Fast R-CNN reduces the computation time by applying the CNN before the region proposal algorithm. Finally, Faster R-CNN reduces the computation time further by eliminating the selective search algorithm and instead using a separate region proposal network (RPN) after the CNN. This neural network architecture is used in this paper to square the defect, and consequently, to evaluate its size.

It is clear that efficiency is generally directly related to the amount of data used for training. While data augmentation is useful in enlarging an experimental dataset, here, the objective is to inspect parts using a minimal amount of experimental data. To this end, GPU-accelerated finite element (FE) simulations are used to generate large scan datasets. Chapon *et al.* (2019) showed that by training a CNN with FE-simulated A-Scans, experimental A-Scans could be deconvolved, even in the presence of strongly overlapping wave packets. The work also demonstrated the need to add noise in order to improve the realism of FE simulations.

One idea investigated in this paper is to use simulation to generate the vast majority of the training dataset and then augment it with a small subset of experimental data. This idea of adapting a preliminary training was recently implemented by Belharbi *et al.* (2019). Using preliminary

training enables contrast detection, which allows the neural network to identify different shapes in input images. Since shape detection is common in all object detection, it can be done with any image datasets. The second training phase allows adapting the classification process and pattern recognition to the specific type of objects of a study. In Belharbi *et al.* (2019) study, a pretrained CNN with a large photograph dataset was supplemented with a low percentage of medical X-ray scans. Ultimately, the neural network allowed the detection of tumours, even if the pretrained network was totally unable to do so.

Using all these concepts, this paper demonstrates that a CNN mostly trained with simulated data and with a small amount of experimental data can be used to detect, locate and size a defect from a single plane wave insonification. The paper begins with the materials and methods section, which describes the finite element model and the plane wave insonification FE simulations employed. This is followed by the Faster R-CNN adaptation and training, which details how the original Faster R-CNN developed for computer vision was adapted to UT images. Next, the results section presents the test of the neural network, followed by discussions. Finally, conclusions are drawn.

## **2.3 Materials and methods**

### **2.3.1 Finite Element simulated dataset**

The challenge in this study was to train the neural network using simulations. The dataset was generated using GPU-accelerated FE models representing an immersed aluminium block. This block either had one side-drilled hole (SDH) or one flat bottom hole (FBH) whose face is parallel to the block surface. Therefore, no more than one defect was included per simulation. In practice a defect is a rare occurrence therefore a part with more than one defect under the probe would be highly unusual. Figure 2.1(A) shows the experimental setup and (B), the 2D plane strain equivalent model. The model was discretised at 20 elements per wavelength in water ( $\lambda_w$ ) using linear square elements. The properties of aluminium, water and piezoelectric material are summarised in Table 2.1. Pogo FEA (Huthwaite, 2014) was used for FE simulations. This

simulation code does not include acoustic elements but only solid elements defined by their density, Poisson's ratio and Young's modulus. The major challenge was then to model water as a solid. The density ( $\rho = 1000 \text{ kg m}^{-3}$ ) and the bulk modulus ( $K = 2.2 \text{ GPa}$ ) of water are well known. The bulk modulus ( $K$ ), Poisson's ratio ( $\nu$ ) and Young's modulus ( $E$ ) are linked together by relation 2.1 :

$$E = 3K(1 - 2\nu) . \quad (2.1)$$

The longitudinal speed of sound in a solid depends on the Young's modulus, the Poisson's ratio and the density as detailed in equation 2.2 :

$$c_w = \sqrt{\frac{E(1 - \nu)}{\rho(1 - \nu)(1 - 2\nu)}} . \quad (2.2)$$

The speed of sound in water ( $c_w$ ) at  $20^\circ\text{C}$  is  $1480 \text{ m s}^{-1}$ . Equations 2.1 and 2.2 can therefore be used to determine the properties required to model water as a solid. Simulation results showed that shear waves propagation resulting from this approximation were 16 dB lower than longitudinal waves.

Figure 2.1(B) also shows the absorbing boundaries (Rajagopal *et al.*, 2012) that were added in order to reduce the mesh size and avoid reverberations. The  $X$  and  $Y$  dimensions were set to 70 mm and 40 mm, respectively.

Tableau 2.1 Materials Properties

| Materials                      | Aluminium | Piezoelectric | Water |
|--------------------------------|-----------|---------------|-------|
| Young Modulus (GPa)            | 70        | 78            | 0.04  |
| Poisson ration                 | 0.33      | 0.31          | 0.497 |
| Density ( $\text{kg m}^{-3}$ ) | 2700      | 7600          | 1000  |

The adequate nodes were excited in order to model the Verasonics P4-2v ultrasonic phased array probe. This probe has 64 elements with a width of 0.2 mm and a pitch of 0.3 mm. The pulse was

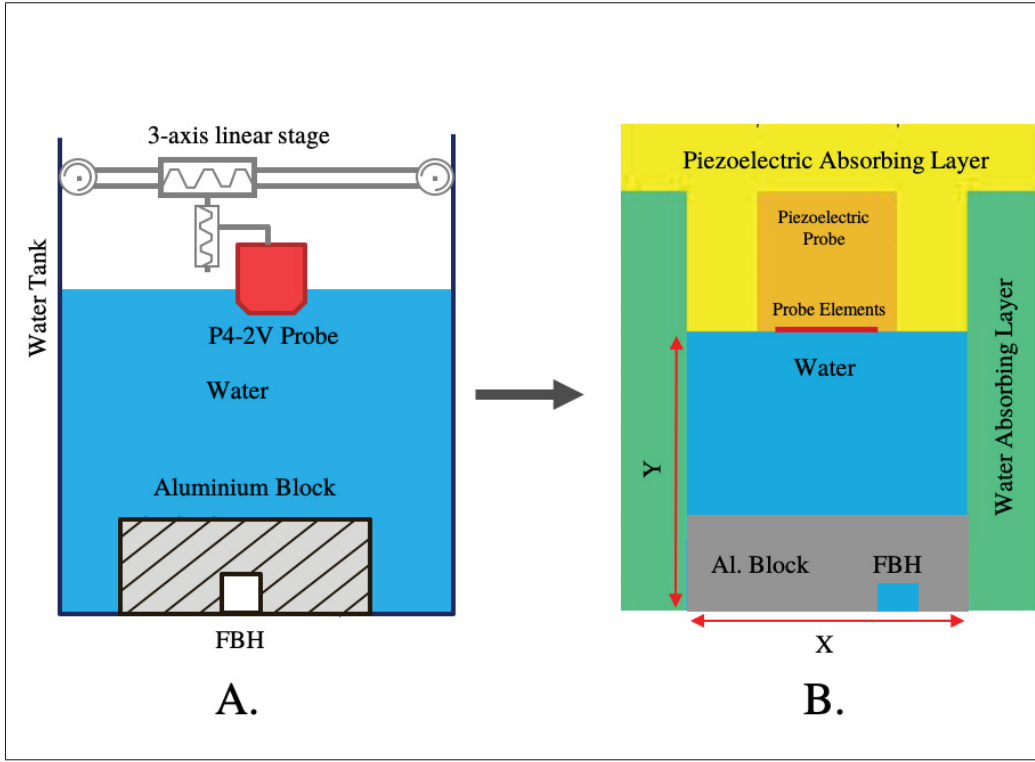


Figure 2.1 Experimental setup (A) and a schematic of FE model (B).

a 3-cycle Hann windowed signal centred on 2.5 MHz. This probe was used in the model as it was readily available in the laboratory for the experimental validation.

The FBH and SDH sizes and positions were randomly generated in order to obtain the most diverse dataset possible. Considering the P4-2v probe centre frequency of 2.5 MHz and the wavelength in aluminium as a reference ( $\lambda_{Al}$ ), the dimensional ranges of the holes are summarised in Table 2.2 in millimetres and relative to the wavelength in aluminium at 2.5 MHz. The holes were filled with water, and all the positions and sizes of the reflectors, as well as the heights of the aluminium block, were stored in memory in order to be used during the supervised learning of the neural network.

Using Pogo FEA (Huthwaite, 2014), these models were generated, and then a single plane wave insonification was simulated for each of them. The same nodes were used for the signal emission

Tableau 2.2 Model dimension ranges

|        |                | FBH          |                | SDH          | Aluminium Block |
|--------|----------------|--------------|----------------|--------------|-----------------|
|        |                | Depth        | Lateral Extent | Diameter     | Height          |
| Ranges | mm             | [0.50 ; 7.0] | [1.0 ; 8.0]    | [1.0 ; 8.0]  | [40 ; 50]       |
|        | $\lambda_{AI}$ | [0.20 ; 2.8] | [0.40 ; 3.2]   | [0.40 ; 3.2] | [16 ; 20]       |

and reception. Each of these simulations produced an 8192 by 64 matrix of time traces , which can be considered as a noiseless plane wave TTM. 2000 cases were simulated over two days on a workstation equipped with two NVIDIA Quadro P6000.

### 2.3.2 Faster R-CNN adaptation and development

Faster R-CNN (Ren *et al.*, 2015) is well known for its computer vision applications. A web article by Xu (2018) and the python code provided were used in the present study. Faster R-CNN uses Keras and Tensorflow libraries. This neural-network was developed to be trained with large datasets of Red-Green-Blue (RGB) low-resolution images (a few hundred pixels width and height) to shorten the processing. While RGB images are 3-channel data inputs, here, the plane wave raw data inputs were single-amplitude layers. The code was modified to be able to process single-layer inputs as well as the specific TTM input sizes. In fact, the neural network could have been trained with RGB TTM images, but tests showed that it was 30 times faster to use single-layer amplitude matrices. As TTM dimensions are very different from those of standard images, it was decided not to use a pretrained network. Figure 2.2 shows an example of a TTM and the desired prediction after neural network processing.

The convolutional neural network composing the whole Faster R-CNN used in this study was a custom VGG16 (Figure 2.3) which is a simple architecture. A relatively basic network was chosen to enable fast computation and allow multiple tries. The parameters of the network were only defined following multiple tests in order to obtain the best possible efficiency. In order for the Region Proposal Network (RPN) to have maximum context for pattern recognition, max pooling operations decreased the data inputs time trace dimension but preserved the 64 A-Scan columns. After the last convolutional layers, a dropout layer with a rate of 0.2 was set. This layer

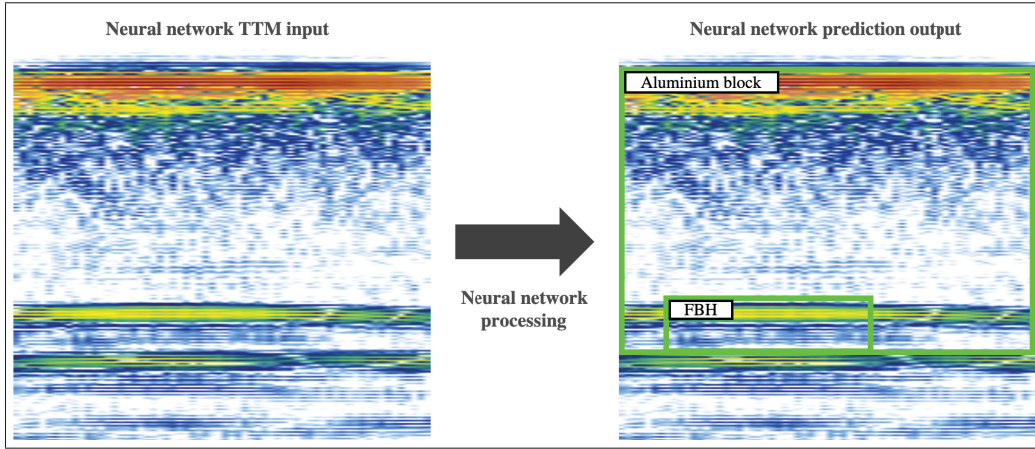


Figure 2.2 Example of a neural network prediction from a raw TTM. The green boxes associated with their label, classify, locate and size the objects.

is expected to avoid overfitting (Srivastava *et al.*, 2014), or specialisation of the CNN to the simulated dataset. Last custom adaptations were applied to the RPN, where the 3 anchors scales  $(i, j)$  and 3 ratios  $(k)$  were calculated using equation 2.3 to allow fitting the different defect and block areas defined by width and height  $(w, h)$  :

$$(w, h) = (i, j) \times k . \quad (2.3)$$

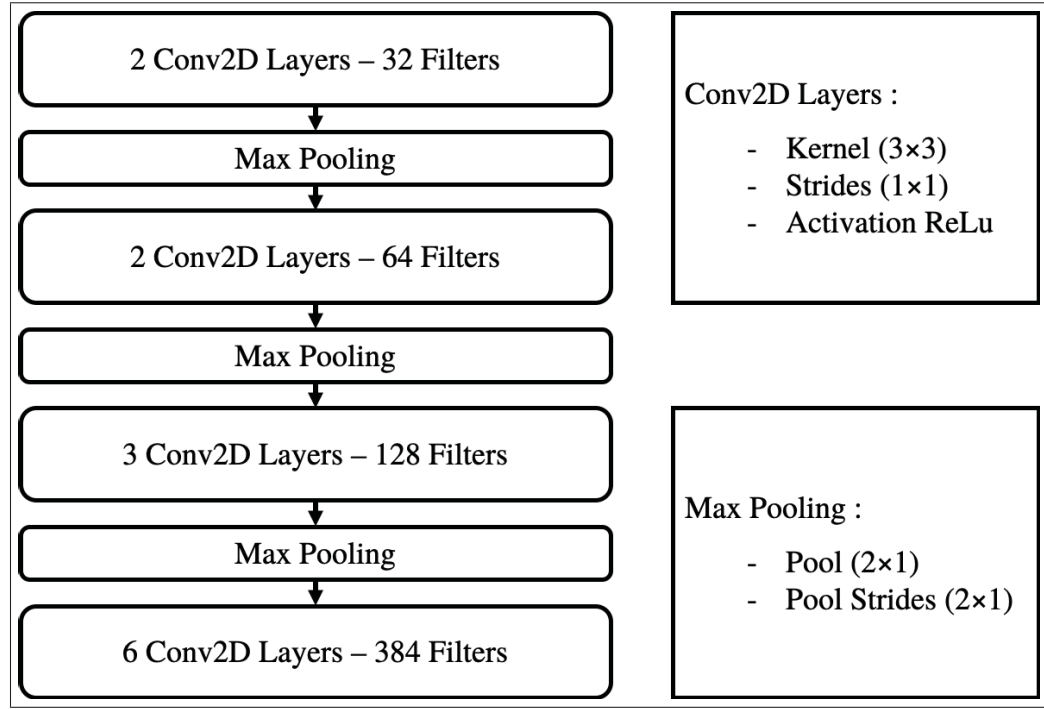


Figure 2.3 CNN architecture.

### 2.3.3 Training the neural network with simulations

In order for each simulation to represent a true experiment, a noise matrix ( $N$ ) was added to each raw simulated amplitude matrix ( $M_{\text{raw}}$ ).  $N$  amplitudes were calculated using equation 2.4 :

$$N = N_{\text{wn}} \frac{10^{-\frac{A_{\text{dB}}}{20} \max(M_{\text{raw}})}}{\max(N_{\text{wn}})} . \quad (2.4)$$

$A_{\text{dB}}$  is a random SNR value ranging from 30 dB to 1 dB.  $N_{\text{wn}}$  is a white noise matrix randomly generated from an uniform distribution and then bandpass-filtered to fit the P4-2v probe bandwidth. After adding  $N$  to  $M_{\text{raw}}$ , a downsampling was applied to divide each input time-trace size by 16, ending as 512 by 64 amplitude matrixes ( $M_{\text{noisy}}$ ). One of the goal was to fit the same size limitation of the experimental setup and another goal was to reduce the calculation cost. Finally, the noisy simulated amplitude matrix ( $M_{\text{noisy}}$ ) was converted to decibel matrixes ( $M_{\text{dB}}$ ) using equation 2.5.

$$M_{\text{dB}} = 20 \log_{10} \left( \frac{M_{\text{noisy}}}{\max(M_{\text{noisy}})} \right). \quad (2.5)$$

This whole process was repeated for each sample of each batch of each epoch of the training. Meaning that each input process was different even if the original noise-less TTM dataset used was the same in each epoch.

Each batch was composed of the whole FE simulated dataset, containing 500 FBH, 500 SDH and 1000 aluminium block with no defects. At maximum the training went through 550 epochs. This is quite low compared to similar neural network but considering the diversity of the dataset and the learning curve horizontal asymptote (Figure 2.4) it would have been useless to train for more epochs.

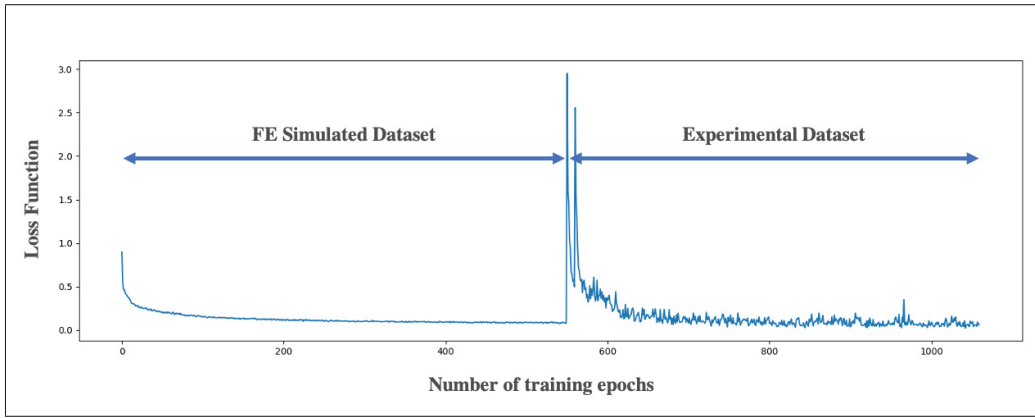


Figure 2.4 Learning curve and loss function as a function of epoch, starting with FE simulated dataset and following with experimental dataset.

### 2.3.4 Experimental setup

An aluminium block was used to generate an experimental dataset. 16 FBH were machined with depths ranging from 4.67 mm ( $\approx 2 \lambda_{\text{Al}}$ ) to 13.77 mm. The FBH had a diameter of 8 mm. The Verasonics P4-2v probe was attached to a 3-dimensional linear stage, which allowed the probe to be moved over the immersed FBH; 3 different acquisitions were made for each FBH, with

the water path length between the probe and the aluminium block varying, as was the lateral alignment between the probe and the FBH. The positions of the probe were stored in order to label the acquisitions with the block and the FBH properties. Using a Verasonics Vantage 64LE data acquisition system, the probe was programmed to fire a 3-cycle Hann windowed plane wave insonification with a centre frequency of 2.5 MHz. After acquisition, the TTM were normalised following the same procedure as for the simulated TTM.

### 2.3.5 Additional training with experiments

In order to improve the previous training, the CNN trained only with FE simulations was then additionally trained with experimental acquisitions (transfer learning). The experimental dataset was first augmented. This method is commonly used in machine learning to enlarge small training datasets. The augmentations consisted of an axial symmetry around the vertical axis, which doubled the experimental dataset to 96 acquisitions. 80% of the acquisitions were used for training and 20% left for the tests. 550 more training epochs were completed. The experimental training loss function (Figure 2.4) reached the same asymptotic value as with the FE simulation training. Transfer learning is a common method used for image recognition networks. Usually, CNN networks are pre-trained with generic images and then trained again with more specific images (Belharbi *et al.*, 2019; Razavian *et al.*, 2014; Yosinski *et al.*, 2014).

### 2.3.6 Performance Metrics

After the tests, the Faster R-CNN outputs were broken down into four groups : true positives, true negatives, false positives and false negatives. Testing one TTM gave a true positive result when the object (FBH, SDH or Aluminium block) was predicted, and there was indeed one. A true negative result was obtained when no defect was predicted by the neural network in a pristine block. These 2 values must be maximised and their counterparts (false positive and false negative) minimised to obtain the best classification results. It is important to minimise the false negative rate as it represents how much confidence can be given to the automated defect detection method. On the other hand, false positives represent unnecessary rejects. To quantify

the success of the testing in locating and sizing the different objects classified, the IoU criterion of equation 2.6 was used :

$$\text{IoU}(A_i, A_u) = \frac{A_i}{A_u} . \quad (2.6)$$

$A_i$  is the area of the intersection between the ground truth boxes and the predicted boxes and  $A_u$  is the area of the union of these two boxes.

IoU is a measurement of the overlap usually used to test Faster R-CNN performance. It is better than pure intersection as it takes into account sizing. The larger the difference between the size of the ground truth box and the size of the prediction box, the lower the IoU score. An example is given Figure 2.5. In this study, an IoU of 40% was often used as a minimum threshold for considering a true positive prediction as a good prediction. For two equal size objects, a 40% IoU represents an intersection of 60% of the equal object area.

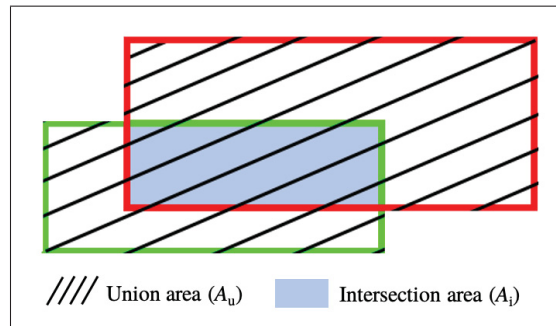


Figure 2.5 Example of two boxes (red and green) IoU. In this example the, IoU is 25%

The  $-6$  dB drop is a method commonly used to size defects in ultrasonic testing (Felice & Fan, 2018). For FBH, the method can be used to evaluate the lateral extent. The lateral extent is calculated by measuring the distance between the maximum reflected amplitude and the points at  $-6$  dB from the maximum reflected amplitude. This method will be used for comparison with the neural network predictions.

## 2.4 Results

### 2.4.1 Tests of the simulation-only training

After training the neural network only with simulations, tests were performed on 200 new FE simulations (50 FBH, 50 SDH and 100 pristine aluminium blocks). Figure 2.6 shows classification test results.

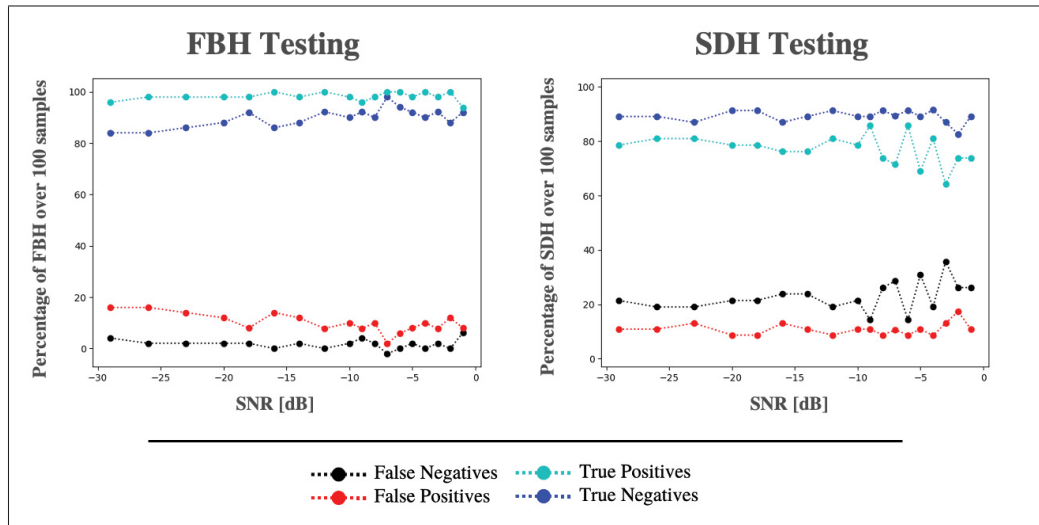


Figure 2.6 Neural Network test results over FE simulations. FBH and SDH testing results in function of SNR.

In view of the 2 graphs, in both cases it was interesting to point out that the classification performance on FBH and SDH is quite stable from 30 dB to 1 dB SNR. It was then encouraging for the neural network to be able to adapt experimental situations without noise-filtering. One criteria pointing out a good classification is the receiver operating characteristic (ROC) curve. This curve is shown Figure 2.7 for a 7 dB SNR test. The area under the curve (AUC) should be as closed as 1 as possible and here AUC is 0.95.

Concerning the aluminium block, it was correctly detected in 100% of tested cases considering an IoU superior to 90%. As it was possible to be confident with the block sizing and localisation, this information was used to assist the neural network locating FBH and SDH. In fact, when the

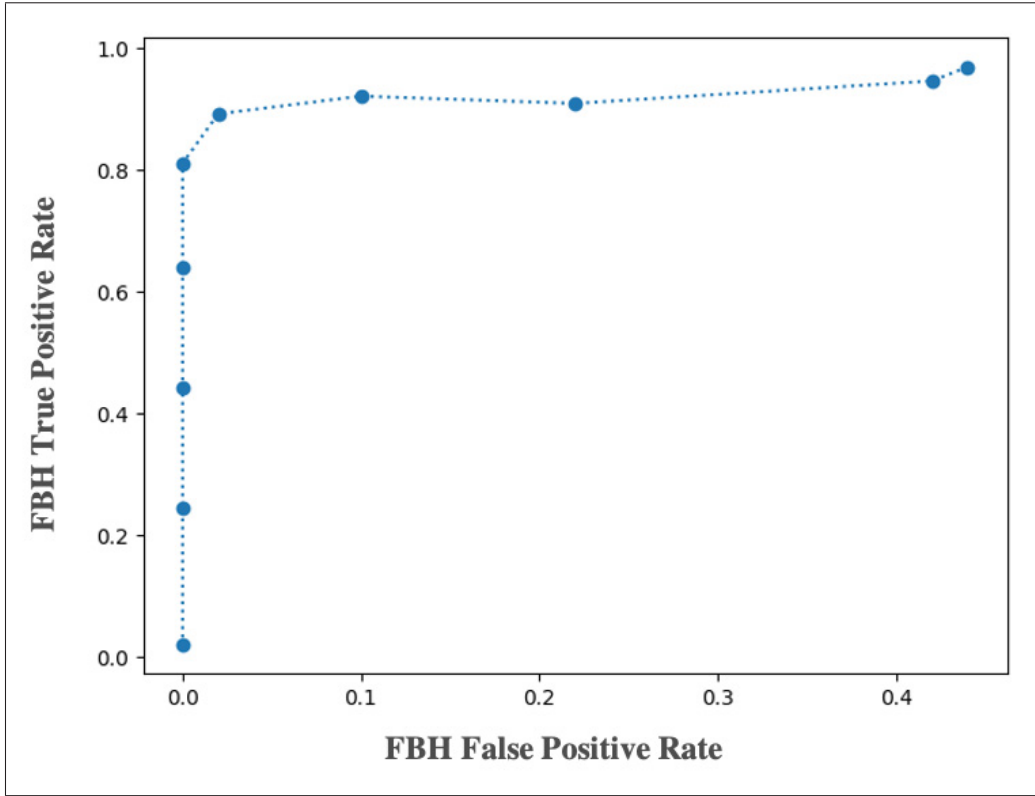


Figure 2.7 ROC Curve of neural network test results over FBH FE simulations.

neural network processed a TTM it made multiple predictions for the same objects and gave it a 0 to 100 score. The part prediction with the highest score was selected and used to point out the highest FBH or SDH prediction score which was located for sure inside this block. With a fixed SNR of 7 dB, Figure 2.8 presents classification, localisation and sizing test result using the IoU criterion for FBH and SDH.

The best results were obtained for FBH prediction. For that kind of defect, false negatives represent less than 5% of the predictions. False positives account here for 10% of the predictions. Then, the IoU criteria shows that considering at least 40% of IoU, 87% of the FBH tested were correctly predicted by the neural network. The smallest simulated FBH predicted was  $0.56 \lambda_{A1}$  depth and  $1.04 \lambda_{A1}$  of lateral extent.

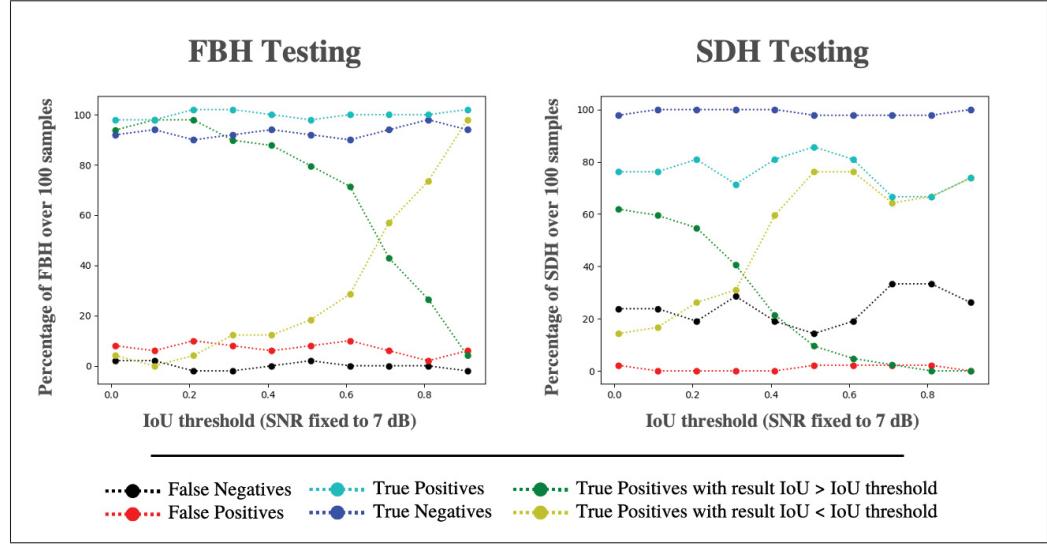


Figure 2.8 Neural Network test results over FE simulations. FBH and SDH testing results in function of IoU Threshold with SNR fixed to 7 dB.

## 2.4.2 Tests on experimental acquisitions

After additional training using a small amount of experimental data, the CNN was able to identify FBH in the aluminium block. All the aluminium blocks ("Al. Block" label) were sized and located when using an IoU of 90%, and 70% of the FBH were correctly predicted with an IoU of 40%. Figure 2.9 shows a typical example of the CNN predictions on a plane wave TTM obtained from the test dataset. False negatives represented 10% of the predictions and the last 15% are predictions with an IoU of less than 40%. The minimum FBH depth detected was  $2 \lambda_{Al}$ .

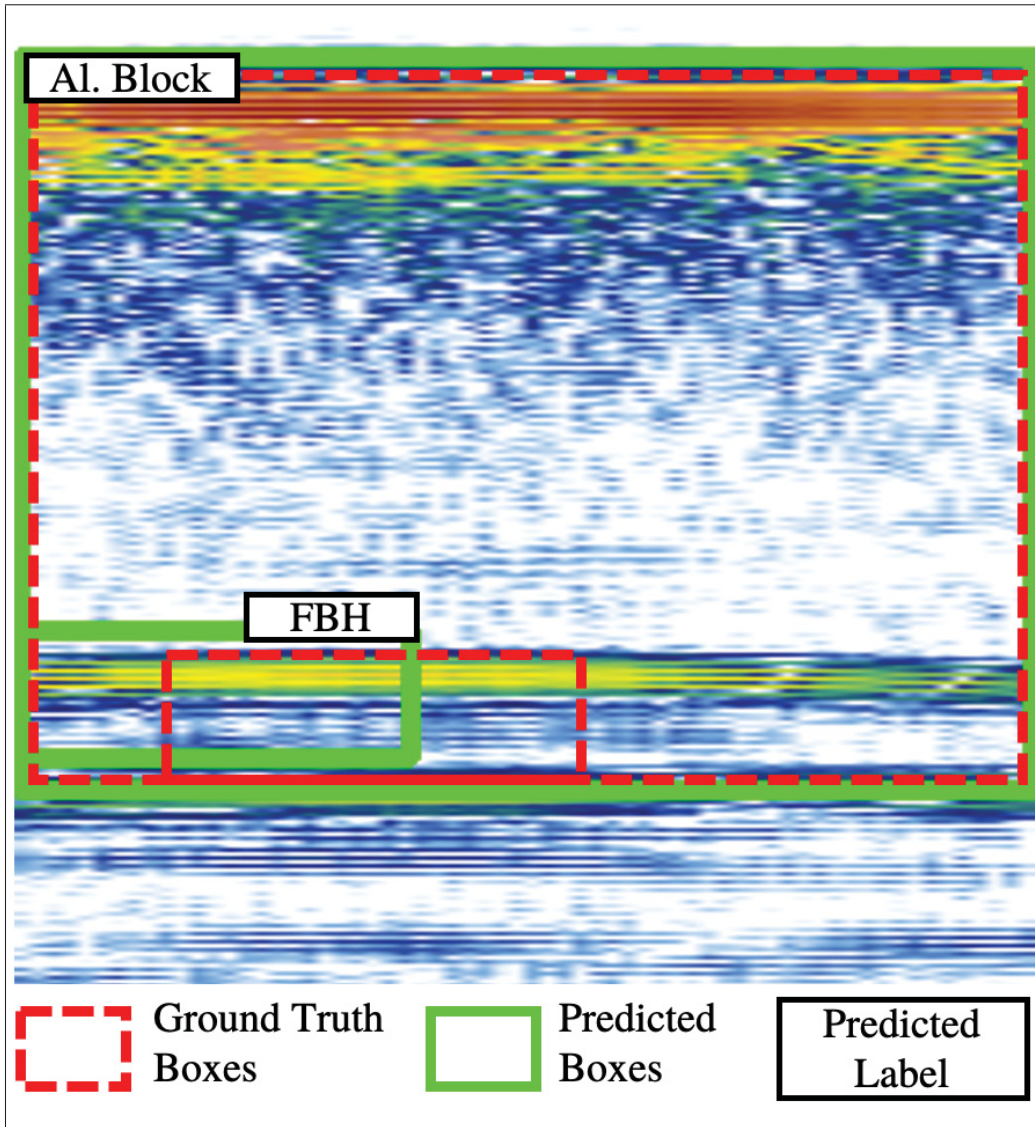


Figure 2.9 Faster R-CNN predictions on an experimental plane-wave insonification TTM of an immersed aluminium block with an FBH.

### 2.4.3 Comparison with the $-6$ dB drop method

Applied to the same dataset used to test the CNN, the  $-6$  dB drop method showed sizing results less accurate than the CNN. On average, both the area calculated with this method and the ground truth area had an IoU of 15%. The calculated lateral extent of the FBH was 3 times larger than the diameter predicted by the CNN. Figure 2.10 shows a comparison between the  $-6$  dB

drop method and the CNN prediction. The  $-6$  dB drop method was always centred on the defect but overestimate the lateral extent because of the echo diffusion.

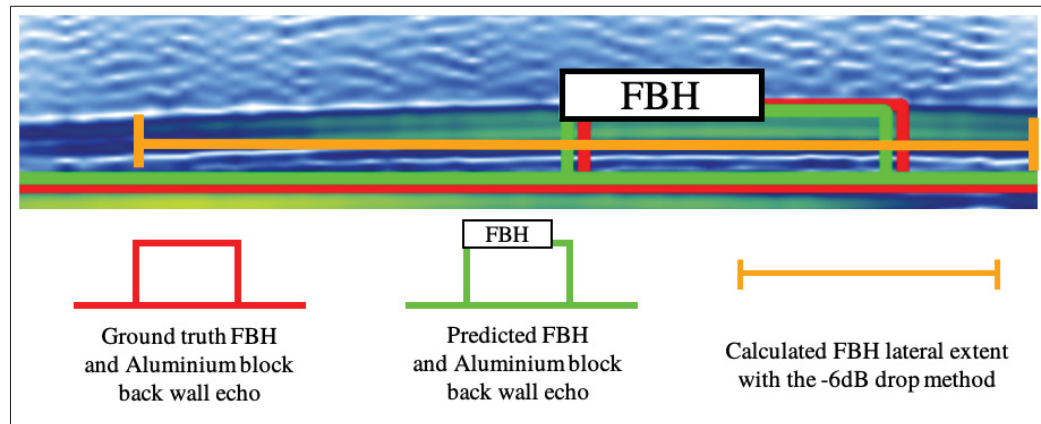


Figure 2.10 Comparison between Faster R-CNN predictions and  $-6$  dB drop method on an FE simulated plane-wave insonification TTM of an immersed aluminium block with an FBH

## 2.5 Discussion

Before the second training with the experimental dataset, experimental TTM were tested directly with the simulation-only training. In that scenario, the CNN was only able to predict half of the block sizes and positions from the experimental TTM. This observation led to the conclusion that it was essential to supplement the training with experimental data. Using a simulation-only training to test experimental data is challenging because simulations do not represent a perfect model of true experiments. The surface of the aluminium block and the bottom of the FBH were not conditioned and the water was not degassed, and was not stationary. This study aimed in part to obtain good predictions from a raw protocol in order to suit industrial situations. Combined, the preceding conditions led to coherent noise, which was not simulated. Due to this noise, the FBH and backwall echoes appeared superposed when the FBH were too shallow. Chapon et al. (2019) showed that by training a CNN with an FE-simulated A-Scan, experimental A-Scans could be deconvolved, even in the presence of strongly overlapping wave packets. It is therefore

anticipated that by using a larger training dataset modelling the coherent noise, it would be possible to detect FBH even when its echo is overlapping with the backwall.

Moreover, the experimental dataset comprised only 8 mm diameter FBHs, whereas the randomly generated FBH diameters of the simulated dataset ranged from 1 mm to 8 mm. It could have been possible to simulate only 8 mm diameters, but the training would not have been diversified. In fact, the ratio of the probe aperture over the 8 mm FBH diameter would have led to the generation of a very low number of different possible simulations. It would have certainly led to a non-versatile training, limited only to a specific size of FBH. This necessity of training with the most diverse dataset, even if it is not composed exactly like the test dataset, was observed. Results with FBH were better with a training combining SDH and FBH simulations than with an FBH-only training.

By using a variety of SNR during training, it was shown that the network could maintain its performance for different SNR in the test dataset. Moreover, a multi-SNR training and a  $\Delta_{dB}$  fixed SNR training were compared on a  $\Delta_{dB}$  fixed SNR test. Results were up to 10% better with a multi-SNR training than with a single-SNR training. In fact, adding different noise amplitudes led to a more diversified training dataset, and this diversity is a key to a successful training. This observation led to the conclusion that it was useless to perform separate training at fixed SNR.

Tests on SDH did not perform as well as test on FBH. In fact, FBH could always be located just above the backwall echo whereas SDH could be located anywhere between the backwall and the surface echoes. This additional degree of freedom concerning the SDH location implies the necessity of a larger training dataset. The neural network is more likely to converge with FBH as their depth location range is more restricted.

No correlations appeared between the size or location of the FBH and the success of the predictions. Some predictions performed well even with the smallest FBH, while some of the largest FBH could lead to false negative predictions. The smallest simulated FBH predicted was a  $0.56 \lambda_{A1}$  depth and a  $1.04 \lambda_{A1}$  lateral extent. Other correctly predicted defects had dimensions

just above the theoretical limit of  $0.5 \lambda_{AI}$ . Therefore, the false negatives are thought to be due to a too small training dataset rather than a problem of method and neural network parameters.

A decision was made not to use the mean average precision score to measure the neural network performance. Usually, CNN uses this metric for comparisons. In this study, as there is only one defect per TTM, the prediction is more like a medical test, so it appeared more relevant to detail false positives, false negatives, true positives and true negatives. It is then easier to judge if it is possible to be confident with the neural network detection capability.

Since it was difficult to predict SDH, and FBH predictions results did not reach perfection, it would be interesting to set up a larger classified TTM dataset with a greater variety of defects. For FBH there were 660 different possible cases but 1,890 for SDH. The training comprised 500 simulations for each defect, with a total of 3 classes (FBH, SDH, aluminium block). Comparatively, the computer vision neural network dataset PASCAL Visual Object Classes (VOC) 2010 Challenge (Everingham *et al.*, 2010) comprised 10,103 images with a total of 20 classes. These images represent more diverse scenarios than the FE simulated TTM of this paper, thus leading to a better training. In order to set up a dataset as large as the PASCAL VOC challenge one, the training dataset could also be composed of simulated and experimental acquisitions with various transducer bandwidths.

A possible limitation of the method presented in this paper is the presence of only one defect in each experimental or simulated TTM. A second study should consider more reflectors in order to open this method up to more varied situations. However, defects are unusual during inspections in the field and the likelihood of a multiple defects under a probe is low.

Finally, this paper presents only one neural network architecture among many others which could have been applied to this case study. In fact, it could be possible to test Mask R-CNN, which is usually a neural network labelling each pixel of an image. Another possibility would be a CNN followed by deconvolutional neural network whose goal would be to generate an image representing the inspected part as a function of an input TTM. Also testing other types of training, involving scenarios such as semi-supervised, unsupervised or generative adversarial

networks could be interesting. Ultimately, it is possible to pre-process the input data with low calculation charge algorithm in order to train the neural network with higher quality data while respecting the real-time constraint. A partial reconstruction would require knowledge on the inspected part defects, however, this is rarely possible in high-speed manufacturing.

## 2.6 Conclusion

This paper investigated the use of CNN to locate and size FBH from plane wave insonification TTM. Trained using 96% simulated data and 4% experimental data, the method developed in this paper could be applied to other inspection scenarios whereby experimental data are difficult to generate. A deliberate choice was made to use a simple neural network architecture and TTM formed from a single plane wave emission without imaging calculations in order to apply this concept to high-speed manufacturing. first, tests on the simulated dataset showed results independent of SNR. Then, when tested on experimental data, and considering a minimum IoU of 40%, the CNN was able to locate and size 70% of the FBH. The minimal FBH depth predicted was approximately  $2 \lambda_{A1}$ .

## 2.7 Acknowledgements

We would like to thank Olympus (Quebec) for supporting this research paper, and for providing equipment, funding and project monitoring. The authors would also like to thank Compute Quebec (<https://www.calculquebec.ca>) and Compute Canada (<https://www.computecanada.ca>) for their computation support.

## CONCLUSION ET RECOMMANDATIONS

Cette étude présente une preuve de concept intéressante et prometteuse. Grâce à un réseau de neurones, il est possible de déterminer, d'une part, la présence ou non d'un défaut dans une pièce et, d'autre part, de le dimensionner et de le localiser dans la pièce inspectée. Plus de 70 % des FBH sont dimensionnés et localisés avec une IoU de 40 % et une influence minimale du SNR. Les bonnes détections ont lieu jusqu'à un minimum de 2 longueurs d'ondes à 2,25 MHz. Il est ensuite tout à fait possible de travailler à des fréquences plus élevées ou plus basses en adaptant le modèle de simulation. En effet, le réseau de neurones est entraîné à 97 % grâce à des simulations numériques. Cela lui procure l'adaptabilité attendue et il est possible de préparer le réseau de neurones en amont de l'inspection grâce à un minimum de mesures expérimentales fastidieuses à effectuer et à classer. Il est important de rappeler que la pièce inspectée dans cette étude est placée en immersion. La sonde n'est donc pas en contact direct avec la pièce et cela contribue aux atténuations, réflexions et interférences supplémentaires.

En simulation comme en expérimentation, le dimensionnement des FBH, grâce au réseau de neurones, s'avère très proche du dimensionnement réel, entre 80 % et 120 % de ce dernier. Alors que la méthode rapide de dimensionnement des FBH à  $-6$  dB dimensionne le défaut comme 3 fois plus grand que la taille réelle. Ainsi, avec un temps de mesures identique, la précision du résultat est ici grandement augmentée. Cette rapidité de la prise de mesures et du calcul ouvre les possibilités de l'automatisation et de la haute cadence attendues. En effet, le traitement des données est minimal ; c'est à dire uniquement un passage à l'échelle logarithmique et la prise de mesures réside en une seule émission d'une onde plane sans phase de calibration. Pour ces deux raisons, il est également envisageable de proposer cette méthode pour une assistance au contrôle en temps réel.

Rappelons ici que les réflecteurs expérimentaux testés (les FBH) sont larges avec une géométrie simple. Alors que les tests effectués sur les SDH simulés qui possèdent un degré de localisation

supplémentaire et une géométrie moins réfléchissante n'ont pas encore atteint des taux de détection satisfaisants. Il serait donc très intéressant d'étoffer la diversité des modèles de simulation mais également d'augmenter la complexité géométrique des défauts simulés afin de trouver davantage d'applications industrielles. Il est également possible de tester d'autres architectures de réseaux de neurones plus complexes mais il faut s'attacher à conserver une possibilité de traitement en temps réel pour valider entièrement la preuve de concept.

## ANNEXE I

### COMPLÉMENTS DE RÉSULTATS

#### 1. Compléments sur les résultats des tests sur les SDH simulés

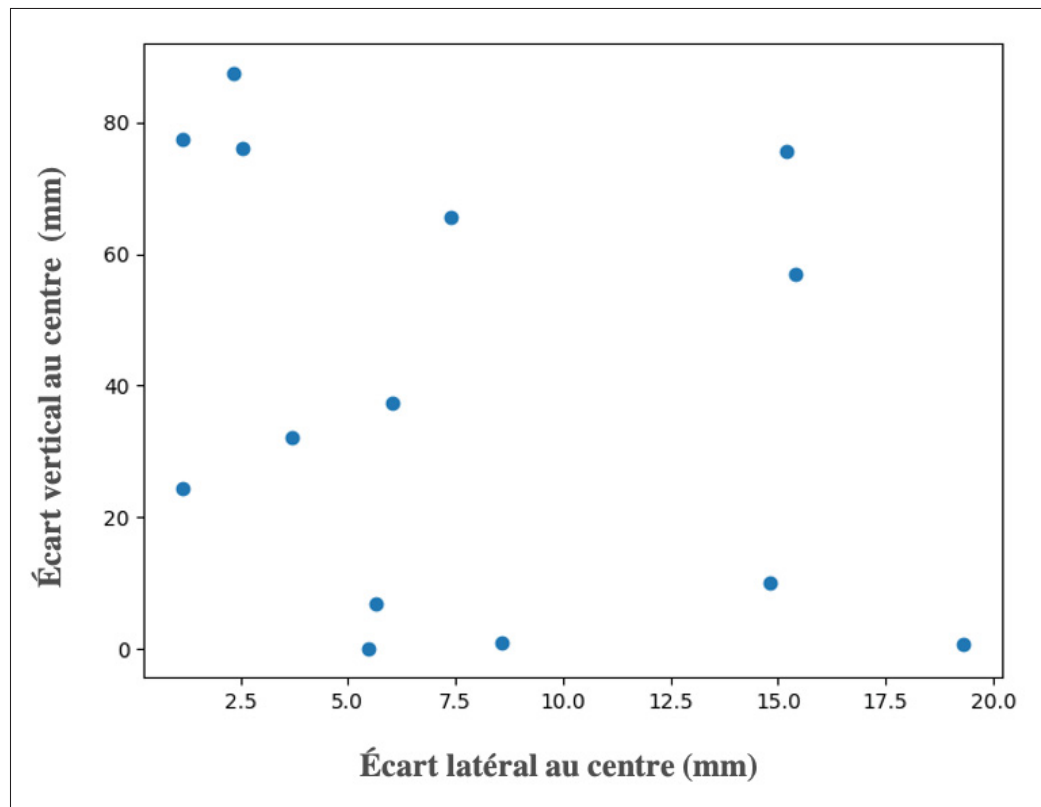


Figure-A I-1 Écarts verticaux et écarts latéraux entre les prédictions et les véritables coordonnées de chaque SDH dont l'IoU est nulle.

## 2. Compléments sur le transfert d'apprentissage

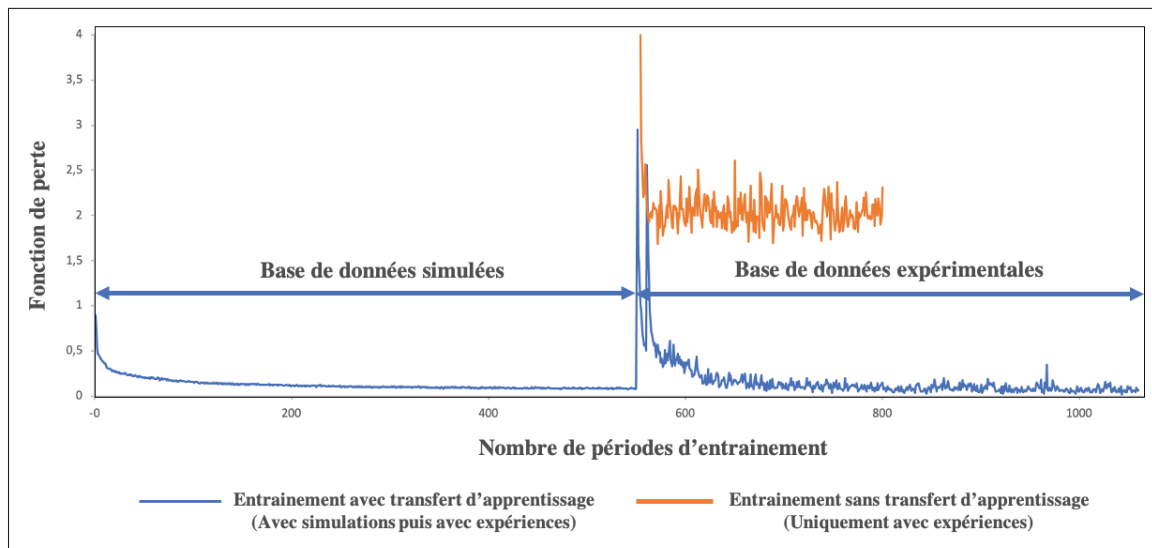


Figure-A I-2 Comparaison entre le transfert d'apprentissage du domaine de simulation au domaine expérimental et l'apprentissage purement expérimental. La base de données expérimentales est la même dans les deux cas. On remarque bien qu'avec si peu de données expérimentales, le transfert d'apprentissage est indispensable.

## BIBLIOGRAPHIE

- Barbedo, J. G. A. (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, 153, 46–53.
- Belharbi, S., Rony, J., Dolz, J., Ayed, I. B., McCaffrey, L. & Granger, E. (2019). *Min-max Entropy for Weakly Supervised Pointwise Localization*. ICLR 2020.
- Chapon, A., Pereira, D. & Bélanger, P. (2019). *Deconvolution of ultrasonic signals using a convolutional neural network*.
- Cheeke, J. (2002). *Fundamentals and Applications of Ultrasonic Waves*.
- Cook, R. D., Malkus, D. S., Plesha, M. E. & Witt, R. J. (2001). *Concepts and Applications of Finite Element Analysis, 4th Edition* (éd. 4). Wiley.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Li, F. F. (2009, 06). ImageNet : a Large-Scale Hierarchical Image Database. pp. 248–255.
- Drozdz, M., Moreau, L., Castaings, M., Lowe, M. & Cawley, P. (2006). Efficient Numerical Modelling of Absorbing Regions for Boundaries Of Guided Waves Problems. *AIP Conference Proceedings*, 820, 126–133.
- Durand, T. (2017). *Weakly supervised learning for visual recognition*. (Thèse de doctorat). Repéré à Thèse de doctorat dirigée par Cord, Matthieu et Thome, Nicolas Informatique Paris 6 2017.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2), 303–338.
- Fatemi, M. & Kak, A. C. (1980). Ultrasonic B-Scan Imaging : Theory of Image Formation and a Technique for Restoration. *Ultrasonic Imaging*, 2(1), 1–47.
- Felice, M. V. & Fan, Z. (2018). Sizing of flaws using ultrasonic bulk wave testing : A review. *Ultrasonics*, 88, 26–42.
- Fukushima, K. (1980). Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, 193–202.
- Huthwaite, P. (2014). Accelerated Finite Element Elastodynamic Simulations Using the GPU. *Journal of Computational Physics*, 257, 687—707.

- Kanopoulos, N., Vasanthavada, N. & Baker, R. L. (1988). Design of an image edge detection filter using the Sobel operator. *IEEE Journal of solid-state circuits*, 23(2), 358–367.
- LeCun, Y., Haffner, P., Bottou, L. & Bengio, Y. (1999). Object Recognition with Gradient-Based Learning. Dans *Shape, Contour and Grouping in Computer Vision* (pp. 319–345). Springer Berlin Heidelberg.
- Mountford, R. A. & Wells, P. N. T. (1972). Ultrasonic liver scanning : the A-scan in the normal and cirrhosis. *Physics in Medicine and Biology*, 17(2), 261–269.
- Munir, N., Kim, H.-J., Park, J., Song, S.-J. & Kang, S.-S. (2019). Convolutional neural network for ultrasonic weldment flaw classification in noisy conditions. *Ultrasonics*, 94, 74–81.
- Pavlov, I., Prado, E., Navab, N. & Zahnd, G. (2019). Towards in-vivo ultrasound-histology : Plane-waves and generative adversarial networks for pixel-wise speed of sound reconstruction. *2019 IEEE International Ultrasonics Symposium (IUS)*, pp. 1913–1916.
- Rajagopal, P., Drozd, M., Skelton, E. A., Lowe, M. J. & Craster, R. V. (2012). On the use of absorbing layers to simulate the propagation of elastic waves in unbounded isotropic media using commercially available Finite Element packages. *NDT & E International*, 51, 30–40.
- Razavian, A., Azizpour, H., Sullivan, J. & Carlsson, S. (2014, 03). CNN Features Off-the-Shelf : An Astounding Baseline for Recognition. 1403.
- Ren, S., He, K., Girshick, R. & Sun, J. (2015). Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks. Dans Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M. & Garnett, R. (Éds.), *Advances in Neural Information Processing Systems* 28 (pp. 91–99). Curran Associates, Inc.
- Rosenblatt, F. F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6, 386–408.
- Schmerr, L. (2015). *Fundamentals of Ultrasonic Phased Arrays*.
- Shin, H., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D. & Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection : CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging*, 35(5), 1285–1298.
- Shull, P. (2002). *Nondestructive Evaluation : Theory, Techniques, and Applications*.
- Simonyan, K. & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556*.

- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56), 1929–1958.
- Virkkunen, I., Koskinen, T., Jessen-Juhler, O. & Rinta-Aho, J. (2019). *Augmented Ultrasonic Data for Machine Learning*. Repéré à [https://www.researchgate.net/publication/332035082\\_Augmented\\_Ultrasonic\\_Data\\_for\\_Machine\\_Learning](https://www.researchgate.net/publication/332035082_Augmented_Ultrasonic_Data_for_Machine_Learning).
- Virupakshappa, K., Marino, M. & Oruklu, E. (2018). A Multi-Resolution Convolutional Neural Network Architecture for Ultrasonic Flaw Detection. *2018 IEEE International Ultrasonics Symposium (IUS)*, 1–4.
- Weston, J., Ratle, F. & Collobert, R. (2008). Deep Learning via Semi-Supervised Embedding. *Proceedings of the 25th International Conference on Machine Learning, (ICML '08)*, 1168–1175.
- Xu, Y. (2018, November, 19). Faster R-CNN (object detection) implemented by Keras for custom data from Google's Open Images Dataset V4 [Tutorial]. Repéré à <https://towardsdatascience.com/faster-r-cnn-object-detection-implemented-by-keras-for-custom-data-from-googles-open-images-125f62b9141a>.
- Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems (NIPS)*, 27.