

UNIVERSITÉ DU QUÉBEC

**DÉTERMINATION DES CARACTÉRISTIQUES GÉOMÉTRIQUES D'UNE SOUDURE À PARTIR DE
DONNÉES ULTRASONORES EN TROIS DIMENSIONS À L'AIDE DE L'INTELLIGENCE ARTIFICIELLE**

**THÈSE PRÉSENTÉE À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES
COMME EXIGENCE PARTIELLE DU
DOCTORAT EN INGÉNIERIE
OFFERT EN EXTENSION
EN VERTU D'UN PROTOCOLE D'ENTENTE
AVEC L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI**

**PAR
ETIENNE DROUIN-PROVENÇAL**

AVRIL 2022

Université du Québec à Trois-Rivières
Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

UNIVERSITÉ DU QUÉBEC

DOCTORAT EN INGÉNIERIE (PH.D.)

Programme offert par l'Université du Québec à Chicoutimi (UQAC)

en extension avec

l'Université du Québec à Trois-Rivières (UQTR)

Cette thèse a été dirigée par :

Luc Laperrière, Ph.D., ing. Directeur de recherche

UQTR

Jury d'évaluation de la thèse :

François Gauthier, Ph.D, ing. Président du jury

UQTR

Luc Laperrière, Ph.D., ing.

UQTR

Sousso Kelouwani, Ph.D., ing. Membre

UQTR

Nicolas Giguère, Ph.D., ing. Membre externe

Centre de Métallurgie du Québec

Thèse soutenue le 18 août 2022

REMERCIEMENTS

L'aide, le dévouement et la collaboration d'un grand nombre de personnes ont été nécessaires afin de mener ce projet jusqu'au bout. Je tiens donc à remercier chaleureusement toutes les personnes qui ont contribué de près et de loin à l'accomplissement de mes travaux de recherche.

Je tiens à remercier vivement mon directeur de recherche Professeur Luc Laperrière de m'avoir guidé dans ce projet. Ses précieux conseils, sa grande disponibilité, son enthousiasme constant m'ont grandement aidé à mener cet ambitieux projet à terme.

Je veux remercier l'équipe de direction de Ondia, M. André Gosselin et M. Richard Rhéaume, de m'avoir offert cette merveilleuse opportunité et m'avoir accueilli chaleureusement dans leur entreprise. Le partage de leur expertise fut un facteur important dans la réussite de cette recherche. Le support que l'équipe de Ondia m'a offert a été un des principaux facteurs de réussite de ce projet. Je tiens donc à remercier toute l'équipe, actuelle et passée : Sofiane Hemch, Charles Umemoto, Larissa Figueiredo Terra de Faria, Samuel Poirier, Daniel Moises Gonzalez Clua, Alexandre Kapune, Nicolas Coulombe, Alexia St-Laurent, Rossana Modolin, Mélanie Beaulieu, Joël Drapeau, Maria Lina Strack et Yan-Éric Cossette. Il m'est important de souligner la grande contribution de mes collègues Maria Lina Strack et Yan-Éric Cossette à la réussite de cette recherche. Ils ont toujours été présents pour discuter des problèmes encourus, rechercher des solutions, partager leurs connaissances et leur bonne humeur. Il m'est primordial de remercier les deux experts en ultrasons, M. Neil Harrap et M. Richard Rhéaume, pour le partage de leur expertise reconnue mondialement, pour les heures passées à annoter les données ultrasonores et la validation de mon travail. Sans leur aide précieuse, ce projet n'aurait pas pu atteindre le niveau de qualité final.

Merci à Mitacs pour la bourse de recherche Accélération et au Conseil de Recherche en Sciences Naturelles et Génie (CRSNG) pour la bourse ES-D qui ont permis à ce projet de voir le jour.

Finalement, je tiens à souligner l'importance de ma conjointe Laurence Poulin dans la réussite de cette thèse. Durant ce doctorat, elle m'a aidé à tant de niveaux, entre autres, en tant que : collègue de bureau, conseillère, correctrice et coach. Je te remercie de m'avoir aidé et encouragé à réaliser ce projet d'envergure.

RÉSUMÉ

La dégradation des matériaux d'ingénierie est un phénomène naturel représentant des risques pour de nombreuses industries telles que le transport et le génie civil. Les essais non destructifs ont pour principal objectif d'acquérir des données sur les matériaux d'ingénierie afin de valider leur intégrité pour en assurer l'utilisation sans risque. En soudage, l'efficacité des essais non destructifs repose sur une acquisition cohérente des données à l'aide d'appareils ultrasonores et sur une analyse fiable réalisée par un expert. L'analyse des données permet d'extraire les informations sur les défauts (types, orientation, degré de risque) et d'émettre un diagnostic sur la soudure. Or, il y a actuellement une pénurie d'experts analystes en soudure et les délais pour procéder aux analyses s'allongent. Ceci peut parfois aboutir à des fraudes et des erreurs de manipulation humaine. Également, les analyses reposent sur des données brutes qui ne permettent pas de localiser la soudure de façon précise. Ce projet vise le développement, la validation et l'implantation de méthodes d'intelligence artificielle pour :

- a) Assurer l'intégrité et l'authenticité des données ultrasonores soumises pour la première fois ;
- b) Identifier les différentes parties d'une soudure ;
- c) Reconstruire la surface géométrique 3D d'une soudure ;
- d) Assurer un meilleur suivi des analyses périodiques d'une même soudure.

Un processus de détection des doublons a d'abord été développé. Le processus est adapté en fonction des deux types de doublons existants (*exact duplicate* et *near duplicate*). La première partie de ce processus utilise des fonctions de hachage (*hash*) pour générer une empreinte numérique à partir des données ultrasonores (S-Scan 3D). Cela permet de détecter rapidement et efficacement les *exacts duplicates*. La deuxième partie, qui se concentre sur la détection des *near duplicates*, se base sur un processus de détection en deux étapes : (1) l'identification des candidats et (2) la mesure de la similarité. L'identification des candidats se fait à l'aide d'un algorithme de la famille de la recherche des plus proches voisins. Cet algorithme, *Locality Sensitive Hashing*, génère un *hash* en fonction des caractéristiques des scans ultrasonores. Il faut tout d'abord créer une base de données de *hash* de chacun des scans ultrasonores. Lorsqu'un nouveau scan ultrasonore est disponible, *Locality Sensitive Hashing* permet de retrouver rapidement les candidats qui sont potentiellement des *near duplicates* grâce au *hash* et à la base de données. Ensuite, chaque candidat est évalué, c'est-à-dire comparé avec le scan en requête, à l'aide d'une mesure de similarité afin de connaître le niveau de similarité de chaque candidat. La mesure de similarité sélectionnée est la *multi-scale structural similarity index measure*. Finalement, chaque candidat dont la mesure de similarité est supérieure à un seuil prédéterminé est considéré comme un *near duplicate*. Ce processus de détection a été testé et évalué sur un ensemble de données fournies par le partenaire industriel Ondia.

À la suite de la détection des doublons, l'identification de la soudure et des défauts est effectuée à l'aide de méthode de segmentation basée sur les plus récents avancements en *deep learning*. Le modèle de segmentation choisi est un *fully Convolutional network* qui prend en intrant un scan ultrasonore 3D et qui produit deux cartes de segmentations distinctes : une des différentes parties de la soudure et une des défauts. Les différentes parties de la soudure ont été divisées en quatre pour cette étude : matériel de base, soudure en V, soudure en double-V et réflexion incomplète de la soudure. Les défauts sont identifiés de façon binaire (défaut ou pas).

La troisième étape consiste à reconstruire la surface de la soudure et des défauts, et ce, à partir des deux cartes de segmentation produite. Cette étude fait appel à une méthode de modélisation basée sur l'apprentissage d'un *prior* local afin de reconstruire en trois dimensions la surface de la soudure et ses éventuels défauts. Cette technique de *deep learning* qui apprend à prédire une fonction de distance signée est très robuste, étant en mesure de reconstruire autant la soudure que les défauts.

Finalement, un dernier outil permet de faire le suivi périodique des soudures. Cet outil se base en grande partie sur le même processus que pour effectuer la détection des doublons. La recherche des candidats est encore une fois effectuée à l'aide de *locality sensitive hashing*. L'intrant à ce processus est toutefois un échantillonnage aléatoire de la surface des modèles géométriques 3D. À l'aide de la distance de chanfrein et d'un seuil prédéterminé, il est possible de retrouver des soudures déjà scannées et de les comparer. Ceci permet d'effectuer un meilleur suivi périodique des soudures. Il est possible de constater la dégradation de la soudure et des défauts au fil de sa vie utile.

En somme, cette recherche a permis de développer un processus d'analyse de soudure complet, précis et automatisé. Le résultat de cette étude est l'augmentation de la qualité du processus d'analyse des soudures par l'ajout de nouveaux éléments (les quatre sous-objectifs) qui étaient indisponibles auparavant. Ces informations aident les analystes certifiés dans le processus de décision, réduisant ainsi leur fardeau et le délai entre l'acquisition des données et la décision finale. Une autre contribution est l'ajout de ce processus à un logiciel infonuagique, permettant une analyse des scans ultrasonores en quelques minutes seulement.

ABSTRACT

Degradation of engineering materials is a natural phenomenon that causes different risks to many industries, such as transportation and civil engineering. The main objective of non-destructive testing is to acquire data on engineered materials to assess their integrity and safety. In welding, the effectiveness of non-destructive testing relies on consistent data acquisition using ultrasonic equipment, and reliable analysis by an expert. Analysis of the data allows professionals to extract information about the defects (types, orientation, degree of risk) and to make a diagnosis about the weld. However, there is currently a shortage of certified analysts, and the time required to perform an analysis is increasing. In this context, analyses are prone to human error and potentially even fraud. Also, the analyses are based on raw data that does not have the precise location of the weld. This project aimed at developing, validating, and implementing artificial intelligence methods to:

- a) Ensure the integrity and authenticity of the ultrasound data submitted for the first time
- b) Identify the different parts of the weld
- c) Reconstruct the 3D geometric surface of these parts
- d) Ensure a better follow-up of the periodic analyses of a given weld

Firstly, a duplicate detection process was developed. The process was adapted according to the two types of existing duplicates (exact duplicate and near duplicate). The first part of this process used hash functions to generate a digital fingerprint from the ultrasound data (3D S-Scan). This allowed to detect exact duplicates quickly and efficiently. The second part, which focused on the detection of near duplicates, was based on a two-step detection process: (1) candidate identification and (2) similarity measurement. Candidate identification was done using an algorithm from the nearest neighbor search family. This algorithm, called Locality Sensitive Hashing, generated a hash based on the characteristics of the ultrasound scans. First, a database with all ultrasound scan hashes was created. When a new ultrasound scan was available, Locality Sensitive Hashing could quickly find the candidates that were potentially near duplicates, thanks to the hash and the database. Then, each candidate was compared with the query scan using a similarity measure called multi-scale structural similarity index measure, to find the level of similarity of each one. Finally, candidates whose similarity measure was higher than a predetermined threshold were considered as near duplicates. This detection process was tested and evaluated on a dataset provided by the industrial partner Ondia.

Following the detection of duplicates, the identification of the weld and its defects was performed using a segmentation method based on the latest advances in deep learning. The segmentation model chosen was a fully convolutional network that took as input a 3D ultrasound scan and produced two distinct segmentation maps: one of the different parts of the weld and one of the defects. The different parts of the weld were divided into four for this study: base material, single-V weld, double-

V weld, and the incomplete reflection of the weld. The defects were identified in a binary way (presence or absence of defect).

The third step was to reconstruct the surface of the weld and its defects by using the two segmentation maps produced. This study used a modeling method based on the learning of a local prior to reconstruct the surface of the weld and its possible defects in three dimensions. This deep learning technique, which learns to predict a signed distance function, is very robust, being able to reconstruct both weld and defects.

Finally, a last tool was developed to periodically monitor the welds. This tool was largely based on the same process used for duplicate detection: Locality Sensitive Hashing. The inputs, however, were random samples of the surface of the 3D geometric models. Using the chamfer distance and a predetermined threshold, it was possible to find previously scanned welds and compare them with each other. This allows for a better periodic monitoring of the welds because it is possible to see their degradation and the emergence or evolution of defects over time.

In summary, this research has resulted in the development of a comprehensive, accurate and automated weld analysis process. One of the main contributions of this study was an increase in the quality of the weld analysis process by adding new features and processes that were previously unavailable. This information helps certified analysts in the decision process, reducing their burden and the time between data acquisition and the final assessment. Another contribution was the addition of this process to a cloud-based software, which allows analysis of ultrasound scans in just a few minutes.

TABLE DES MATIÈRES

REMERCIEMENTS	IV
RÉSUMÉ	V
ABSTRACT	VII
TABLE DES MATIÈRES	IX
LISTE DES TABLEAUX	XIII
LISTE DES FIGURES	XIV
LISTE DES ÉQUATIONS.....	XVII
LISTE DES ABRÉVIATIONS	XIX
TRADUCTION DES TERMES ANGLAIS	XX
DÉFINITIONS	XXI
1 INTRODUCTION	1
1.1 Contexte global de la thèse	1
1.2 Lacunes ciblées par cette thèse.....	2
1.3 Objectifs de la thèse	5
2 REVUE DE LITTÉRATURE	7
2.1 Détection des doublons.....	7
2.2 Identification de la soudure	9
2.2.1 Classification des défauts de soudure	10
2.2.1.1 Méthodes avec débruitage	10
2.2.1.2 Méthodes sans débruitage	12
2.2.2 Segmentation à l'aide de l'apprentissage profond.....	17
2.2.2.1 Segmentation à partir d'algorithmes supervisés	17

2.2.2.2	Segmentation à partir d'algorithmes non supervisés	24
2.3	<i>Modélisation géométrique en trois dimensions</i>	31
2.3.1	Principes de base.....	31
2.3.2	Modélisation en trois dimensions à l'aide de l'intelligence artificielle.....	35
2.4	<i>Suivi périodique des soudures</i>	41
2.5	<i>Positionnement de la recherche</i>	44
3	MÉTHODOLOGIE DE LA RECHERCHE	46
3.1	<i>Description des données</i>	46
3.2	<i>Détection des doublons</i>	49
3.3	<i>Identification de la soudure</i>	54
3.3.1	Apprentissage multitâche	54
3.3.2	Choix du modèle de segmentation.....	55
3.3.3	Sélection du modèle optimal et de ses hyperparamètres.....	57
3.3.3.1	Résultats de Hyperband.....	57
3.3.3.2	Augmentation des données	63
3.4	<i>Modélisation géométrique en trois dimensions</i>	65
3.5	<i>Comparaison périodique des soudures</i>	68
4	RESULTATS ET DISCUSSION	72
4.1	<i>Détection des doublons</i>	72
4.1.1	Détection des ED	72
4.1.2	Détection des <i>ND</i>	73
4.1.2.1	Capacité de détection	74
4.1.2.2	Précision de LSH	75
4.1.2.3	Performances sur des scans inédits	76
4.1.2.4	Robustesse au bruit	78
4.1.2.5	Résumé des quatre expériences	78
4.2	<i>Identification de la soudure</i>	79

4.2.1	Entrainement du modèle de segmentation	79
4.2.2	Segmentation des quatre principales parties de la soudure	80
4.2.2.1	Analyse des résultats	80
4.2.2.2	Effets de la ZAT	84
4.2.3	Segmentation des défauts.....	87
4.2.4	Segmentation intégrée de la soudure et des défauts	90
4.2.5	Comparaison avec un auto-encodeur convolutif	93
4.3	<i>Modélisation géométrique en trois dimensions</i>	95
4.3.1	Résultats de <i>Hyperband</i> appliqués à <i>P2S</i>	95
4.3.2	Entrainement du modèle <i>P2S</i>	99
4.3.3	Comparaison des techniques de reconstruction de surface	100
4.4	<i>Suivi périodique d'une soudure</i>	103
4.4.1	Étude de cas	103
4.4.2	Valeur du seuil de similarité	105
4.4.3	Configuration de <i>LSH</i>	106
4.4.4	Maximisation des performances	106
4.4.5	Résumé des quatre expériences	107
4.5	<i>Implantation</i>	107
4.5.1	Exemple d'utilisation	107
4.5.2	Implémentation dans un logiciel infonuagique	110
5	CONCLUSION	113
5.1	<i>Retour sur les objectifs et contributions scientifiques de la thèse</i>	113
5.2	<i>Limitations et perspectives</i>	115
	LISTE DE RÉFÉRENCES	118
6	ANNEXE 1 : PARAMÈTRES IMPORTANTS DES RÉSEAUX DE NEURONES	130
6.1	<i>Fonction d'activation</i>	130
6.2	<i>Méthode de régularisation</i>	132

6.3	<i>Taux d'apprentissage</i>	135
6.4	<i>Fonction d'optimisation</i>	136
6.5	<i>Fonction de perte</i>	139

LISTE DES TABLEAUX

TABLEAU 1.1 : MESURES DE PERFORMANCE LES PLUS COMMUNES	XXIII
TABLEAU 1.1 : PRINCIPALES CONSEQUENCES DES LACUNES ACTUELLES	5
TABLEAU 2.1 : RESUME DES ETUDES DE CLASSIFICATION EN END UTILISANT L'IA	16
TABLEAU 2.2 : RESUME DES ETUDES DE SEGMENTATION SUPERVISEES.....	24
TABLEAU 2.3 : RESUME DES ETUDES DE SEGMENTATION NON SUPERVISEES.....	30
TABLEAU 2.4 : RESUME DES ETUDES SUR LA MODELISATION 3D BOUT-EN-BOUT	41
TABLEAU 3.1 : COMPARAISON QUANTITATIVE DES ANNOTATIONS PAR LES EXPERTS	49
TABLEAU 3.2 : MODELES DE SEGMENTATION 3D SELECTIONNES	56
TABLEAU 3.3 : RESULTATS DE <i>HYPERBAND</i> POUR LA SELECTION DES HYPERPARAMÈTRES OPTIMAUX.....	58
TABLEAU 3.4 : COMPARAISON DES FONCTIONS DE PERTE	62
TABLEAU 4.1 : EXEMPLES DE HASH ASSOCIES AUX FICHIERS SCANNES	73
TABLEAU 4.2 : MESURE DE SIMILARITÉ DES <i>ANN</i> POUR LA SOUDURE #282 270°	76
TABLEAU 4.3 : RESUME DES EXPERIENCES	79
TABLEAU 4.4 : TABLEAU DES SCORES F1 POUR CHAQUE CLASSE DE LA SOUDURE	81
TABLEAU 4.5 : COMPARAISON QUANTITATIVE DES RESULTATS DE SEGMENTATION ENTRE UN MODELE A APPRENTISSAGE SUPERVISE ET NON SUPERVISE	93
TABLEAU 4.6 : RESULTATS DE <i>HYPERBAND</i> POUR LA SELECTION DE HYPERPARAMÈTRES OPTIMAUX DE <i>P2S</i>	96
TABLEAU 4.7 : COMPARAISON QUANTITATIVE DE L'ERREUR DE RECONSTRUCTION	100
TABLEAU 4.8 : COMPARAISON QUANTITATIVE DE L'ERREUR DE RECONSTRUCTION DE <i>P2S</i>	101
TABLEAU 4.9 : OPTIMISATION DES HYPERPARAMÈTRES DE <i>P2S</i>	103
TABLEAU 4.10 : ÉTUDE DE CAS DE LA SOUDURE #37 270°	104
TABLEAU 4.11 : LISTE DES CONFIGURATIONS TESTEES ET LEURS RESULTATS	106
TABLEAU 4.12 : RESUME DES EXPERIENCES	107

LISTE DES FIGURES

FIGURE 1.1 : SCHEMA DES ETAPES DE REALISATION DES 4 OBJECTIFS SPECIFIQUES	6
FIGURE 2.1 : L'ARCHITECTURE DE <i>U-NET</i>	18
FIGURE 2.2 : REPRESENTATION DE <i>V-NET</i>	20
FIGURE 2.3 : ARCHITECTURE D'UN AUTO-ENCODEUR.....	26
FIGURE 2.4 : REPRÉSENTATION VISUELLE DE <i>F-ANoGAN</i>	29
FIGURE 2.5 : RECONSTRUCTION PAR <i>MARCHING CUBES</i> D'UNE TÊTE À PARTIR DE 150 IMAGES DE RÉSONANCE MAGNÉTIQUE	33
FIGURE 2.6 : MAILLAGE GÉNÉRÉ À PARTIR D'UNE SURFACE IMPLICITE.....	34
FIGURE 2.7 : RECONSTRUCTION PAR DIFFERENTES TECHNIQUES, DONT <i>SPR</i> , D'UN MODELE DE RAPTOR.....	35
FIGURE 2.8 : ARCHITECTURE DE <i>DEEP MARCHING CUBES</i>	36
FIGURE 2.9 : RESULTATS VISUELS DE <i>DEEP MARCHING CUBES (DMC)</i> A PARTIR D'UN NUAGE DE POINTS.....	37
FIGURE 2.10 : COMPARAISON DE <i>POINT2MESH</i> AVEC LA RECONSTRUCTION PAR POISSON.....	39
FIGURE 2.11 : EXEMPLES DE SURFACES RECONSTRUITES PAR <i>P2S</i> A PARTIR D'UN NP BRUITE.....	39
FIGURE 2.12 : PROCESSUS COMPLET DE <i>P2S</i>	40
FIGURE 3.1 : UN SCAN REPRESENTÉ SOUS DEUX FORMATS.	47
FIGURE 3.2 : EXEMPLES DE DONNÉES BRUTES <i>S-SCAN</i>	48
FIGURE 3.3 : EXEMPLES DE <i>ED</i> : <i>S-SCAN</i> (A ET B) ET <i>C-SCAN</i> (C ET D).....	50
FIGURE 3.4 : EXEMPLES DE <i>ND</i> : <i>S-SCAN</i> (A ET B) ET <i>C-SCAN</i> (C ET D).	50
FIGURE 3.5 : PROCESSUS DE DETECTION DES DOUBLONS	51
FIGURE 3.6 : PROCESSUS DE DETECTION DES <i>ED</i>	53
FIGURE 3.7 : UN PROCESSUS STANDARD DE DETECTION DES DOUBLONS	53
FIGURE 3.8 : EXEMPLE DE <i>CNN</i> MULTITACHES	55
FIGURE 3.9 : ILLUSTRATION DE <i>HYPERBAND</i> AVEC UN BUDGET DE 8.....	57
FIGURE 3.10 : ILLUSTRATION DE L'EXECUTION DE <i>HYPERBAND</i> AVEC R=81 ET H=3	58
FIGURE 3.11 : MODÈLE DE SEGMENTATION MULTI-OBJECTIF	59
FIGURE 3.12 : EXEMPLES DE TECHNIQUE D'AUGMENTATION DES DONNÉES UTILISÉES.....	64

FIGURE 3.13 : EXEMPLES DE TECHNIQUES D'AUGMENTATION DES DONNEES UTILISEES DE LA FAMILLE DE L'INJECTION DE BRUIT ...	65
FIGURE 3.14 : PROCESSUS DE RECONSTRUCTION DE LA SURFACE DE LA SOUDURE ET DES DEFAUTS	66
FIGURE 3.15 : EXEMPLES D'UNE MEME SOUDURE ISSUE DE DEUX SCANS DIFFERENTS	68
FIGURE 3.16 : PROCESSUS DE RECONNAISSANCE DES MODELES GEOMETRIQUES 3D	69
FIGURE 4.1 : PROCESSUS DE DETECTION DES <i>ED</i>	72
FIGURE 4.2 : EXEMPLE DE <i>ED</i> ET LEUR <i>HASH</i> RESPECTIF	73
FIGURE 4.3 : PROCESSUS DE DETECTION DES DOUBLONS UTILISE	74
FIGURE 4.4 : PERFORMANCE DE DÉTECTION DES <i>ND</i> À DIFFÉRENTES VALEURS DE τ	75
FIGURE 4.5 : PERFORMANCE DE DÉTECTION DE CHACUNE DES CONFIGURATIONS TESTÉES	76
FIGURE 4.6 : ÉTUDE DE CAS DE DÉTECTION DE <i>ND</i> D'UN FICHIER TEST.	77
FIGURE 4.7 : COMPARAISON DE DEUX S-SCANS MAXIMALES SIMILAIRES (#280 270° ET #282 270°)	77
FIGURE 4.8 : PRECISION DE DETECTION DES <i>ND</i> A DIFFERENTS NIVEAUX DE BRUIT GAUSSIEN.....	78
FIGURE 4.9 : ÉVOLUTION DE LA VALEUR DE LA PERTE ET DE LA PRÉCISION AU FIL DES <i>EPOCHS</i>	79
FIGURE 4.10 : MATRICE DE CONFUSION DE LA SEGMENTATION DE CHAQUE VOXEL DE LA SOUDURE	80
FIGURE 4.11 : EXEMPLES DES INTRANTS, PREDICTIONS ET <i>GT</i> DU MODELE DE SEGMENTATION.....	83
FIGURE 4.12 : DISTRIBUTION DES <i>DSC</i> DE L'ECHANTILLON DE VALIDATION POUR LA SOUDURE	83
FIGURE 4.13 : PIRE PREDICTION DE L'ECHANTILLON DE VALIDATION.....	84
FIGURE 4.14 : VISUALISATION DE CHAQUE CLASSE DE LA SOUDURE POUR DEUX PREDICTIONS	86
FIGURE 4.15 : ILLUSTRATION EN 3D DE LA ZAT POTENTIELLE DE DEUX PREDICTIONS.....	87
FIGURE 4.16 : MATRICE DE CONFUSION DE LA SEGMENTATION DE CHAQUE VOXEL DES DÉFAUTS	88
FIGURE 4.17 : EXEMPLES DES INTRANTS, PREDICTIONS ET ANNOTATIONS DU MODELE DE SEGMENTATION	89
FIGURE 4.18 : DISTRIBUTION DES <i>DSC</i> DE L'ECHANTILLON DE VALIDATION POUR LES DEFAUTS.....	89
FIGURE 4.19 : VISUALISATION POUR DEUX PREDICTIONS DES CLASSES SANS DEFAUT ET DEFAUT.....	90
FIGURE 4.20 : EXEMPLE 3D DES INTRANTS, PREDICTIONS ET EXTRANTS DU MODELE DE SEGMENTATION	91
FIGURE 4.21 : EXEMPLE 3D DES INTRANTS, PREDICTIONS ET EXTRANTS DU MODELE DE SEGMENTATION	91
FIGURE 4.22 : EXEMPLES DES INTRANTS, PREDICTIONS ET ANNOTATIONS DU MODELE DE SEGMENTATION SUPERPOSES	92

FIGURE 4.23 : MATRICE DE CONFUSION DE LA SEGMENTATION PAR UN AUTO-ENCODEUR CONVOLUTIF	94
FIGURE 4.24 : COMPARAISON DES RESULTATS DE SEGMENTATION D'UN AUTO-ENCODEUR CONVOLUTIF AUX ANNOTATIONS.....	95
FIGURE 4.26 : RESULTATS DE <i>HYPERBAND</i> REPRESENTES SOUS FORME DE NUAGE DE POINTS	97
FIGURE 4.27 : ÉVOLUTION DE LA VALEUR DE LA PERTE (ENSEMBLE D'ENTRAÎNEMENT ET DE VALIDATION) AU FIL DES <i>EPOCHS</i>	99
FIGURE 4.28 : MODELES GEOMETRIQUES RECONSTRUITS PAR <i>SPR</i> ET <i>P2S</i>	101
FIGURE 4.29 : DISTRIBUTION DES <i>DSC</i> DE L'ECHANTILLON DE VALIDATION POUR LES DEFAULTS.....	102
FIGURE 4.30 : ÉTUDE DE CAS DE DETECTION DES MODELES GEOMETRIQUES : LE MODELE GEOMETRIQUE ETUDIE ET LES PLUS PROCHES VOISINS	105
FIGURE 4.31 : RÉSUMÉ VISUEL DE L'EXPÉRIENCE SUR LE SEUIL	106
FIGURE 4.32 : SYNTHÈSE DU PROCESSUS DEVELOPPE DANS CETTE ETUDE	109
FIGURE 4.33 : INTERFACE DU LOGICIEL ONDIA : INTÉGRATION DU PROCESSUS DE DÉTECTION DES DOUBLONS	111
FIGURE 4.34 : RAPPORT DE DUPLICATION POUR UN SCAN ULTRASONORE CHOISI ALEATOIREMENT	111
FIGURE 6.1 : EXEMPLES VISUELS DE QUELQUES FONCTIONS D'ACTIVATION	130
FIGURE 6.2 : DIFFÉRENTES FONCTIONS D'ACTIVATION ET LEUR PERFORMANCE SUR CIFAR10.....	131
FIGURE 6.3 : <i>GELU</i> , <i>RELU</i> ET <i>ELU</i>	131
FIGURE 6.4 : VALEURS DE PERTE AVEC OU SANS LA NORMALISATION PAR <i>BATCH</i>	133
FIGURE 6.5 : ILLUSTRATION DES DIFFÉRENTS TYPES DE NORMALISATION POUR UN EXEMPLE EN 2D	134
FIGURE 6.6 : TAUX D'APPRENTISSAGE DE <i>CYCLICLR</i> : <i>TRIANGULAR</i> ET <i>TRIANGULAR2</i>	136
FIGURE 6.7 : TYPE DE FONCTIONS DE PERTE POUR LA SEGMENTATION	139
FIGURE 6.8 : VISUALISATION DE <i>IoU</i>	141

LISTE DES ÉQUATIONS

(3.1)	$\text{SCORE F1} = \frac{2 \cdot \text{PRECISION} \cdot \text{RAPPEL}}{\text{PRECISION} + \text{RAPPEL}}$	61
(3.2)	$\text{SCORE F1} = \frac{\text{VP}}{\text{VP} + \frac{1}{2}(\text{FN} + \text{FP})}$	62
(3.3)	$D_{\text{CH}}(A, B) := \frac{1}{ A } \sum_{P_i \in A} \min_{P_j \in B} \ P_i - P_j\ _2^2 + \frac{1}{ B } \sum_{P_i \in B} \min_{P_j \in A} \ P_i - P_j\ _2^2$	66
(3.4)	$\text{WD}(P, Q) = \frac{\sum_{l=1}^M \sum_{j=1}^N F_{l,j} D_{l,j}}{\sum_{l=1}^M \sum_{j=1}^N F_{l,j}}$	70
(3.5)	$\text{DH}(P, Q) = \max\{\sup_{P \in \mathcal{P}} D(P, Q), \sup_{Q \in \mathcal{Q}} D(P, Q)\}$	70
(4.1)	$\text{PRECISION} = \frac{\text{VP}}{\text{VP} + \text{FP}} = \frac{0,17}{0,17 + 0,10} = 63,2$	88
(4.2)	$\text{RAPPEL} = \frac{\text{VP}}{\text{VP} + \text{FN}} = \frac{0,17}{0,17 + 0,11} = 61,2 \%$	88
(4.3)	$\text{SCORE F1} = \frac{2 \cdot \text{PRECISION} \cdot \text{RAPPEL}}{\text{PRECISION} + \text{RAPPEL}} = \frac{2 \cdot 0,632 \cdot 0,612}{0,632 + 0,612} = 62,2 \%$	88
(4.4)	$\text{PRECISION} = \frac{\text{VP}}{\text{VP} + \text{FP}} = \frac{28}{28 + 2} = 93,3 \%$	100
(4.5)	$\text{RAPPEL} = \frac{\text{VP}}{\text{VP} + \text{FN}} = \frac{28}{28} = 100 \%$	100
(4.6)	$\text{SCORE F1} = \frac{2 \cdot \text{PRECISION} \cdot \text{RAPPEL}}{\text{PRECISION} + \text{RAPPEL}} = \frac{2 \cdot 0,933 \cdot 1}{0,933 + 1} = 96,5 \%$	100
(6.1)	$\text{BN}(x) = \Gamma \frac{x - \text{MB}}{\Sigma_B} + B$	132
(6.2)	$R = \frac{\Sigma_B}{\Sigma} \quad D = \frac{\text{MB} - M}{\Sigma}, \text{BR}(x) = \frac{x - \text{MB}}{\Sigma_B} R + D$	133
(6.3)	$\Lambda \sum_l W_l^2$	135
(6.4)	$W_{T+1} = W_T - A \left(\frac{\partial \text{ERREUR}}{\partial W_T} \right)$	136
(6.5)	$\theta_{T+1} = \theta_T - H_T \nabla L(\theta_T)$	137
(6.6)	$\gamma_0 = 0, \quad \gamma_{T+1} = \gamma_T + \nabla L(\theta_T), \quad \theta_{T+1} = \theta_T - H_T \gamma_{T+1}$	137
(6.7)	$G_T = \frac{1}{N} \sum_{i=1}^N \nabla_{\theta} L(x^{(i)}, y^{(i)}, \theta_T), \quad G_T = \sum_{T=1}^T G_T G_T^T, \quad \theta_{T+1} = \theta_T - \frac{H}{\sqrt{\text{EI} + \text{DIAG}(G_T)}} \cdot G_T$	137
(6.8)	$V_0 = 1, \quad M_0 = 0, \quad V_{T+1} = P V_T + (1 - P) \nabla L(\theta_T)^2, \quad M_{T+1} = \Gamma M_T + \frac{H_T}{\sqrt{\gamma_{T+1} + \epsilon}} \nabla L(\theta_T), \quad \theta_{T+1} = \theta_T - M_{T+1}$	137

- (6.9) $M_0 = 0, V_0 = 0, M_{T+1} = B_1 M_T + (1 - B_1) \nabla L(\theta_T), V_{T+1} = B_2 V_T + (1 - B_2) \nabla L(\theta_T)^2, B_{T+1} = \frac{\sqrt{1 - B_1^{T+1}}}{1 - B_1^{T+1}}, \theta_{T+1} = \theta_T - A_T \frac{M_{T+1}}{\sqrt{V_{T+1} + \epsilon}} B_{T+1} \dots\dots\dots 138$
- (6.10) $M_0 = 0, V_0 = 0, M_{T+1} = B_1 M_T + (1 - B_1) \nabla L(\theta_T), V_{T+1} = B_2 V_T + (1 - B_2) \nabla L(\theta_T)^2, B_{T+1} = \frac{\sqrt{1 - B_1^{T+1}}}{1 - B_1^{T+1}}, \theta_{T+1} = \theta_T - A_T \frac{B_1 M_{T+1} + (1 - B_1) \nabla L(\theta_T)}{\sqrt{V_{T+1} + \epsilon}} B_{T+1} \dots\dots\dots 138$
- (6.11) $FL = -A_T (1 - P_T)^\Gamma \log(P_T) \dots\dots\dots 140$
- (6.12) $DICE(Y, \hat{P}) = 1 - \frac{2Y\hat{P} + 1}{Y + \hat{P} + 1} \dots\dots\dots 140$
- (6.13) $SDL(Y, \hat{P}) = 1 - \frac{2Y\hat{P}}{Y^2 + \hat{P}^2} \dots\dots\dots 140$
- (6.14) $GDL(Y, \hat{P}) = 1 - 2 \frac{\sum_{L=1}^2 W_L \sum_N Y_{LN} \hat{P}_{LN}}{\sum_{L=1}^2 W_L \sum_N Y_{LN} + \hat{P}_{LN}} \dots\dots\dots 140$
- (6.15) $F_B = (1 + B^2) \frac{PRECISION \times RAPPEL}{B^2 \times PRECISION + RAPPEL} \dots\dots\dots 141$
- (6.16) $F_B = \frac{(1 + B^2) \sum_{i=1}^N P_i G_i}{(1 + B^2) \sum_{i=1}^N P_i G_i + B^2 \sum_{i=1}^N (1 - P_i) G_i + \sum_{i=1}^N P_i (1 - G_i)} \dots\dots\dots 141$
- (6.17) $IoU(Y, \hat{P}) = 1 - \frac{\sum_{C=1}^C \sum_{N=1}^N Y_C \hat{P}_C}{\sum_{C=1}^C \sum_{N=1}^N Y_C + \hat{P}_C - Y_C \hat{P}_C} \dots\dots\dots 141$
- (6.18) $SS(Y, \hat{P}) = \Lambda \frac{\sum_{N=1}^N (Y_N - \hat{P}_N)^2 R_N}{\sum_{N=1}^N Y_N + \epsilon} + (1 - \Lambda) \frac{\sum_{N=1}^N (Y - \hat{P}_N)^2 (1 - Y_N)}{\sum_{N=1}^N (1 - Y_N) + \epsilon} \dots\dots\dots 142$
- (6.19) $TL(Y, \hat{P}) = 1 - \frac{1 + Y\hat{P}}{1 + Y\hat{P} + B(1 - Y)\hat{P} + (1 - B)Y(1 - \hat{P})} \dots\dots\dots 142$
- (6.20) $FTL = (L_{TVERSKEY})^{\frac{1}{\Gamma}} \dots\dots\dots 142$
- (6.21) $LC - D = \log(\cosh(DICELOSS)) \dots\dots\dots 142$
- (6.22) $DH(X, Y) = \max_{X_\epsilon} \min_{Y_\epsilon} \|X - Y\|_2 \dots\dots\dots 143$

LISTE DES ABRÉVIATIONS

Abréviation	Signification intégrale	Traduction
Adam	Adaptive Gradient Methods	
ANN	Approximate Nearest Neighbors	Plus proches voisins approximatifs
CNN	Convolutional Neural Network	Réseau de neurones convolutif
DSC	Dice Similarity Coefficient	Coefficient de Dice
ED	Exact-Duplicate	Doublon exact
END	Non-Destructive Testing	Essai Non-Destructif
FCN	Fully Convolutional Network	Réseau de neurones entièrement convolutif
GPU	Graphics Processing Unit	Unité de traitement graphique
GT	Ground-truth	Valeur cible
IA	Artificial Intelligence	Intelligence Artificielle
IoU	Intersection over Union	Intersection sur l'union
LR	Learning rate	Taux d'apprentissage
MSSSIM	Multi Scale Structural Similarity Index Measure	
ND	Near-Duplicate	Quasi-Doublon
P2S	Points to Surface (Points2surf)	
PAUT	Phased-Array Ultrasonic Testing	Transducteur multi-éléments
PReLU	Parametric Rectified Linear Unit	
RDN	Artificial Neural Network	Réseau de Neurones artificiel
ReLU	Rectified Linear Unit	
SGD	Stochastic Gradient Descent	Algorithme du gradient stochastique
SPR	Screened Poisson Reconstruction	Reconstruction par Poisson filtrée
SSIM	Structural Similarity Index Measure	
ZAT	Heat Affected Zone	Zone affectée thermiquement

TRADUCTION DES TERMES ANGLAIS

Signification intégrale	Traduction
Adversarial	Contradictoire
Batch size	Taille des lots
Binary Cross Entropy	Entropie croisée binaire
Blockchain	Chaîne de blocs
Chamfer Distance	Distance de Chanfrein
Data augmentation	Augmentation des données
Denoising Convolutional Auto-Encoder	Auto-Encodeur convolutif débruitant
Denoising Auto-Encoder	Auto-Encodeur débruitant
Divide-and-conquer	Diviser pour mieux régner
Dropout	Abandon
Feature maps	Cartes des caractéristiques
Global coarse subsample	Sous-échantillon global grossier
Hash	Valeur de hachage
Hidden layer	Couche cachée
Local Binary Pattern	Motif binaire local
Lookup table	Table de consultation
Machine Learning	Apprentissage automatique
Multi-Layer Perceptron	Perceptron multicouche
Neighborhood descriptors	Descripteurs de voisins
Overfitting	Surapprentissage
Patch	Carreau de surface
Pattern	Motif, forme
Pooling	Mise en commun
Prior	Précédent
Reinforcement Learning	Apprentissage par renforcement
Shrink-wrap	Rétracter
Signed Distance Function	Fonction de distance signée
Skip connection	Connexion entre l'encodeur et le décodeur
Smoothness	Finesse
Stride	Espacement
Up-convolutions	Convolution ascendante
Watertight	Étanche
Watermarking	Filigrane
Wavelet transform	Transformation en ondelettes

DÉFINITIONS

Type d'apprentissage en intelligence artificielle

Il existe deux principes d'apprentissage : l'apprentissage supervisé ou non supervisé [1]. Le choix du type d'apprentissage dépend de la taille de l'ensemble de données disponibles pour l'entraînement de l'architecture du réseau de neurones (RDN), de la variabilité des données analysées, du type de données et de l'objectif du modèle.

Apprentissage supervisé

En général, l'apprentissage supervisé requiert des centaines (souvent des milliers) d'exemples annotés [2]. Il est supposé que les données d'entraînement soient disponibles sous forme de paires (x, y) , où $x \in \mathbb{R}^m$ est un exemple d'entraînement et y est l'objectif, le résultat attendu. Les exemples annotés décrivent les résultats que le modèle d'intelligence artificielle (IA) doit produire (prédire). L'apprentissage se fait par la mise à jour de la pondération des poids entre chaque neurone en générant un signal d'erreur entre les données annotées (désirées) et les résultats du RDN. Il s'agit de la méthode d'apprentissage la plus utilisée. L'annotation de milliers de données peut-être un exercice long, complexe et couteux [3].

Apprentissage non supervisé

Entraîner un modèle d'IA sans utiliser d'exemples annotés s'appelle l'apprentissage non supervisé. Ce type d'apprentissage se base sur l'apprentissage des tendances intrinsèques aux données [4]. L'application typique est le partitionnement (*clustering*) qui consiste à partitionner les données en groupes, appelés *cluster*, en fonction des similitudes de leurs caractéristiques intrinsèques. L'apprentissage non supervisé est également utile pour effectuer de la détection d'anomalies. Le modèle apprend alors à reconnaître les données normales. Une fois que le modèle est en mesure de bien reconnaître ce qu'est une donnée normale, il est capable de détecter les anomalies. Entraîner un modèle d'IA à l'aide de l'apprentissage non supervisé permet donc d'utiliser des données qui ne sont pas annotées, ce qui représente un avantage dans bien des cas. Les architectures qui utilisent l'apprentissage non supervisé sont les Auto-Encodeurs, les *Generative Adversarial Networks*, les *Deep Belief Networks* et les *Restricted Boltzman Machines*.

Contrôle par Ultrasons

Le contrôle par ultrasons utilise un transducteur piézoélectrique pour générer des ondes ultrasonores, variant de 500 kHz à 20 MHz, qui se propagent à travers une plaque métallique [5]. Les ondes ultrasonores sont réfléchies par les défauts et retournent vers le transducteur [6]. Le transducteur

capte les ondes réfléchies et les convertit en signaux électroniques. Ces signaux, se nommant A-Scan, contiennent ainsi l'information sur le type, la taille et l'orientation des défauts. En plus des signaux de type A-Scan, il existe des signaux de type B-Scan, C-Scan et S-Scan. Le principe des signaux de type B-Scan est de balayer le transducteur suivant une direction sur la pièce. La position du transducteur et le signal de type A-Scan reçu à cette position sont adjoints permettant d'obtenir une image représentant la structure interne de la pièce suivant un plan de coupe. Pour les signaux de type C-Scan, le même principe que pour le B-Scan est utilisé, mais en réalisant plusieurs plans de coupes. On a ainsi une représentation en voxel de la structure de la pièce. Finalement, les signaux de type S-Scan (sectoriel) sont obtenus à la suite d'un balayage, mais cette fois-ci suivant un angle. Ce type de signal est obtenu généralement avec un transducteur multiélément (*phased-array ultrasonic testing : PAUT*). Le *PAUT* crée des données 3D de la structure de la surface inspectée, permettant une analyse approfondie de celle-ci. Les données récoltées par le *PAUT* sont au format (*frames, beams, samples*) où les *frames* correspondent aux tranches du scan dans le sens de l'axe *y*. Les *beams* correspondent au nombre d'éléments utilisé pour générer la vague ultrasonore. Les *samples* sont le nombre d'échantillons de données récoltées par *beam* lorsqu'une vague ultrasonore est générée. De nos jours, le contrôle par ultrasons est devenu l'une des techniques d'essai non destructif les plus populaires. Le résultat de cette popularité est la production de grandes quantités de scans ultrasonores (A-Scan, B-Scan, C-Scan, S-Scan).

Convolutional Neural Networks (CNN)

Les *CNNs* sont des RDN qui combinent des couches convolutives et des couches entièrement connectées [2]. Les premières couches se spécialisent dans l'extraction des caractéristiques, tandis que la deuxième partie du modèle est constituée des couches pour la classification. Les couches d'extraction de caractéristiques sont des couches de convolutions et de *pooling*, tandis que la classification est effectuée par une couche entièrement connectée. Les *CNNs* peuvent apprendre des *patterns* dans différentes zones des images en se concentrant sur des caractéristiques de bas niveau qui sont assemblées en caractéristiques de haut niveau. Il permet la classification sans extraction préalable des caractéristiques. Les *CNNs* ont toutefois besoin d'un large ensemble de données pour l'apprentissage, en plus de l'utilisation intensive des techniques d'augmentation des données.

Augmentation de données

Le problème avec l'entraînement des modèles d'apprentissage profond (*deep learning*) est la rareté des données représentatives disponibles, et ceci est d'autant plus vrai pour les données ultrasonores [7]. Les données ultrasonores réelles sont utilisées pour créer des données ultrasonores virtuelles. L'augmentation des données est nécessaire par le besoin important de données d'entraînement des

CNNs et des RDN profonds. En fait, sans une quantité importante de données couplée à l'augmentation des données, les résultats de ces réseaux sont décevants. L'augmentation des données consiste en la modification des données d'entraînement à des fins d'augmenter la performance des modèles. Ces techniques modifient légèrement les intrants de façon à respecter les modifications naturellement présentes dans les données. Les techniques les plus couramment utilisées sont la rotation, la translation, l'ajout de bruit et le redimensionnement.

Hyperparamètres

En IA, un hyperparamètre est un paramètre dont la valeur est utilisée pour contrôler le processus d'apprentissage. Les valeurs des autres paramètres (par exemple les poids) sont obtenues par l'entraînement (apprentissage). Les hyperparamètres peuvent être classifiés comme étant des hyperparamètres de modèle, qui touchent à la sélection du modèle, ou des hyperparamètres d'algorithmes, qui en principe n'exercent aucune influence sur la performance du modèle, mais affectent la rapidité et la qualité du processus d'apprentissage. Un exemple de hyperparamètre de modèle est la taille du modèle, le nombre de couches, le choix de la normalisation, les fonctions d'activation utilisées, la fonction de perte utilisée ou la valeur de *dropout*. Des exemples de hyperparamètre d'algorithme sont le *learning rate*, la taille des *batches* et la fonction d'optimisation utilisée.

Mesures de performance

Étant donné la variété des tâches et des architectures rencontrées en IA, un tableau résumant les différentes mesures de performance utilisées est présenté. Le Tableau 1.1 décrit les mesures les plus fréquentes. Il présente les abréviations utilisées, détaille les mesures de performance et leurs domaines d'application.

Tableau 1.1 : Mesures de performance les plus communes

Abréviation	Mesure de performance	Domaine d'application
<i>Accuracy</i>	$\frac{VP + VN}{VP + FP + VN + FN}$	Classification
<i>AP</i>	L'aire sous la courbe Précision-Rappel	Détection
<i>AUC</i>	L'aire sous la courbe <i>ROC (Receiver Operating characteristic)</i>	Segmentation, Détection
<i>DSC</i>	Dice Similarity Coefficient = $2 \cdot \frac{VP}{2VP + FP + FN}$	Détection, Segmentation

<i>IoU</i>	Intersection over Union = $\frac{\text{Aire de chevauchement}}{\text{Aire de l'union}}$	Détection
<i>MAP</i>	Moyenne de l'AP calculée pour toutes les classes	Détection
Précision	$\frac{VP}{VP + FP}$	Classification, Segmentation
Rappel	$\frac{VP}{VP + FN}$	Classification, Segmentation
Score F1	$2 \cdot \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$	Segmentation

Convention sur la terminologie

Il est important de noter que le mot *défaut* est réservé aux analystes certifiés qui l'utilisent lorsque leur analyse confirme la présence d'un défaut dans un scan ultrasonore [5]. Lorsqu'un algorithme ou une personne non-qualifiée identifie un *défaut* dans un scan ultrasonore, le mot *indication* doit être utilisé. Il s'agit alors de preuves contenues dans un examen non-destructif. Le mot *flaw* est également utilisé pour qualifier une imperfection ou une discontinuité qui peut être détectée par des essais non destructifs et qui n'est pas nécessairement rejetable. Finalement, un *défaut* correspond à un ou plusieurs *flaws* dont la taille, la forme, l'orientation, l'emplacement ou les propriétés ne répondent pas aux critères d'acceptation spécifiés et sont rejetables.

Il faut préciser que dans la littérature, le mot *defect* est majoritairement utilisé. *Flaw* est également utilisé plus occasionnellement. Le mot *indication* est pratiquement absent de la littérature. Dans cette étude, le mot *défaut* est utilisé pour simplifier la compréhension des lecteurs.

1 INTRODUCTION

1.1 Contexte global de la thèse

La dégradation des matériaux d'ingénierie est un phénomène naturel qui présente néanmoins d'importants risques environnementaux, économiques et de sécurité [8]. La société moderne dépend de l'intégrité continue de ces matériaux. De nombreux domaines, incluant le transport, l'énergie, le stockage de matériau liquide et gazeux, les systèmes sanitaires et les communications reposent sur un suivi soutenu de la dégradation des matériaux pour en assurer l'utilisation efficace et sécuritaire. Les essais non destructifs (ou contrôle non destructif) représentent une étape critique de l'évaluation des matériaux pour en assurer l'intégrité durant leur service. En soudage, les Essai Non-Destructif (END) reposent sur une acquisition cohérente de données sur la soudure réalisée, à la suite de laquelle elles doivent être soumises à une analyse fiable. L'analyse des données permet, entre autres, de valider la qualité de la soudure et de vérifier s'il y a la présence de défauts [5]. Plusieurs techniques de END ont pour objectif d'analyser des données recueillies à même les matériaux d'ingénierie pour en évaluer la qualité [9]. Parmi ces techniques, le contrôle par ultrasons se distingue par ses nombreux avantages tels que son accessibilité, sa portabilité, sa sécurité, son rapport coût-efficacité et son rapport signal-bruit élevé. Le contrôle par ultrasons est l'une des méthodes d'END les plus efficaces pour faire l'évaluation de la qualité et la détection des défauts dans les matériaux d'ingénierie [10].

La popularité actuelle de l'IA est sans précédent [11]. Pendant des décennies, l'intérêt pour l'IA était uniquement académique, car les applications étaient limitées et les performances étaient au mieux modestes [2]. Toutefois, l'augmentation récente de la puissance de calcul, due aux progrès du matériel informatique, comme l'unité de traitement graphique (*GPU*), et au développement de nouvelles architectures d'IA, a contribué à faire monter en flèche les performances de l'IA dans plusieurs domaines [12]. De nombreuses industries ont ajouté des outils basés sur l'IA à leurs processus de décision et à leurs activités [13]. L'apprentissage profond révolutionne le fonctionnement de nombreuses industries en permettant le développement de méthodes puissantes capable d'interpréter de grandes quantités de données automatiquement et rapidement [1]. Le domaine des END est l'un des domaines où l'apprentissage profond a un impact notable. Il faut savoir que la détérioration des matériaux est souvent multifactorielle, ce qui fait qu'elle est difficile à comprendre et à modéliser en raison du grand nombre de variables inconnues qui influencent le degré de dégradation des matériaux.

De nombreuses études se sont concentrées à développer des outils d'analyse des signaux ultrasonores pour classer automatiquement les défauts en utilisant l'IA [14, 15]. Ces études ont pour objectif de classer les différents signaux en fonction du type de défaut rencontré [16]. Ces études

développent des techniques d'apprentissage automatique (*machine learning*) pouvant classer les défauts jusqu'à 95 % des cas [17]. Afin d'améliorer la précision des outils d'analyses, certaines recherches utilisent des RDN conventionnels et des modèles profonds [18-21]. Ceci permet de diminuer l'erreur de classification à moins de deux pour cent avec des données sans bruit, ce qui se rapproche notamment du niveau de performance d'un analyste certifié [13, 22].

1.2 Lacunes ciblées par cette thèse

L'analyse des données ultrasonores pour faire le diagnostic des défauts de soudure est une tâche fastidieuse, sujette au biais de l'analyste. L'interprétation des données repose sur le jugement subjectif d'analystes certifiés, ce qui rend les conclusions vulnérables aux erreurs humaines. En plus, la quantité de données produite actuellement par les END dépasse la capacité du personnel qualifié disponible pour en faire l'analyse [23, 24]. En effet, il y a actuellement, au Canada et mondialement, une pénurie d'analystes certifiés de soudure (inspecteur en soudage certifié [Certification d'inspecteur visuel du Bureau Canadien de Soudage [BCS] CSA W178.2 niveau 2 et certification ONGC en Ultrasons niveau 2]), menant à divers problèmes tels que des délais dans les audits d'END, la falsification des données, et les erreurs de manipulation [25, 26]. En raison du vieillissement des infrastructures, des catastrophes naturelles, des accidents industriels (par exemple les explosions d'un entrepôt de produit chimique à Tianjin en 2015 faisant 173 morts) et des investissements croissants dans de nouveaux projets d'infrastructures, la demande pour de nouveaux analystes répondant aux exigences de l'industrie des END s'est accrue progressivement depuis 2013. Toutefois, le succès des END dépend en grande partie sur la capacité d'analystes certifiés à analyser les données ultrasonores.

Dans le secteur des END, l'IA est au cœur de plusieurs recherches depuis des décennies [27]. Alors que les premières applications étaient loin d'être au niveau d'un analyste certifié, les récentes applications utilisant l'apprentissage profond sont maintenant plus performantes que les analystes certifiés pour classer les défauts de soudure dans les scans ultrasonores [13, 28]. L'objectif de ces recherches est généralement de développer un RDN capable de soutenir des décisions critiques, comme l'évaluation des défauts de soudage à partir de données issues de END [29, 30]. Le problème de pénurie d'analystes est d'autant plus important qu'à l'heure actuelle, les modèles de détection et de classification de défauts ne peuvent remplacer le travail des analystes certifiés, car ces modèles ne considèrent qu'un faible nombre de défauts et la plupart des modèles ignorent totalement la localisation des défauts par rapport à la soudure [31-34]. Pourtant, ignorer la localisation du défaut par rapport à la soudure diminue la qualité de l'analyse et peut réduire le risque perçu d'un défaut. Par exemple, un défaut, telle une fissure, localiser près de la racine de la soudure est davantage critique qu'une inclusion située en plein cœur de la soudure.

Parmi les techniques d'analyse recensées, les RDN profonds et les réseaux de neurones convolutifs profonds font figure de chef de file. Ces deux méthodes permettent une classification sans extraction préalable de caractéristiques. Des *CNNs* sont actuellement capables de détecter entre 85 % et 98 % des principaux défauts d'une soudure (manque de fusion, manque de pénétration, fissure, porosité et inclusions) [10, 22]. Bien qu'ils représentent une nette amélioration par rapport aux techniques antérieures, notamment les RDN traditionnels et les machines à vecteurs de support (SVM) [17, 35], une méthode générale de détection des quelque 25 différents types de défauts possibles d'une soudure semble encore loin de voir le jour [36]. En effet, la majorité des modèles actuellement recensés classifient les données ultrasonores en fonction du type de défaut qui est inclus dans les données de soudure [37-41]. Bien que ces études représentent des avancées techniques intéressantes permettant de réduire le fardeau des analystes ainsi que le délai de traitement entre l'acquisition des données et la décision finale, ces études comportent plusieurs failles qui impactent leur mise en pratique à grande échelle [7, 42]. Les limites de ces études ont rapidement été dévoilées lors des essais en milieu industriel. Or, les modèles qui effectuent la détection, ce qui consiste à trouver les coordonnées d'une boîte encadrant le défaut à travers les données, n'utilisent pas de données ultrasonores en trois dimensions [43]. Les intrants analysés varient d'images ultrasonores à des images radiographiques [44-46] en passant par le courant de Foucault [47].

En raison de la difficulté de l'entraînement de modèles 3D, les méthodes d'apprentissage profond actuellement appliquées dans le domaine des END fonctionnent principalement sur des images en deux dimensions. En fait, l'apprentissage profond en trois dimensions est une tâche ardue. Les raisons proviennent des limitations suivantes : l'entraînement d'un réseau profond sur un grand volume peut s'avérer trop onéreux en termes de calcul pour une application réelle ; et un réseau profond utilisant des *patches* 3D comme intrant nécessite davantage d'échantillons d'entraînement, puisqu'un réseau 3D contient des paramètres d'un ordre de grandeur supérieur à celui d'un réseau 2D [48]. Cela augmente considérablement le risque de surapprentissage (*overfitting*), étant donné les données d'entraînement limitées. Le domaine des END a souvent du mal à disposer d'un nombre important d'échantillons d'entraînement en raison de la difficulté à générer et à obtenir des données de soudures, souvent propriété de compagnies n'ayant aucun incitatif à partager leurs données.

Étant donné la quantité importante de données ultrasonores, les erreurs et les fraudes sont inévitables. Les erreurs se produisent, entre autres, lors de la manipulation des données. Les opérateurs peuvent soumettre un mauvais fichier ou le même fichier à plusieurs reprises. Des fraudes peuvent être commises pour gagner du temps, de l'argent ou pour masquer le manque d'analystes qualifiés pour effectuer le travail. Elle se fait généralement en remplaçant un fichier inexistant par un fichier sans défaut obtenu précédemment. Les fichiers frauduleux peuvent provenir de fichiers

existants légèrement modifiés ou de nouveaux fichiers acquis par le scan de soudures sans défaut. Dans ce dernier cas, l'opérateur peut également modifier la configuration de l'appareil *PAUT* pour induire l'analyste en erreur. La validation de l'intégrité et de l'authenticité des données ultrasonores fournies pour analyse est donc aspect critique également.

Les logiciels d'analyse de données ultrasonores actuellement utilisées en END ne valident pas l'authenticité des données, ce qui permet aux utilisateurs de soumettre les mêmes données sous différents noms et dans des situations différentes. Chaque scan soumis pour analyse doit être certifié comme étant original. Cependant, la difficulté de comparer des milliers de scans similaires est importante et cela peut prendre beaucoup de temps pour un analyste. Le risque d'erreur humaine est donc très élevé. Comme les erreurs et les fraudes proviennent généralement de soudures déjà scannées, elles peuvent être considérées comme des doublons des fichiers originaux [49]. En effet, il n'y a pas de validation, ce qui fait que la détection de cas problématique est impossible. Il y a actuellement des cas de falsifications (copier-coller de données, scans successifs de soudure conforme, etc.) ou d'erreurs de manipulation.

En connaissant la signature d'une soudure, il sera non seulement possible d'en assurer un suivi périodique, mais également d'assurer l'intégrité des END d'une entreprise. En effet, la signature géométrique pourra également permettre d'éviter les falsifications par le scan répétitif de la même soudure. Il s'agit d'un cas connu de fraudes. En voici une brève description : un employé d'une firme d'audit scanne une soudure et envoie les données pour analyse. La soudure est évaluée par un analyste certifié qui la juge conforme. L'employé fautif procède alors à un deuxième scan (ou plus encore) de la même soudure : il possède alors différents fichiers de données de la même soudure qu'il sait conforme. Ce stratagème lui permet « d'emmagasinier » un plus grand nombre de soudures conformes, tout en minimisant le temps et les efforts requis.

Une méthode permettant de calculer la ressemblance de deux modèles géométriques de soudures (mesure de similarité basée sur les caractéristiques géométriques) sera implémentée afin d'envoyer l'information aux personnes touchées par ce problème dans le cas où la similarité des deux soudures est supérieure à un seuil établi. Compte tenu de l'importance des soudures, notamment dans le domaine du génie civil (pont, viaduc), des réservoirs sous pression (réservoir de pétrole, gazoduc, oléoduc) et du transport (bateau, remorque-citerne, sous-marin), il est d'une importance capitale que chacune des soudures soumises pour analyse soit conformément analysée, qu'aucune erreur ou falsification ne soit possible, que l'évolution des soudures soit facile à suivre et qu'aucune possibilité d'erreur ne subsiste. Dans le cas d'une soudure, le matériau testé peut comporter différents types de défauts, tels que des fissures, un manque de fusion, une porosité et une pénétration incomplète. Selon le type de défaut et leur taille, la soudure peut nécessiter une réparation, un contrôle périodique ou une reprise complète. La défaillance d'une soudure peut entraîner des conséquences

dramatiques. Par exemple, une explosion suite à une fuite d'un joint soudé non qualifié a causé la mort d'un chimiste et blessé 7 autres personnes à Singapour en 2015 [50]. Considérant la gravité potentielle des erreurs de diagnostic, il est toujours nécessaire qu'un analyste certifié contre-valide les conclusions des modèles de détection et de classification basés sur l'IA. Les méthodes actuellement utilisées ne sont pas suffisamment automatisées et performantes pour soustraire les analystes du processus de diagnostic des données ultrasonores.

Le Tableau 1.1 résume les différentes lacunes énoncées ci-dessus et en identifie les conséquences.

Tableau 1.1 : Principales conséquences des lacunes actuelles

Lacunes	Conséquences
Pénurie d'analystes qualifiés	Augmentation des délais Des erreurs ou des fraudes peuvent survenir fréquemment et facilement
Modèles d'IA ne sont pas suffisamment performants	Impossible de soustraire les analystes du processus de diagnostic
Modèles d'IA ne sont pas en mesure de localiser la soudure	Impossible de connaître l'impact réel d'un défaut
Modèles d'IA ne considèrent pas des intrants en 3D	Réduction de l'efficacité des modèles

1.3 Objectifs de la thèse

L'objectif principal consiste à reconstituer la géométrie 3D d'une soudure à partir de données ultrasonores issues d'un END. Cette reconstruction permet une meilleure validation des algorithmes actuels de reconnaissance des défauts, tout en offrant une avenue possible de détection de fichiers ultrasonores erronés ou falsifiés. De cet objectif général découlent quatre objectifs spécifiques :

- a) Assurer l'intégrité et l'authenticité des données ultrasonores soumises pour la première fois ;
- b) Identifier les différentes parties d'une soudure ;
- c) Reconstruire la surface géométrique 3D d'une soudure ;
- d) Assurer un meilleur suivi des analyses périodiques d'une même soudure.

La recherche se divise en plusieurs étapes, présentées en détail à la Figure 1.1. Les quatre zones pointillées encadrent chacun des quatre objectifs spécifiques ci-dessus. Les étapes en vert représentent les actions répondant directement à l'un des objectifs spécifiques du projet, tandis que les étapes en bleu représentent des actions externes au projet (par exemple la soudure) ou des décisions.

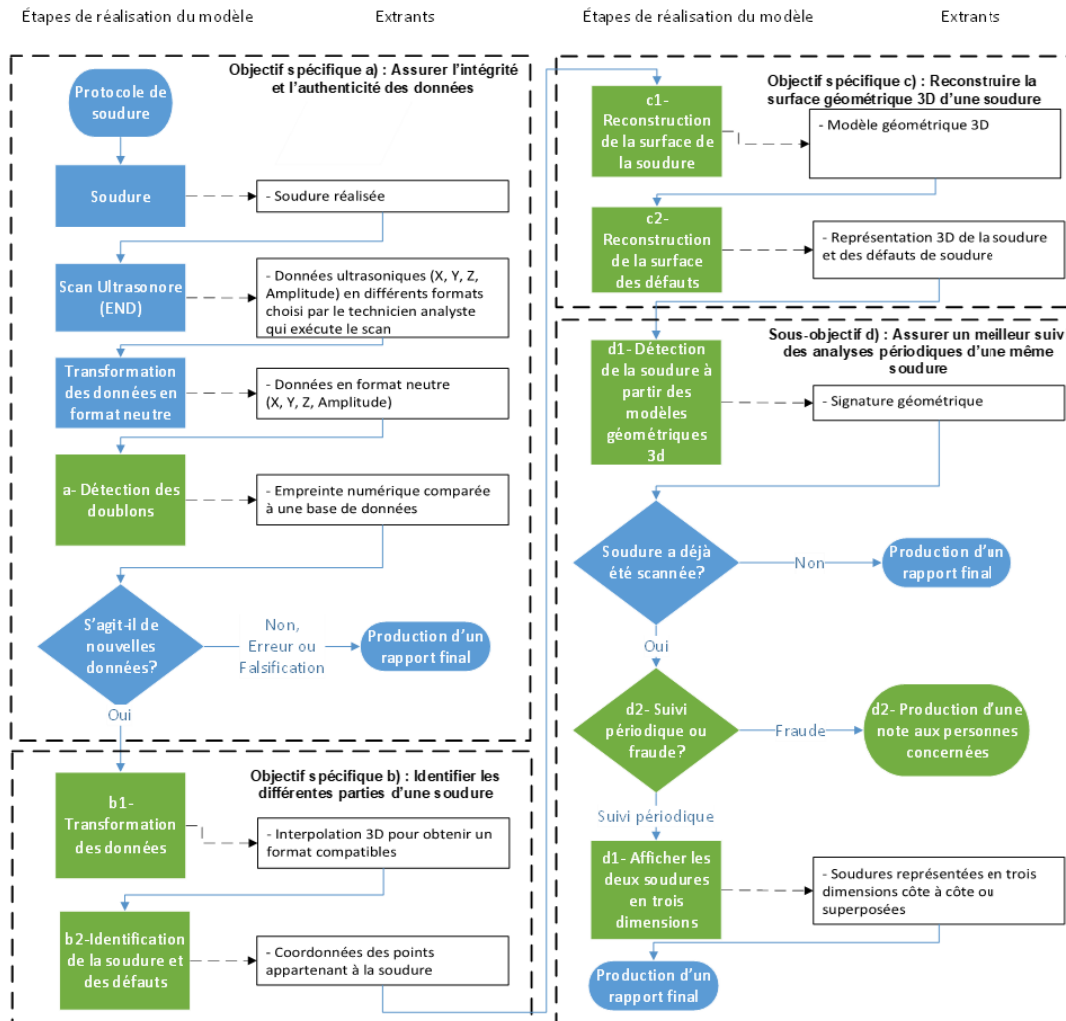


Figure 1.1 : Schéma des étapes de réalisation des 4 objectifs spécifiques

La validation de l'atteinte de chacun de ces quatre objectifs spécifiques est réalisée à partir de fichiers de données ultrasonores réelles, grâce à une collaboration du partenaire industriel Ondia dans le cadre d'une bourse Mitacs accélération. Les objectifs découlent directement de la vision de l'entreprise pour son logiciel d'analyse automatisée. Il est important de mentionner que les solutions développées dans ce projet seront implémentées dans le logiciel infonuagique de Ondia. Il est donc primordial de rechercher des solutions qui sont efficaces algorithmiquement.

Le reste de la thèse se présente comme suit. Le chapitre 2 discute de la revue de littérature, couvrant chacun des thèmes importants de ce projet. Le chapitre 3 expose la méthodologie utilisée. Le chapitre suivant présente les résultats de chacun des quatre objectifs de cette étude. Finalement, la conclusion au chapitre 5 revient sur les objectifs, discute des principales contributions, et précise les perspectives de travaux futurs.

2 REVUE DE LITTÉRATURE

Ce chapitre est divisé en cinq sections. Les quatre premières sont associées à chacun des quatre objectifs de cette recherche et une section finale discute de son positionnement. Les techniques permettant l'identification de doublons et la création d'une empreinte numérique débutent la revue de la littérature. Les avancements en IA, particulièrement au niveau de la vision artificielle par ordinateur, pertinents à cette recherche, sont ensuite présentés. Les concepts entourant le domaine des END y sont discutés, suivis par les études et techniques permettant la création et la visualisation d'un objet en trois dimensions. Par la suite, les méthodes permettant de faire un suivi périodique des soudures sont survolées. Pour conclure ce chapitre, une discussion sur le positionnement de cette recherche est présentée.

2.1 Détection des doublons

La littérature définit couramment deux types de doublons : le double exact et le quasi-doublon, traduction respective de *exact-duplicate* (*ED*) et de *near-duplicate* (*ND*) [51]. Un *ED* est une copie identique de l'original. L'identification des *ED* n'est pas une tâche complexe. La littérature comprend de nombreuses études démontrant que c'est une tâche facile, et ce même en considérant différents types de données [52]. La méthode traditionnelle de détection des *ED* consiste à créer un identifiant unique, basé sur les différentes caractéristiques des données d'entrée, pour chaque fichier [51]. Cet identifiant unique est ensuite ajouté dans une base de données. Lorsqu'un nouveau fichier est disponible, son identifiant unique est calculé et une requête est envoyée à la base de données pour vérifier s'il est déjà présent. Un *ED* est détecté si son identifiant unique est déjà présent dans la base de données.

Les algorithmes traditionnellement utilisés pour créer un identifiant unique sont les fonctions de hachage. Les fonctions de hachage calculent une valeur de hachage unique pour une entrée donnée [53]. La valeur de hachage est de longueur fixe, généralement 64 bits. Une propriété très intéressante des fonctions de hachage est qu'elles sont très sensibles aux changements des données d'entrée. La même entrée génère toujours la même valeur de hachage alors qu'une légère modification de l'entrée produira une grande modification. La valeur de hachage peut être calculée, quelle que soit la longueur des données d'entrée. La probabilité de conflit, qui implique de générer la même valeur de hachage pour différentes valeurs d'entrée, est très faible. De plus, les fonctions de hachage nécessitent très peu de puissance de calcul, même pour des données d'entrée importantes. Une autre propriété intéressante est qu'une fois la valeur de hachage générée, il n'est plus nécessaire de conserver le contenu des données d'entrée pour effectuer d'autres comparaisons (requêtes) dans la base de données. Cela signifie que les scans *PAUT*, qui peuvent contenir plusieurs millions de points en quatre dimensions, n'ont pas besoin d'être entièrement sauvegardés dans la base de données,

ce qui permet d'économiser de la puissance de calcul et du temps. Les fonctions de hachage telles que SHA256 ou MD5 peuvent détecter efficacement les *ED* dans une grande base de données [53, 54]. De telles fonctions sont utilisées dans les *Blockchains* pour identifier les blocs et assurer l'intégrité de la chaîne [55].

La littérature n'inclut que très peu d'études ayant eu à déterminer une empreinte numérique pour des soudures. Il n'existe aucun exemple d'utilisation d'empreinte numérique dans le but d'éviter les erreurs de manipulations et les fraudes par falsification des données. En règle générale, les modèles d'analyses des défauts ne vérifient pas l'identité des soudures et ne peuvent donc garantir l'authenticité des données associées à celles-ci. En revanche, il existe des recherches qui ont utilisé un certain type de signature (par exemple acoustique, image, etc.) pour reconnaître des défauts de soudure [56-58]. Bien que ces recherches soient intéressantes d'un point de vue de la détection de défauts, la signature obtenue n'est pas utilisée afin d'identifier de façon unique chacune des soudures. Ces méthodes de détection de défauts ne sont donc pas à l'abri d'erreurs ou de fraudes.

La détection des *ND* consiste à trouver des versions altérées, légèrement modifiées ou alternatives des données originales dans un vaste ensemble de données [59]. La détection d'un *ND* est une tâche beaucoup plus complexe que la détection d'un *ED*. Le développement d'un détecteur de *ND* efficace est un domaine de recherche très actif [59-62]. La détection automatique des *ND* n'est pas une tâche triviale pour diverses raisons. Premièrement, les *ND* présentent des différences inhérentes dans leurs valeurs. Deuxièmement, comparer un à un chaque fichier d'une base de données pour rechercher les doublons possibles est tout simplement impossible en temps raisonnable lorsqu'il s'agit de grands ensembles de données. Par conséquent, une méthode efficace réduit la nécessité de comparer des paires de fichiers, sous un seuil acceptable.

Les chercheurs se sont attardés au problème de la détection des *ND* pendant de nombreuses décennies compte tenu de son importance dans divers domaines [51, 60, 63]. Les *ND* sont présents dans plusieurs aspects de la société moderne. Ils sont présents dans les médias sociaux, dans la réutilisation du matériel photographique, dans la détection des violations des droits d'auteur, à la fraude par copier-coller et à de nombreux autres cas [59]. Dans toutes ces applications, les fonctions de hachage standard ne peuvent pas détecter correctement les *ND*, car de légères modifications des données originales entraînent une différence significative de la valeur de hachage (identifiant unique), trompant les requêtes à la base de données.

Les méthodes de détections des *ND* sont généralement divisées en deux étapes : (1) l'algorithme passe en revue tous les fichiers d'une base de données et détermine quels fichiers sont les η voisins les plus proches du fichier original, où η est un nombre prédéterminé et (2) les η voisins les plus proches sont comparés au fichier original en utilisant une mesure de similarité. Il est alors possible

de classer les η voisins les plus proches en fonction de leur similarité avec le fichier original. Ce processus comporte deux éléments principaux : (1) un algorithme qui fonctionne efficacement dans un grand ensemble de données et (2) une mesure de similarité qui peut effectivement quantifier la similarité de deux fichiers.

Dans la famille des algorithmes de recherches des plus proches voisins, la fonction de hachage *Locality Sensitive Hashing* (LSH) est très utilisée puisqu'elle peut traiter des entrées de grande dimension sous la forme d'un grand ensemble de données. *LSH* est une variante des fonctions de hachage qui est moins sensible aux changements que les fonctions de hachage standard [64]. Le principe de *LSH* est que les points proches dans l'espace d'origine ont une forte probabilité d'avoir la même valeur de hachage. Ainsi, de petites différences n'occasionnent pas de grand changement. Cette fonction est donc plus robuste que les fonctions de hachage standard. *LSH* est un algorithme efficace de recherche des plus proches voisins. Il peut alors être utilisé pour faire une recherche par similarité ou pour trouver des doublons (*ED* et *ND*) [63]. De ce fait, *LSH* semble en mesure de créer des signatures numériques efficacement tout en permettant de retracer les cas où les signatures numériques sont proches, ce qui peut correspondre à des erreurs ou fraudes.

Les études récentes sont désormais basées sur des RDN profonds [65]. Différents ensembles de paires d'images sont utilisés pour entraîner le RDN profond. Le raisonnement qui sous-tend ces réseaux est de classer les données en fonction de leur ressemblance [66]. L'avantage d'appliquer des RDN profonds, tel que les *CNNs*, au problème de la détection des *ND* est que ces réseaux peuvent apprendre des abstractions hiérarchiques de haut niveau proches du processus cognitif humain. Deux méthodes sont utilisées dans la littérature. La première tire profit d'un modèle pré-entraîné pour l'extraction de caractéristiques [65]. La seconde méthode fait appel à un RDN formé à l'apprentissage d'une fonction de classement dans un processus de complet, de sorte qu'il cartographie l'espace d'entrée dans un espace latent afin que la distance de similarité (par exemple la distance euclidienne ou la distance cosinus) puisse se rapprocher de la similarité visuelle [66]. La fonction de perte qui classe les éléments est basée sur une architecture spéciale appelée réseau siamois. L'utilisation d'un RDN profond pour détecter des doublons donne des résultats prometteurs. Cependant, elle nécessite un entraînement et une grande puissance de calcul pour traiter les données en deux ou trois dimensions. Ceci limite clairement leur utilisation dans un environnement infonuagique comme celui ciblé par cette thèse.

2.2 Identification de la soudure

Les sections 2.2.1 et 2.2.2 sont divisées en fonction des objectifs des modèles d'IA développés : classification et segmentation. La classification de données de soudure, majoritairement les défauts,

fait partie du parcours historique pertinent en contrôle par ultrasons, elle sera donc discutée en premier.

2.2.1 Classification des défauts de soudure

La classification des défauts de soudure est bien documentée dans la littérature [14, 29]. Au début, des techniques d'extraction de caractéristiques sont utilisées pour obtenir les caractéristiques statistiques à partir des A-Scans ou pour débruiter le signal. Ces caractéristiques sont alors utilisées comme intrant à des modèles d'apprentissage machine ou des RDN conventionnels, tel que le *Multi-Layer Perceptron (MLP)* et les machines à vecteurs de support [67, 68]. Ces premières études obtiennent des résultats intéressants dans certaines situations (données sans bruit, données de laboratoire), mais performant moins bien que les analystes certifiés et ont de la difficulté dans les conditions réelles (industrielles) [17, 69, 70]. Ces premiers classificateurs intelligents peuvent classer correctement entre quatre et six types de défauts différents. Leur précision varie de 80 % à 91 %.

La littérature fait état de deux approches distinctes : l'utilisation d'algorithmes d'extraction de caractéristiques, tels que des algorithmes de traitement du signal pour débruiter les données d'entrée et extraire les caractéristiques [20, 37, 40, 71-73] ou l'utilisation sans débruitage du signal comme intrant [22, 74, 75]. La première approche transforme les données obtenues des A-Scan par un traitement de signal avant de procéder à la classification des défauts. Le traitement de signal a pour objectif de raffiner l'information disponible, soit débruiter les données et extraire des caractéristiques qui décrivent les données, ce qui augmente les performances des modèles. La deuxième méthode utilise directement les signaux des A-Scan comme intrant du modèle de détection et de classification des défauts. Elle permet d'automatiser davantage le processus en limitant les actions d'un opérateur [13, 22, 35].

Les sections 2.2.1.1 et 2.2.1.2 décrivent les techniques utilisées afin de classer les défauts de soudure à partir des données ultrasonores. La section 2.2.1.1 décrit les méthodes avec le débruitage des signaux ultrasonores alors que la section 2.2.1.2 présente les méthodes qui n'ont pas recours au débruitage des signaux ultrasonores.

2.2.1.1 Méthodes avec débruitage

La détection et la classification des défauts de soudures à partir de données ultrasonores sont un domaine de recherche actif et de nombreuses approches basées sur l'IA ont été appliquées pour automatiser ce processus [39, 74, 76-78]. L'objectif est de classer les différents signaux selon le type de défaut rencontré. En END, la méthode classique pour classer les défauts de soudure obtenus à partir de scans ultrasonores consiste à utiliser un RDN et les données ultrasonores de type A-Scan. Les données annotées correspondent au type de défauts observés lors de l'analyse par un analyste

certifié. Les RDN sont capables de classer correctement la plupart des défauts suivants : porosité, manque de fusion, inclusion, pénétration incomplète et fissure (à différents endroits) [15, 35, 41].

Le signal des scans ultrasonores est bien souvent contaminé par du bruit provenant de diverses sources, par exemple du système de mesure, du matériau lui-même, du cordon d'alimentation, du milieu électromagnétique où la machine à souder est en fonction, de la trop rapide répétition des impulsions (*Pulse Repetition Frequency*) et de la rétrodiffusion (*backscattering noise*), ce qui masque les échos des défauts dans le signal mesuré [79]. Par exemple, lorsque la taille du grain du matériau est similaire à la taille du défaut, son écho sera masqué par le bruit structurel. Ce bruit limite l'efficacité des analystes et des méthodes intelligentes de détection des défauts [67, 80].

Les premières recherches ont montré que l'utilisation d'un algorithme de traitement du signal comme étape de prétraitement d'un classificateur améliore considérablement les performances globales. Le recours au traitement du signal est utilisé pour éliminer le bruit afin d'améliorer le ratio signal sur bruit (*Signal to Noise Ratio*) et extraire les caractéristiques pour la classification des défauts. Le rapport signal sur bruit est un indicateur de la qualité de la transmission d'une information correspondant au rapport de puissance entre le signal d'amplitude maximale et le bruit de fond [13]. Il est donc important de maximiser ce rapport pour faciliter l'exploitation des données.

Dans ce contexte, différents algorithmes de traitement du signal ont été testés afin de déterminer ceux qui donnent les meilleurs résultats. La transformée de Fourier, la transformée cosinus, l'analyse en composantes principales (*Principal Component Analysis : PCA*) et la transformée en ondelettes (*wavelet transform*) ont toutes été comparées, sans déterminer à l'unanimité un gagnant [73]. Il est important toutefois de noter que la transformation en ondelettes est la méthode d'extraction de caractéristiques la plus utilisée en END [14, 21, 27, 67]. La transformation convertit alors un signal de son domaine d'origine (souvent le temps ou l'espace) en une représentation dans les domaines du temps et/ou des fréquences et vice versa [68, 81]. Cela permet d'améliorer le rapport signal sur bruit et d'extraire des caractéristiques pertinentes pour une analyse plus approfondie et une classification des défauts. Les caractéristiques descriptives extraites (moyenne, variance, amplitude maximale, amplitude minimale, énergie maximale, fréquence moyenne, fréquence minimale, demi-point) sont ensuite utilisées comme intrant du classificateur. Les coefficients d'ondelettes sont utilisés en entrée des classificateurs, tels qu'un RDN [35, 37, 82] ou une machine à vecteurs de support [32]. Leurs performances de classification peuvent atteindre une précision de 98 % sur des ensembles de données privées [15, 41].

Le débruitage des données a pour objectif d'extraire le bruit y et l'image x d'une image bruitée Z ou $Z = x + y$ [80]. La plupart des techniques essaient d'approximer x à partir de Z en supposant que y est généré par un processus bien défini, par exemple suivant une fonction gaussienne. Dans le domaine

des END par ultrasons, les techniques de traitement de signal, notamment la transformation en ondelettes, sont les techniques les plus utilisées [14, 21, 83, 84]. Plus récemment, les *Denoising Auto-Encoder* sont devenus plus couramment utilisés pour réduire le bruit d'une image ultrasonore [80]. Il suffit d'entraîner un auto-encodeur à partir d'images bruitées artificiellement et faire apprendre le réseau à reconstruire les intrants débruités. Les images bruitées utilisées pour l'entraînement doivent correspondre fidèlement, en termes de distribution du bruit, aux images réelles utilisées lors de l'inférence. Les *Denoising Auto-Encoder* ont deux objectifs complémentaires : extraire les caractéristiques les plus représentatives des intrants, ce qui correspond à l'espace latent, et minimiser l'effet du bruit appliqué sur les intrants. La fonction de perte compare alors les valeurs de sorties avec les intrants bruités et apprend à reconstruire l'image de façon à éliminer le bruit. Il est à noter que *Denoising Convolutional Auto-Encoder* est la version améliorée des *Denoising Auto-Encoder* partageant la même structure que les *CNNs* [72]. Comparés au *Denoising Auto-Encoder*, les *Denoising Convolutional Auto-Encoder* sont plus robustes faces aux images corrompues. Ils peuvent être utilisés pour débruiter des données ultrasonores brutes, permettant ensuite de distribuer les données à un modèle pour une classification [85]. Les résultats de cette étude montrent qu'un *Denoising Convolutional Auto-Encoder* peut débruiter avec succès les données brutes ultrasonores, ce qui améliore par conséquent la performance de classification des défauts de soudure des modèles d'analyse.

Dans [18], deux méthodes ont été développées pour l'extraction de caractéristiques. La première empile deux *Sparse Auto-Encoder* et extrait les activités dans la couche cachée (*hidden layer*) de la dernière *Sparse Auto-Encoder* tandis que la seconde méthode est basée sur un *CNN* profond. Les caractéristiques sont extraites d'images radiographiques pour la classification des défauts de soudure (manque de pénétration, porosité, inclusion de scories et fissures). Ces caractéristiques sont comparées aux caractéristiques extraites à l'aide de la méthode traditionnelle de la matrice de co-occurrence de niveau de gris. Le *CNN* atteint une précision de 97 % alors que les méthodes d'extraction traditionnelles atteignent une précision de 82 %.

Il est à noter que les études utilisant le traitement de signal n'utilisent pas les données ultrasonores directement comme intrant. En effet, ces études utilisent des caractéristiques extraites à partir du traitement de signal. Il est alors impossible pour les modèles utilisant cette approche de localiser le défaut dans les données brutes. Ces modèles ont simplement pour objectif de classer correctement le type de défaut.

2.2.1.2 Méthodes sans débruitage

Plusieurs études avancent que l'utilisation de ces techniques de débruitage n'est pas nécessaire et que ces techniques augmentent inutilement la complexité du processus [10, 22, 35]. Avec

l'augmentation récente de la puissance de calcul des ordinateurs, la taille et la complexité des RDN profonds ont augmenté de façon importante, ce qui a entraîné une amélioration importante des performances des RDN et plus particulièrement des modèles d'évaluation des défauts [39, 74]. Il n'est plus nécessaire d'extraire les caractéristiques des données brutes, les RDN peuvent désormais traiter des entrées et des ensembles de données volumineux. Cela permet d'automatiser davantage le processus et d'augmenter les performances des modèles d'analyse [22, 35]. À plusieurs occasions, les *CNNs* profonds ont surpassé les méthodes traditionnelles de classification des défauts de soudures, et ce à partir des données brutes bruitées [10, 75].

Les *CNNs* utilisent les signaux A-Scan comme intrants pour classifier les défauts de soudure [13, 35, 73]. Les *CNNs* sont en mesure de traiter des signaux ultrasonores bruités, car leur structure est plus complexe (plus profonde). Les *CNNs* sont des RDN profonds qui combinent des couches convolutives et des couches entièrement connectées. Les premières couches se spécifient dans l'extraction des caractéristiques, tandis que la deuxième partie du modèle est constituée des couches pour la classification. Les couches d'extraction de caractéristiques sont des couches de convolutions et de regroupement (*pooling*), tandis que la classification est effectuée par une couche entièrement connectée. Les *CNNs* peuvent apprendre des *patterns* dans différentes zones des images en se concentrant sur des caractéristiques de bas niveau qui sont assemblées en caractéristiques de haut niveau. Il permet la classification sans extraction préalable des caractéristiques. Les *CNNs* ont toutefois besoin d'un large ensemble de données pour l'apprentissage. L'utilisation de techniques d'augmentation des données pour accroître la taille de l'ensemble d'entraînement est commun [47].

Des études récentes indiquent qu'un RDN profond peut augmenter la performance de classification, permettant d'atteindre une précision allant jusqu'à 99 %, ce qui est supérieur à la précision d'un analyste certifié [28, 74, 75]. Selon ces études, les *CNNs* peuvent traiter des données bruitées brutes et atteindre une grande précision pour la classification des défauts de soudure les plus fréquents. L'un des avantages des *CNNs* est qu'ils peuvent traiter des entrées à haute dimension, comme des données ultrasonores tridimensionnelles. Par exemple, les défauts de soudure peuvent être détectés dans les scans ultrasonores et être identifiés par des boîtes (*bounding box*) [86].

Un réseau entièrement connecté avec *dropout* (abandon) a été proposé pour classer les défauts de soudure à partir de signaux ultrasonores mixtes sans extraction préalable de caractéristiques [22]. Pendant l'entraînement, les couches *dropout* sont désactivées de manière aléatoire, selon une probabilité choisie, ce qui évite au réseau de concentrer les poids sur un nombre des neurones limités. L'efficacité de leur modèle est évaluée en comparant les performances d'un réseau peu profond à celles d'un réseau profond avec des couches *dropout*. Les résultats indiquent que l'architecture du réseau profond est plus robuste et permet une meilleure précision de classification sans extraction de caractéristiques ni débruitage des signaux ultrasonores.

La plupart des classificateurs développés dans la littérature utilisent des données générées artificiellement, ce qui implique des scans ultrasonores presque sans bruit. Cependant, le bruit est fréquent dans les applications industrielles. Une étude compare l'efficacité d'un *CNN* à différents niveaux de bruit [13]. Des techniques d'augmentation des données (décalage des signaux) et des couches de *dropout* ont été utilisées pour éviter le surapprentissage. Les résultats montrent que le *CNN* est très performant pour un faible niveau de bruit. Sa précision diminue lorsque le niveau de bruit augmente. L'efficacité du modèle varie de 88 % à 99 % pour des signaux à bruit élevé ou faible, lorsqu'entraînée sur une base de données incluant les principaux types de défauts de soudure.

Un *Sparse Auto-Encoder* a été entraîné à la détection automatique des défauts de soudure à partir de scanners radiographiques [19]. Un *Sparse Auto-Encoder* peut inclure plus de neurones cachés que de neurones à l'entrée, mais seul un petit nombre de neurones cachés peut être actif en même temps. Son entraînement est d'abord basé sur des algorithmes d'apprentissage non supervisés qui ne nécessitent pas de données annotées. L'entraînement est ensuite affiné en utilisant un petit ensemble de 88 données annotées, bien qu'un prétraitement important des données soit nécessaire. Un *sliding window* est utilisé pour détecter l'emplacement des défauts. Le *Sparse Auto-Encoder* atteint une précision de classification maximale de 91,84 %. Une telle approche est un cas très intéressant pour les situations où les données annotées sont difficiles à produire.

Une étude visant à identifier les défauts de délamination d'un matériau composite à partir de signaux A-Scan standardisé a permis de générer une représentation 3D des défauts, augmentant ainsi considérablement la quantité d'informations qu'un analyste peut utiliser pour les détecter [10]. Cette étude utilise les signaux ultrasonores pour classer les défauts des plaques polymères renforcées de fibres de carbone. La première étape consiste à appliquer *wavelet packet decomposition* aux signaux ultrasonores. Les caractéristiques sont ensuite transmises à un *CNN* pour la classification. Le modèle atteint une précision de classification de 98,15 %. Des C-Scan ont été utilisés pour créer une représentation 3D des défauts. Les points appartenant à la même classe de défauts dans une zone spécifique sont conservés tandis que les points restants sont éliminés et considérés comme du bruit de fond. La reconstruction des défauts est effectuée en utilisant la position de l'amplitude maximale du A-Scan en un point spécifique pour localiser le défaut et ensuite, la représentation du modèle 3D est créée par un ajustement de la surface lisse pour chaque classe de défauts. Dans le domaine du soudage, la littérature ne rapporte aucune approche permettant la génération d'une telle représentation complète en trois dimensions (éléments géométriques de la soudure avec ses défauts possibles). Les données géométriques sont totalement absentes du processus d'identification. Pourtant, la localisation de chacun des éléments géométriques d'une soudure (métal de base, ligne de fusion, métal fondu, défauts) augmenterait à coup sûr tant la performance de classification [87] que l'efficacité d'identification des défauts [39, 40].

La plupart des systèmes automatisés de classification des signaux ultrasonores sont basés sur des images, ce qui signifie qu'ils peuvent classer les défauts dans des images. Les signaux A-scan peuvent être considérés comme une série temporelle, il est donc possible d'appliquer un algorithme de classification des séries temporelles tel que le *Gated Recurrent Unit* [28]. Une combinaison de *Fully Convolutional Network (FCN)* et de *Gated Recurrent Unit* a été appliquée au problème de la classification des défauts de soudure. Le *FCN* contient six convolutions temporelles et constitue le principal extracteur de caractéristiques, tandis que le *Gated Recurrent Unit* est composé de deux couches. Les extraits de ces deux architectures sont concaténés et transférés à une couche softmax pour la classification binaire. L'ajout de couches *Gated Recurrent Unit* augmente la précision de la classification. Le réseau atteint une performance de classification binaire de 99,72 % (absence ou présence de défauts).

En END, les modèles de détection des défauts visent à localiser le défaut dans des scans ultrasonores. Ces modèles permettent généralement de localiser le défaut et de créer une boîte de délimitation autour de l'objet. En utilisant la précision moyenne (*MAP*) comme mesure de performance, *You Only Look Once (YOLO)* [88] et *Single-Shot Detector (SSD)* [89] se sont révélés être parmi les modèles de détection les plus précis et les plus rapides. *You Only Look Once* est un modèle de détection d'objets permettant de localiser les objets dans les images en les identifiant à l'aide de boîtes. Le modèle a pour objectif de déterminer les coordonnées de la boîte (x_{centre} , y_{centre} , x_{largeur} , y_{hauteur}) et de déterminer la catégorie de l'objet localisé. *YOLOv3* (la troisième version), basée sur *ResNet* et *Feature Pyramid Network*, consiste en une série de couches de convolutions (52 au total), qui inclut des *skip connections*, permettant d'obtenir des cartes de caractéristiques à différentes tailles, ce qui augmente la capacité du modèle à détecter des objets de différentes tailles. Plusieurs boîtes peuvent être prédites par *You Only Look Once* pour un même objet. L'algorithme *Non-Maximum Suppression* est alors appliqué. Cette technique détermine quelles boîtes se superposent, trouve la meilleure boîte et supprime toutes les autres boîtes.

En END, *You Only Look Once* et *Single-Shot Detector* ont été récemment implémentés dans un problème de détection de défauts. Un total de 490 images B-Scan contenant 1562 annotations ont été utilisées. Un sous-ensemble de 157 annotations classé comme difficile en raison de ses échos géométriques et du bruit de fond est inclus [86]. L'augmentation des données est utilisée pour ajouter des données d'entraînement. *Single-Shot Detector* atteint un AP de 84,5 % sur un ensemble de tests de 98 images faciles, tandis que *You Only Look Once* atteint 89,7 %. Les résultats diminuent respectivement à 81,4 % et 85,6 % lorsque les cas difficiles sont inclus. Cependant, l'utilisation d'images (2D) comme intrants ne permet pas de mesurer précisément les défauts, ces modèles ne prennent en compte que des tranches du défaut. L'étape suivante consiste à appliquer ces architectures données *PAUT 3D*.

Il est important de noter qu'il n'existe actuellement aucune base de données publique de scans ultrasonores de soudure [40]. Étant donné que les performances des modèles d'apprentissage profond dépendent des données d'entrée (quantité et qualité), les performances présentées ne peuvent pas être comparées explicitement. Néanmoins, le Tableau 2.1 présente un recueil des meilleures pratiques rencontrées en END. Il est à ce stade impossible de déterminer le meilleur classificateur sans effectuer une analyse comparative à l'aide d'une base de données commune.

Tableau 2.1 : Résumé des études de classification en END utilisant l'IA

Études	Année de publication	Types d'intrant	Méthode d'extraction des caractéristiques	Architecture	Performance de classification (%)
[10]	2017	Signal ultrasonore	<i>Wavelet Packet Transform</i>	<i>CNN</i>	98
[13]	2018	Signal ultrasonore	Aucune	<i>CNN</i>	88-99 dépendant du bruit
[18]	2019	Images radiographiques	Aucune	<i>CNN</i>	97
[19]	2018	Images radiographiques	Défauts de soudure	<i>Sparse Auto-Encoder</i>	92 (classification)
[22]	2018	Signal ultrasonore	Aucune	Dense	92
[28]	2019	Signal ultrasonore	<i>Gated Recurrent Unit</i>	<i>FCN</i>	99,7
[39]	2018	Images optiques	Aucune	<i>CNN</i>	96
[74]	2018	Signal ultrasonore	Aucune	<i>CNN</i>	95
[75]	2019	Signal ultrasonore	Aucune	<i>CNN</i>	95
[85]	2020	Signal ultrasonore	<i>Denoising Auto-Encoder</i>	<i>CNN</i>	97-99 dépendant du bruit
[86]	2019	Images ultrasonores	Détection de défauts	<i>You Only Look Once</i>	AP de 89,7

La recherche d'un classificateur efficace des défauts de soudure a conduit à tester de nombreuses architectures différentes, dont les *CNNs*, les auto-encodeurs et les *Gated Recurrent Unit*. Les meilleures performances actuelles appartiennent à des *CNNs* pour la classification des défauts de soudure à partir de scans ultrasonores sans bruit [13, 22]. Par exemple, une combinaison d'un auto-encodeur et d'un *CNN* peut atteindre une performance de classification de 99 % [85]. Ces classificateurs deviennent de précieux outils dans le processus d'analyse des données ultrasonores.

Parmi la liste de plus de vingt défauts de soudure différents, les plus couramment considérés sont la porosité, le manque de fusion, l'inclusion de scories (*slag inclusion*), la pénétration incomplète et les fissures (à différents endroits) [9, 36, 90]. Si les récentes implémentations permettent d'obtenir une très grande performance de classification des défauts les plus fréquents, tous les défauts de soudure ne sont pas pris en compte. Certains défauts spécifiques sont très peu fréquents dans les applications industrielles, c'est pourquoi les études sont généralement limitées à quatre ou cinq types différents.

2.2.2 Segmentation à l'aide de l'apprentissage profond

La segmentation se spécialise dans la détection précise des objets, plus particulièrement leur contour, et la classification de ceux-ci. Le principe de la segmentation d'images pourra ainsi être modifié pour ne pas considérer des pixels (indice sur 256 de rouge, bleu et vert) par position, mais des amplitudes par position (X, Y, Z) [91-93]. En général, pour effectuer la segmentation, les RDN profonds de type *FCN* sont utilisés. Ces réseaux produisent des cartes de segmentation permettant de connaître l'appartenance de chaque voxel à une classe d'objets, par exemple plaque d'acier, soudure, rebonds, défauts et autres [94, 95].

2.2.2.1 Segmentation à partir d'algorithmes supervisés

Les *CNNs* représentent une architecture classique utilisée pour segmenter efficacement les objets dans une image [96]. *U-Net* est l'un des réseaux convolutifs les plus populaires. *U-Net* est un *FCN*, un réseau entièrement convolutif, développé à l'origine pour la segmentation en imagerie médicale. Ce modèle consiste en deux parties : un encodeur consistant en une succession d'opérations de contraction suivi par un décodeur, c'est-à-dire une série d'étapes d'expansion de l'image, donnant à l'architecture une forme de U. L'encodeur capture le contexte tandis que la partie d'expansion permet une localisation précise de l'objet. L'encodeur consiste en une couche *CNN* typique, suivi d'une unité linéaire rectifiée (*ReLU*) et d'un *max-pooling*. L'encodeur réduit l'information spatiale tout en augmentant l'information sur les caractéristiques (*features map*). Le décodeur consiste en des convolutions ascendantes (*up-convolution*) et des concaténations à haute résolution. L'idée est de remplacer l'opération typique de mise en commun par des opérations de suréchantillonnage (*upsampling*) afin d'augmenter la taille de l'image jusqu'à ce qu'elle atteigne sa taille initiale. À chaque bloc, des *skip connections* transfèrent l'information résiduelle de l'encodeur vers le décodeur, ce qui permet de conserver l'information spatiale. L'architecture de *U-Net* est présentée à la Figure 2.1.

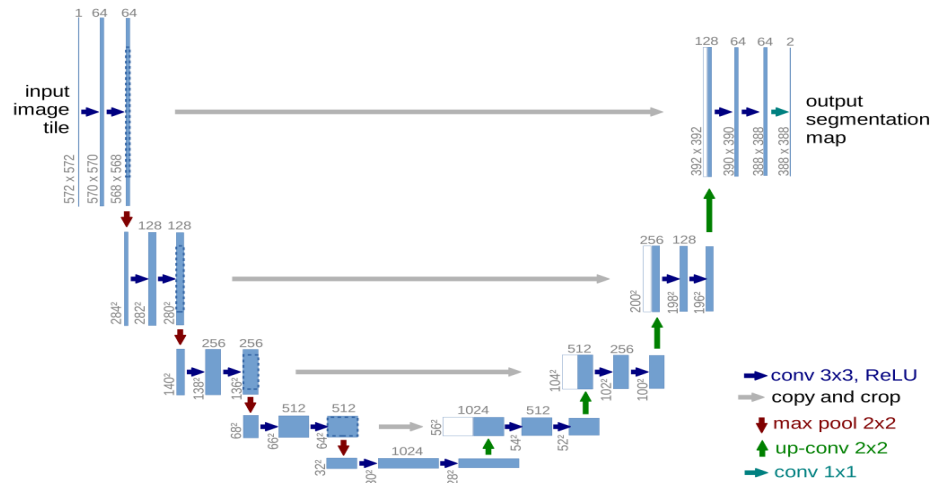


Figure 2.1 : L'architecture de *U-Net*, tiré de [96]

Lors de sa publication en 2015, *U-Net* est devenu la référence en segmentation d'images. En effet, *U-Net* a été testé sur différents ensembles de données biomédicales surpassant les autres architectures dans chacun des cas, en se basant sur la très populaire *Intersection over Union (IoU)* comme mesure de performance. De plus, la segmentation est rapide, nécessitant moins d'une seconde par image. Un autre avantage est que *U-Net* peut être entraîné en n'utilisant qu'un petit nombre de données d'entraînements, et ce sans que ces performances en souffrent. Dans la publication originale, seules 35 images originales ont été utilisées pour l'entraînement. L'augmentation des données est alors utilisée [96].

Diverses améliorations ont été apportées à l'architecture *U-Net* originale, permettant d'obtenir des performances de référence dans diverses tâches [97-99]. Une adaptation de *U-Net* réussit à segmenter des images ultrasonores au lieu des images traditionnelles [97]. Un modèle 3D a été adapté à partir du modèle original pour effectuer de la segmentation volumétrique. Les convolutions 2D sont remplacées par leur équivalent 3D [99]. Le modèle apprend alors des images volumétriques peu annotées. Un problème important pour les modèles 3D est la quantité de données disponibles pour l'entraînement, car les modèles 3D requièrent exponentiellement plus de données d'entraînement que les modèles 2D. Ce problème est résolu en utilisant des images volumétriques peu annotées comme données d'entrée. Seules des tranches 2D annotées sont donc nécessaires pour générer une segmentation volumétrique dense. Le réseau atteint une *IoU* moyenne de 86,3 %. D'autres implémentations 3D de *U-Net* sont appliquées avec succès à la segmentation du cerveau en utilisant des images de résonance magnétique comme données d'entrée [100, 101].

No-New-UNet (nnU-Net) est l'une des implémentations les plus intéressantes de *U-Net*, car il s'agit d'un modèle construit pour s'adapter à diverses tâches de segmentation [98]. L'objectif de cette version améliorée de *U-Net* est de développer un modèle plus robuste, s'adaptant facilement à d'autres domaines que la segmentation de données médicales. *nnU-Net* propose une structure en cascade afin d'augmenter les performances du réseau ainsi que l'utilisation de *LeakyRelu* comme fonction d'activation et une combinaison de *dice* et d'entropie croisée comme fonction de perte. Ceci se traduit en de meilleures performances que *U-Net* sur un ensemble varié de tâches de segmentation (10 objets médicaux différents).

Basé en partie sur la structure de *U-Net*, *V-Net* est un modèle de segmentation d'images très performant [102]. Comme *U-net*, *V-Net* est composé de deux blocs principaux : un encodeur et un décodeur. L'encodeur est divisé en quatre étapes qui incluent d'une à trois convolutions chacune. À chaque étape, une fonction résiduelle est apprise. L'intrant de chaque étape est transmis aux couches de convolutions, aux couches d'activations non linéaires et ajoutées à l'extrant de la dernière couche de convolution de cette étape. Cette architecture permet une meilleure convergence que les modèles sans fonction résiduelle (tel que *U-Net*). Pour compresser les intrants, des convolutions avec un kernel de 2x2x2 et un stride de 2 sont utilisées. Le remplacement des couches de *pooling* avec des convolutions permet de réduire la consommation de mémoire durant l'entraînement. Le fait de réduire la taille des intrants augmente la taille du champ récepteur (*receptive field*). Finalement, *Parametric Rectified Linear Unit (PReLU)* est utilisée en tant que fonction d'activation non linéaire [103]. *Parametric Rectified Linear Unit* diffère de *Relu* au niveau de la gestion des valeurs négatives. *Relu* sature les valeurs négatives à une valeur nulle (zéro) alors que *Parametric Rectified Linear Unit* considère les valeurs négatives en les pénalisant de façon paramétrique. La partie décodeur est composée de quatre étapes permettant d'augmenter la résolution des données compressées afin de produire une segmentation multi-classe. À chaque étape du décodeur, une déconvolution est appliquée afin d'augmenter la taille des données, suivie par une à trois couches de convolutions. Des fonctions résiduelles sont également présentes, comme dans la première partie du modèle. Le modèle reconstruit les données de la même taille que les intrants, produisant une carte de segmentation multi-canaux, où pour chaque voxel une probabilité d'appartenance à chaque classe est disponible. Finalement, des *skips connections* sont présentes dans le modèle afin de garder l'information spatiale.

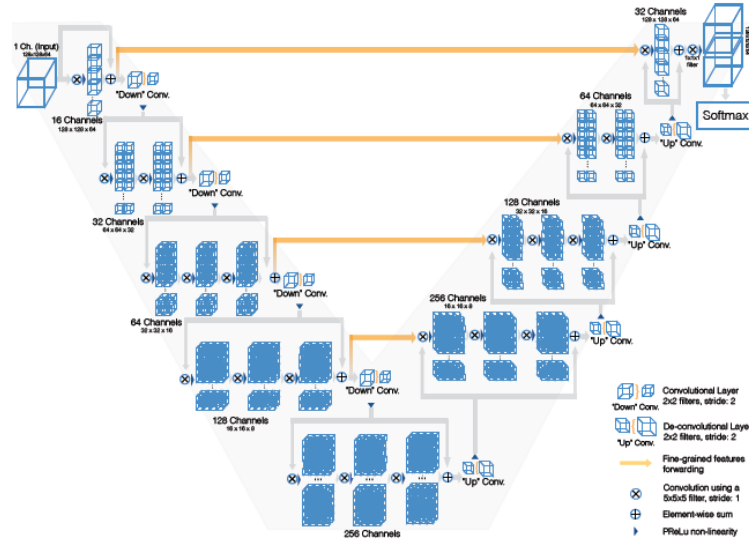


Figure 2.2 : Représentation de *V-Net*, tiré de [102]

Tout comme *U-Net*, *V-Net* ne nécessite pas beaucoup de données d'entraînement pour produire des résultats intéressants, et ce, même en 3 D. L'article original n'utilise que 50 images de résonance magnétique de taille 128x128x64 pour l'apprentissage. Il faut préciser que des techniques d'augmentation des données sont appliquées. *V-Net* atteint un *DSC* de 86 % sur une tâche de segmentation binaire, surpassant les autres modèles au moment de sa publication.

DeepLAB est l'un des modèles de segmentation sémantique les plus populaires [104]. Il partage plusieurs caractéristiques avec *U-Net*, *V-Net* et les autres modèles de segmentation sémantique telle que les *FCN*. Parmi les caractéristiques qui le distinguent des autres modèles, il y a l'utilisation de convolution *atrous*. Une convolution *atrous* est une convolution avec un espacement (stride) de taille définie au niveau de l'échantillonnage. Cela permet de contrôler la résolution à laquelle les réponses des caractéristiques sont calculées. *DeepLAB* se distingue également par l'utilisation de *atrous spatial pyramid pooling*, ce qui lui permet de segmenter de façon robuste des objets de taille diverse. Il est à noter que la version originale *DeepLAB* a été améliorée depuis sa publication. *DeepLABv3* est la version la plus à jour actuellement. Dans la troisième version, la normalisation par *batch* est ajoutée pour faciliter l'entraînement.

DenseNet est un *FCN* très profond développé originalement en 2017 (2D) [105]. *DenseNet* soulève un problème avec les *CNNs* traditionnels lorsqu'ils deviennent trop profonds et propose une architecture profonde résolvant ce problème. Lorsque les *CNNs* deviennent trop profonds, l'information des intrants subit une multitude de transformations à travers chacune des couches, ce qui peut avoir l'effet de faire disparaître cette information, résultant en problème de *vanishing*

gradient. *DenseNet* simplifie la connectivité entre les couches en connectant chacune des couches entre elles. Ce principe se nomme le *feature reuse* puisque le réseau apprend en exploitant les informations contenues dans les intrants à plusieurs reprises. Il en résulte un modèle moins complexe (moins de paramètres) qui offre des performances de segmentation élevées.

Bien que les *CNNs* et les *FCN* atteignent des performances intéressantes, ils possèdent généralement plusieurs millions de paramètres à entraîner et de nombreuses couches. Cela nécessite des heures d'entraînements et limite l'adaptabilité des réseaux plus profonds. Une solution possible pour réduire la durée des entraînements est de combiner un *CNN* avec un algorithme de partitionnement non supervisé. Le *CNN* est alors entraîné pour localiser la région d'intérêt et l'algorithme de partitionnement effectue la segmentation. Ce processus a été appliqué à la segmentation des vaisseaux sanguins à partir d'images échographiques du foie [106]. Cette recherche atteint une précision de 99,14 % au niveau de précision des pixels tout en maintenant le réseau à une taille modeste en utilisant seulement 132 images d'entraînement. L'efficacité de ce processus vient du fait que la tâche principale est effectuée par le *CNN*, qui localise la région d'intérêt, tandis que la segmentation est effectuée par un algorithme simple en dehors du réseau de *CNN*.

L'acquisition des données ultrasonores implique un réglage précis des paramètres des appareils d'acquisition pour augmenter la précision visuelle des données et les données résultantes peuvent contenir des artefacts et du bruit de fond. L'intégration de connaissances préalables (atlas) contribuerait à réduire les effets des artefacts et du bruit introduits par l'acquisition des données d'entrée. Dans [107], les modèles de segmentation ont appris une représentation 3D des propriétés anatomiques cardiaques à partir d'images de résonance magnétique et de scans ultrasonores. Combiné à une nouvelle procédure d'entraînement, il en résulte une performance accrue par rapport aux performances de référence.

AtlasNet utilise également les propriétés anatomiques en entraînant de multiples réseaux redondants (en parallèle) en fonction des configurations anatomiques [108]. L'idée est que des objets spécifiques, tels que le cœur ou des soudures, possèdent un nombre très limité de formes et qu'un réseau devrait donc exploiter cette situation de redondance. AtlasNet est composé de deux parties : un transformateur et un dé-transformateur. La partie transformatrice est composée d'un opérateur déformable et d'un réseau de segmentation. L'inverse de l'opérateur déformable est la partie dé-transformateur. Par rapport aux réseaux dans son domaine, *AtlasNet* les surpasse en atteignant un coefficient *dice* (*Dice Score Coefficient* : *DSC*) de 67,7 %. Il faut noter que les performances varient en fonction du nombre d'atlas.

Une autre façon d'exploiter la présence de forme récurrente est d'ajouter une forme préalable comme terme de régularisation dans la fonction de perte des *FCN*. Dans [109], un *prior* en forme d'étoile a été introduit dans la fonction de perte pour segmenter les lésions cutanées, pénalisant les points segmentés qui ne satisfont pas une forme convexe. L'ajout de la forme étoilée a permis de surpasser les modèles populaires tels que *U-Net* et *ResNet* de respectivement 5 % et 2 % sur le *DSC* dans la tâche de segmenter des points appartenant à des maladies pulmonaires interstitielles dans des images de tomodensitométrie.

Les *CNNs* traditionnellement utilisés en vision par ordinateur possèdent deux limitations importantes : ils ne tiennent pas compte du contexte et ont des difficultés avec les structures déformables [110]. En effet, une segmentation précise (par un analyste humain) requiert généralement des indices provenant du contexte. Ces indices correspondent, par exemple, à une position ou la présence d'un objet (artère par exemple). Les *CNNs* prédisent les classes des pixels de façon individuelles, ignorant les dépendances à plus longue portée. Les *CNNs* peuvent réduire certains éléments à du bruit/contexte (*background*). Ce genre de problème est le résultat de la réduction de la résolution des données qui arrive à chaque couche des *CNNs* standards.

Afin d'éviter ces deux problèmes, la littérature propose d'utiliser l'entraînement contradictoire (*adversarial*). Un *Generative Adversarial Networks* est un modèle génératif où deux réseaux sont placés en compétition dans un scénario de théorie des jeux [111]. Le premier réseau est le générateur, il génère un échantillon (par exemple une image ou une segmentation) basé sur les caractéristiques de l'ensemble des données d'entraînement, tandis que son adversaire, le discriminateur essaie de détecter si l'échantillon (par exemple l'image) est réel ou le résultat du générateur [112]. Les deux réseaux sont mis en compétition dans un scénario de théorie des jeux, où le générateur tente de synthétiser des échantillons plus réalistes pour tromper le réseau discriminateur qui lui apprend à distinguer entre les vrais et les faux échantillons. Les deux réseaux sont donc entraînés à partir du même ensemble de données [111]. Les méthodes de segmentation basées sur les *Generative Adversarial Networks* comprennent à la fois des méthodes supervisées et non supervisées [112]. Lorsque des données annotées sont disponibles, l'entraînement contradictoire permet d'améliorer les performances de segmentation des modèles génératifs, souvent des *FCN* tel que *U-Net*. Une expérience démontre que l'utilisation de l'entraînement contradictoire avec un *CNN* améliore la capacité des méthodes de segmentation sémantique tout en augmentant la performance des méthodes non sémantiques [113].

Dans [110], le réseau discriminateur évalue la qualité de la segmentation des images de nerfs et punit le réseau si la segmentation n'est pas bonne. L'entraînement du réseau de segmentation minimise la fonction de perte, dégradant ainsi les performances du discriminateur (*adversarial*). L'entraînement du discriminateur encourage le réseau de segmentation à produire des images qui

sont de mieux en mieux segmentées, les rendant difficiles à distinguer si elles sont réelles (provenant des images d'entraînement) ou créer de toutes pièces par le réseau d'attaque. Cette étude atteint un *IoU* moyen de 73,3 %, surpassant la compétition (meilleur modèle de référence : *CNN + Recurrent Networks*) de plus de 1 %.

L'entraînement contradictoire a été mis à profit à plusieurs occasions afin de segmenter de façon précise plusieurs types d'objets, souvent en imagerie médicale, tels que poumon [114], des tumeurs mammaires [115], le foie [116], le cancer de la prostate [117], le cerveau [113] et les nerfs [110]. Par exemple, *LGAN* utilise 220 images de tomodensitométrie seulement pour entraîner un *Generative Adversarial Networks* à segmenter des images de poumons. Plusieurs structures ont été testées et analysées permettant d'atteindre des performances de segmentation sémantique atteignant un *IoU* moyen de 92,25 %. Les résultats maximaux ont été obtenus à l'aide d'un *Generative Adversarial Networks* doublé d'un réseau de régression. La principale contribution consiste en la modification de la perte du discriminateur en une fonction de perte combinée définie comme étant $E[D(G(x))] - E[D(réel)]$.

Le Tableau 2.2 résume les études présentées dans cette section. Les performances des études présentées ne peuvent pas être comparées, car les ensembles de données utilisés pour l'évaluation sont différents.

Comme le montre le Tableau 2.2, la littérature est très peu abondante en ce qui concerne la segmentation des données ultrasonores de l'industrie END, telles que les défauts de soudure, en utilisant les récentes découvertes en IA. Cependant, la littérature inclut plusieurs exemples de segmentation d'objets complexes à partir de données ultrasonores ou de données obtenues par des techniques d'acquisition similaires. Bien que le réseau *U-Net* n'ait pas été appliqué directement dans l'industrie des END, il a été appliqué à une variété de tâches et en utilisant divers types d'entrée, ce qui démontre sa grande flexibilité.

Il est à noter qu'afin d'élargir la quantité d'études pouvant être considérées par cette recherche, le domaine de l'imagerie médicale a été inclus dans cette revue de littérature. Bien que les objets analysés soient complètement différents (organes, lésions et masses vs défauts et soudures), les outils utilisés dans ce domaine sont très similaires à ceux utilisés dans le domaine des END. En effet, l'imagerie médicale utilise les ondes ultrasonores [106, 118-120], l'imagerie par résonance magnétique [121-124], la tomodensitométrie [114, 125] et la radiographie par rayons X [126]. Il est donc possible de transposer les méthodes développées dans le domaine de l'imagerie médicale au domaine des END. Un aspect intéressant est que le domaine de l'imagerie médicale est largement plus actif que celui des END, résultant en un plus grand nombre d'études utilisant l'IA pour diagnostiquer des problèmes de santé à partir de données ultrasonores [4, 127, 128].

Tableau 2.2 : Résumé des études de segmentation supervisées

Études	Année de publication	Type d'intrants	Tâche de segmentation	Architecture	Performance (%)
[114]	2019	Images de tomodensitométrie	Poumons	Contradictoire	<i>IoU</i> moyen de 92
[96]	2015	Images	Structures neuronales	<i>U-Net</i>	<i>IoU</i> moyen de 92
[97]	2019	Images ultrasonores	Paroi artérielle	<i>U-Net</i>	<i>IoU</i> moyen de 87
[98]	2018	2D images et 3D Voxels	10 disciplines	<i>U-Net</i>	<i>DSC</i> le plus élevé sur plusieurs classes
[99]	2016	2D Tranches d'images volumétriques	Rein	<i>U-Net</i>	<i>IoU</i> moyen de 86
[100]	2019	Images de résonance magnétique	Cerveau	<i>U-Net</i>	<i>DSC</i> de 80 ± 1
[101]	2019	Images de résonance magnétique	Cerveau	<i>U-Net</i>	<i>DSC</i> de 73
[102]	2016	3D image de résonance magnétique	Prostate	<i>V-Net</i>	<i>DSC</i> de 86,9
[104]	2018	Images	Objets (Pascal VOC-2012)	<i>DeepLab</i>	<i>IoU</i> moyen de 79,7
[105]	2017	Images	Objets (ImageNet)	<i>DenseNet</i>	Erreur Top1 de 25,02
[106]	2018	Images ultrasonores	Régions vasculaires	<i>CNN</i>	Précision pixel de 99
[107]	2017	3D image de résonance magnétique 3D image ultrasonore	Cœur	<i>ACNN</i>	<i>DSC</i> de $93,9 \pm 1,7$
[108]	2018	Images de tomodensitométrie	Maladie pulmonaire interstitielle	<i>AtlasNet</i>	<i>DSC</i> de 67,7
[109]	2018	Images	Lésions de peau	<i>FCN</i> avec <i>prior</i>	<i>DSC</i> de 85,7
[110]	2018	Images ultrasonores	Nerfs	Contradictoire	<i>IoU</i> moyen de 73,3
[113]	2017	3D image de résonance magnétique	Cerveau	Contradictoire	<i>DSC</i> de 92 ± 3
[115]	2019	Images ultrasonores	Seins	Contradictoire	<i>DSC</i> de 93,8 <i>IoU</i> de 88,8
[116]	2017	3D image tomodensitométrie	Foie	Contradictoire (<i>U-Net</i>)	<i>DSC</i> de 95 ± 2 <i>DSC</i>
[129]	2019	3D images ultrasonores	Seins	<i>CNN</i>	<i>IoU</i> moyen de 85

2.2.2.2 Segmentation à partir d'algorithmes non supervisés

Étant donné qu'il est difficile d'annoter des scans ultrasonores de soudures, et considérant la structure des soudures et la nature de l'acquisition des données *PAUT*, les méthodes d'apprentissage non supervisées peuvent s'avérer très intéressantes pour réaliser la segmentation des soudures [9, 90]. C'est pourquoi, dans cette section, les méthodes d'apprentissage non supervisées qui ont fait leurs preuves sont présentées, telles que les auto-encodeurs et les *Generative Adversarial Networks* [94, 95].

L'apprentissage par renforcement (*Reinforcement learning*) implique l'apprentissage en interagissant avec un environnement, ce qui permet à l'agent artificiel d'apprendre des tâches complexes qui pourraient prendre plusieurs étapes avant d'atteindre la solution [130, 131]. Dans le domaine de l'imagerie médicale, l'objectif est de détecter un point d'intérêt, soit une lésion, un organe ou un repère anatomique, à travers un ensemble de données à l'aide d'une séquence de décisions markovienne qui permet à l'agent de naviguer dans les données [125, 132]. L'agent artificiel apprend donc à identifier le chemin optimal vers le point d'intérêt en interagissant avec un environnement 3D. L'agent artificiel peut exécuter six différentes actions pour se diriger vers l'objectif, soit se diriger vers : la droite, la gauche, droit devant, derrière, en haut et en bas. L'agent artificiel réussit à trouver les repères anatomiques avec une efficacité d'environ trois millimètres (2,37 à 6,29 mm dépendamment du repère).

Un auto-encodeur est un modèle en deux parties qui encode les intrants dans une représentation latente, une version compressée des intrants, et qui reconstruit la représentation latente dans son format d'origine [1]. Lors de ces transformations, les réseaux tentent de minimiser les erreurs de reconstruction entre les intrants et la reconstruction [3]. L'auto-encodeur apprend donc la répartition des données brutes. Les auto-encodeurs sont entraînés à partir d'images sans défauts. Lorsqu'ils sont confrontés à des images comprenant des défauts, ils ne sont plus en mesure de reconstruire l'image correctement due aux défauts. Une comparaison entre l'image reconstruite et l'image initiale permet de détecter et segmenter les défauts. Les auto-encodeurs ne nécessitent pas de données annotées pour l'entraînement, ce qui constitue un avantage intéressant dans les situations où l'annotation est difficile. La Figure 2.3 montre l'architecture générale d'un auto-encodeur.

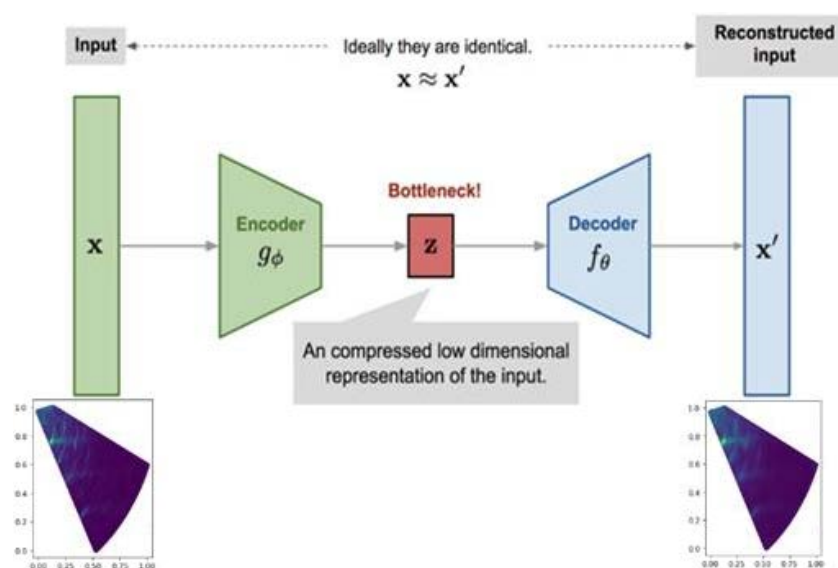


Figure 2.3 : Architecture d'un auto-encodeur, adapté de [133]

Afin de minimiser l'impact du bruit sur les résultats de classification, un auto-encodeur a été mis à l'essai afin de développer un modèle robuste au bruit [85]. Trois essais ont été réalisés, l'entraînement d'un *CNN* avec des données bruitées, l'entraînement d'un auto-encodeur avec des données sans bruit et l'entraînement d'un auto-encodeur avec des données bruitées. Les résultats de ces essais confirment l'efficacité de l'auto-encodeur. En effet, l'auto-encodeur obtient les meilleures performances de classification des défauts parmi les trois essais.

Les auto-encodeurs convolutifs (*Convolutional Auto-Encoder*) ont été appliqués avec succès à la tâche de segmentation des défauts dans les images [95]. Les auto-encodeurs convolutifs ont des capacités de traitement des images supérieures que les auto-encodeurs traditionnels [72, 80]. En utilisant un ensemble de données contenant 200 images de nanofibres, un auto-encodeur convolutifs atteint une aire sous courbe (*AUC*) de 96,6 % [94]. Les résultats optimaux sont obtenus en utilisant *Structural Similarity Index Measure* (*SSIM*) comme fonction de perte au lieu de la distance ℓ^p , un espace latent de 500 et un *batch size* de 128. La distance ℓ^p Est une fonction de perte par pixel qui suppose que les valeurs des pixels voisins sont indépendantes les unes des autres tandis que les interdépendances des régions de patches locales sont prises en compte avec les *SSIM*. Le passage de la distance ℓ^p à *SSIM* entraîne une augmentation des performances (*AUC* de 68,7 % à 96,6 %) pour la segmentation des images. Il est à noter qu'une version améliorée du *SSIM*, appelée *Multi-Scale Structural Similarity Index Measure* (*MSSSIM*) a été proposée, qui surpasse le *SSIM* traditionnel [134]. *MSSSIM* utilise différentes tailles de fenêtre pour capturer des détails les plus fins aux objets globaux des images. Toutefois, il n'y a pas encore d'utilisation combinée de *MSSSIM* et d'un auto-encodeur convolutifs en END.

Diverses améliorations aux auto-encodeurs traditionnels sont proposées dans [135]. Premièrement, l'utilisation de la fonction de perte de perceptuel est proposée en remplacement de la fonction de perte traditionnelle basée sur la distance ℓ^p . Comme pour les *SSIM*, cette étude décrit comment les mesures de distance basée sur la distance ℓ^p manque de contexte puisque ces mesures se concentrent sur les pixels individuellement. La fonction de perte *perceptuel* correspond à la différence entre les caractéristiques profondes des images des intrants x et les images reconstruites $\hat{x}$. L'idée derrière l'utilisation de la fonction de perte *perceptuel* est pour stocker les caractéristiques des images saines (sans défaut). Par conséquent, l'auto-encodeur apprend à connaître la distribution des images saines, ce qui lui permet de comprendre les données d'entraînement. Pour augmenter l'efficacité de ce processus, l'utilisation de la croissance progressive est proposée, qui consiste à démarrer l'entraînement à partir d'un réseau basse résolution et à ajouter des couches progressivement. Enfin, leur troisième modification par rapport à l'auto-encodeur traditionnel consiste à utiliser un très petit nombre de données annotées pour définir les valeurs optimales des hyperparamètres, ce qui

augmente la précision de détection de la *AUC ROC* de 1 % à 5 % selon les données utilisées. Bien que cette approche ne soit pas entièrement non supervisée, elle n'implique pas d'hypothèses déraisonnables puisque, souvent, quelques exemples peuvent facilement être annotés. Cette étude atteint une précision de détection de la *ROC AUC* de 93,4 % pour la détection des ganglions lymphatiques, dépassant de 2,8 % les meilleurs performances précédentes.

En utilisant une méthode basée sur la similarité par région et un *CNN*, il est possible de détecter et de localiser les défauts dans les images de microscopie électronique à balayage des matériaux composites [43]. Cette méthode débute par la construction d'un dictionnaire des régions sans défauts (régions normales). Le dictionnaire est construit en extrayant les caractéristiques des régions des images d'entraînement à l'aide d'un *CNN*, puis en appliquant l'analyse en composantes principales pour réduire la dimensionnalité des vecteurs de caractéristiques. *K-Means* est également appliqué dans la construction du dictionnaire. La méthode peut alors mesurer à quel point une sous-régression est une anomalie par rapport au dictionnaire des régions normales. Le degré d'anomalie est mesuré comme étant la similarité visuelle entre une sous-région donnée et une sous-région normale. La similarité visuelle correspond à la distance euclidienne entre les vecteurs caractéristiques extraits des deux sous-régions. Un seuil est utilisé pour classer la sous-région donnée en une région normale ou anormale. Lors de la prédiction, l'image entrante est divisée en sous-régions et les sous-régions sont ensuite comparées au dictionnaire des régions normales. Cette méthode innovante permet d'atteindre une *AUC* de 97 % sur les grandes images.

Dans [136], la capacité de deux architectures convolutives profondes est évaluée pour reconstruire un crâne de fœtus à partir de scans ultrasonores 3D incomplets. Un auto-encodeur convolutifs et un auto-encodeur conditionnel variationnel sont analysés. Un auto-encodeur conditionnel variationnel est une combinaison d'inférence bayésienne probabiliste et d'apprentissage profond pour la représentation conditionnelle des multiples. Il s'agit d'un modèle génératif qui approxime la probabilité conditionnelle $P(Y_i|X_i)$, où Y_i et X_i représentent l'objectif (crâne complet) et la condition variable (crâne incomplet). Dans ce type de réseau, le modèle génératif conditionné peut être représenté comme $Y_i = G(X_i, z)$, où la fonction de cartographie G est mise en œuvre sous la forme d'un RDN. L'idée derrière un auto-encodeur conditionnel variationnel est aussi d'apprendre à échantillonner les valeurs de z qui sont susceptibles de produire Y_i . Les deux modèles ont été entraînés à partir de 107 échantillons et évalués sur 26 échantillons de test. Le modèle génératif permet d'obtenir un *DSC* de 86 %.

Dans [110], les auteurs ont souligné les limites des *CNNs* lors de la segmentation des nerfs à partir d'images ultrasonores. Les *CNNs* prédisent la classe des pixels séparément les uns des autres et ignorent donc les dépendances anatomiques de plus haut niveau. Les *CNNs* éprouvent des difficultés également contre les structures déformables, ce qui entraîne une mauvaise classification d'objets

déformables. Par exemple, les nerfs sont parfois confondus en bruit de fond. Ceci est causé par les couches de convolution qui réduisent la résolution de l'image à chaque couche. Pour surmonter ces problèmes, un *Generative Adversarial Networks* composé d'un réseau de segmentation et d'un réseau discriminateur est proposé. Le réseau de segmentation est un *CNN* profond avec une convolution dilatée. La convolution dilatée augmente la taille du champ de réception sans perte de résolution. L'expérience surpasse de plus de 1 % (*mean IoU*) la performance de segmentation de référence sur les nerfs, ce qui prouve une fois de plus la puissance de l'entraînement contradictoire.

La combinaison de l'architecture *Least Squares Generative Adversarial Networks* et de quatre fonctions de pertes différentes (*adversarial*, reconstruction, région et délimitée) génère une segmentation très précise des images cardiaques [137]. Leur expérience a été réalisée en utilisant des scans ultrasonores et des images de résonance magnétique comme intrants. Le réseau discriminateur évalue la qualité de la segmentation et pénalise le réseau si la segmentation n'est pas bonne. L'entraînement du réseau de segmentation minimise la fonction de perte, ce qui dégrade les performances du réseau générateur. Leur modèle est similaire à un auto-encodeur, le réseau de segmentation étant l'encodeur qui est suivi par un réseau de reconstruction. L'entraînement du réseau générateur encourage le réseau à produire une segmentation de meilleure qualité, ce qui rend difficile de distinguer si elles sont réelles (à partir d'images d'entraînement) ou créer de toutes pièces par le réseau.

La détection des anomalies correspond au développement d'un modèle d'apprentissage profond qui vise à détecter les objets anormaux dans les images en apprenant la distribution des données normales, permettant ainsi aux modèles de mettre en évidence les anomalies avec précision [138, 139]. Le principal avantage de la détection d'anomalies est qu'elle ne nécessite pas de données annotées puisqu'elle est basée sur le principe de détecter les patterns inconnus. Les données d'anomalies sont généralement limitées par leur nature, car de nombreuses anomalies, comme les défauts de soudure, se produisent rarement. C'est pourquoi les efforts nécessaires pour obtenir suffisamment de données anormales et pour les étiqueter sont importants. La détection d'anomalies utilise une grande quantité de données saines non annotées pour entraîner un réseau qui apprend la distribution normale des données, ce qui lui permet de détecter les anomalies lorsqu'il en rencontre une.

Le modèle *F-anoGAN* combine le principe de détection des anomalies et la grande capacité des *Generative Adversarial Networks* [138]. *F-anoGAN*, qui signifie *fast anomaly detector using Generative Adversarial Networks*, permet d'obtenir une grande précision de segmentation tout en ayant un temps d'inférence rapide. La structure de *F-anoGAN* est illustrée à la Figure 2.4. *F-anoGAN* se compose d'un encodeur, qui apprend la correspondance des images avec l'espace latent, d'un générateur qui reconstruit les images à partir de l'espace latent et d'un discriminateur qui classe les

images comme réelles ou reconstruites. Le générateur apprend à générer des images correspondant à la distribution fournie par l'entraînement en saisissant la variabilité normale. Le discriminateur peut estimer l'ajustement des images générées à la distribution des images d'entraînement. Ce processus est basé sur le modèle de *Wasserstein Generative Adversarial Networks (WGAN)* [140]. Les anomalies sont obtenues en combinant les erreurs de reconstruction et l'erreur résiduelle de la caractéristique discriminante. Des scores d'anomalies élevés sont attendus des anomalies. *F-anoGAN* atteint 80 % sur le score f-1 et 93 % sur la *AUC*, surpassant ainsi les autres détecteurs d'anomalies.

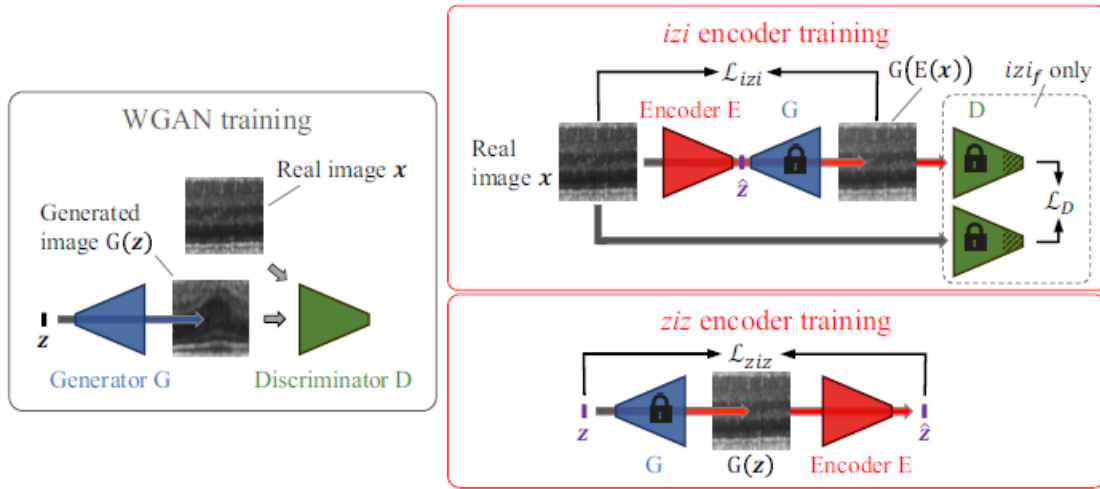


Figure 2.4 : Représentation visuelle de *F-anoGAN*, tiré de [138]

Un modèle profond combinant les structures d'un auto-encodeur variationnel et d'un *Generative Adversarial Networks*, nommé *AnoVAEGAN*, est proposé pour la segmentation des anomalies cérébrales à l'aide d'une métrique par pixel basée sur la distance ℓ^1 [141]. Les modèles d'auto-encodeurs spatiaux peuvent être utilisés pour capturer les variabilités anatomiques normales du cerveau à partir d'images de résonance magnétique. La qualité de la reconstruction est améliorée en exigeant des données d'entrée alignées et en ajoutant la fonction de perte contradictoire pour améliorer la qualité visuelle des images reconstruites. En utilisant le principe des auto-encodeurs, le réseau apprend à cartographier les tissus cérébraux sains et peut donc détecter, délimiter et segmenter les pathologies comme des déviations de la norme. D'autres détecteurs d'anomalies sont construits sur des principes similaires, augmentant à chaque fois les performances de détection et de segmentation [142, 143].

La plupart des réseaux contradictoires présentés jusqu'à présent utilisent des intrants en 2D et génèrent des images en 2D. Une version 3D du *Generative Adversarial Networks* est proposée dans des études plus récentes [114, 116]. Cependant, l'entraînement d'un réseau générateur 3D nécessite

davantage de données et la durée de l'entraînement est significativement augmentée. Par exemple, une étude propose une segmentation 3D du foie à partir de scans de tomodensitométrie en utilisant un *Generative Adversarial Networks* [116]. Le réseau générateur utilise une architecture *U-Net* pour segmenter des scans de tomodensitométrie 3D de l'abdomen. Il atteint une performance de 95 % en *DSC*, surpassant les méthodes *CNNs* traditionnelles (92 %) en termes de performance et d'efficacité de calcul.

Le Tableau 2.3 résume les études présentées dans cette section, avec leurs performances respectives. Comme on peut le constater, les études portant sur la segmentation des défauts de soudure sont très peu nombreuses. En revanche, les études utilisant des *Generative Adversarial Networks* pour segmenter des objets complexes sont nombreuses. Les *Generative Adversarial Networks* ont été appliqués avec succès à la segmentation du cerveau, du cœur, des poumons et de la rétine [113, 114, 138, 144]. Son efficacité réside dans sa structure qui permet l'entraînement en parallèle des réseaux générateurs et de discriminateurs.

Tableau 2.3 : Résumé des études de segmentation non supervisées

Études	Année de publication	Type d'intrants	Tâche de segmentation	Architecture	Performance (%)
[43]	2018	Images	Défauts	<i>CNN-based self-similarity</i>	<i>DSC</i> de 97
[94]	2018	Images	Défauts	Auto-encodeur convolutifs	97 (<i>AUC</i>)
[95]	2019	Images	Défauts	Auto-encodeur convolutifs	Non disponible
[114]	2019	3D image tomodensitométrie	Poumon	<i>Generative Adversarial Networks</i>	<i>DSC</i> de $98,5 \pm 3$
[141]	2018	Images de résonance magnétique	Cerveau	AnoVAEGAN	<i>DSC</i> de $60,5 \pm 19,3$
[135]	2020	Images radiographiques	Ganglions lymphatiques	<i>Perceptual auto-encoder</i>	93.4 (<i>AUC</i>)
[136]	2018	3D scans ultrasonore	Reconstruction du crâne du fœtus	Auto-encodeur convolutifs	<i>DSC</i> de 86 ± 4
[137]	2018	Tomodensitométrie et Images de résonance magnétique	Cœur entier	<i>Generative Adversarial Networks</i>	<i>DSC</i> de 66 et 51
[138]	2019	Images oculaires	Rétine	<i>F-anogan</i>	79.8 f-score 93.0 <i>AUC</i>
[142]	2019	Images radiographiques	Contrôle de sécurité	<i>Skip-GANOMALY</i>	94 <i>AUC</i> moyen
[144]	2018	3D scans ultrasonore	Atrium gauche	Contradictoire	<i>DSC</i> de 63
[145]	2019	3D images de résonance magnétique	Lésions cérébrales	Auto-encodeur convolutifs	<i>DSC</i> de $76,6 \pm 11,4$

[146]	2018	Images ultrasonores	Détection de points de repère de la prostate	<i>Generative Adversarial Networks</i>	DSC de 92 ± 3
-------	------	---------------------	--	--	-------------------

On constate qu'il existe des modèles de segmentation basés sur l'apprentissage non supervisé atteignant des performances de haute qualité. Ce genre de solutions offre l'avantage de ne pas nécessiter de données annotées, ce qui facilite le processus d'entraînement. En revanche, la qualité des solutions dépend entièrement de la sélection de données d'entraînement dites normales qui adoptent fidèlement la distribution des données utilisées au niveau de l'inférence future.

Il est à noter que le Tableau 2.3 n'est pas exhaustif. Il existe beaucoup d'autres applications intéressantes de *Generative Adversarial Networks* pour détection d'anomalies [142, 143, 147-155]. Comme il serait impossible de les présenter toutes, on en retrouve simplement ici une sélection judicieuse.

2.3 Modélisation géométrique en trois dimensions

2.3.1 Principes de base

La section précédente présentait les études dans le domaine des END ayant eu recours à des modèles d'IA pour classer et détecter divers types d'objets d'intérêt. Une seule étude recensée a utilisé les résultats de son modèle d'IA pour modéliser l'objet d'intérêt en 3D [10]. Dans ce cas précis, des défauts de délamination, une fois segmentés, sont ensuite représentés en trois dimensions à l'aide d'un ajustement de surface lisse pour chaque classe de défauts.

La plupart des recherches en visualisation médicale se sont concentrées sur des images ultrasonores 2D, qui peuvent être représentées sous forme de séquences d'images 2D, et lorsqu'elles sont empilées sous forme d'images, appelées *slices*, elles peuvent fournir une imagerie volumétrique 3D [156]. L'imagerie volumétrique 3D est donc une série de tranches 2D. Seulement quelques études se sont concentrées sur la visualisation en 3D de données ultrasonores (Voxel) [6, 157, 158]. Les articles qui consistent en des revues de littérature confirment qu'il existe des études qui font l'analyse des données en 3D, mais ce courant n'est que très récent [3, 159, 160].

Pour visualiser les données en 3D, il existe trois types de familles de rendus (*rendering*), à savoir : les techniques de reformatage multi-planaire, les techniques de rendues de surface, et les techniques de rendues de volume [6]. Les techniques de reformatage multi-planaire sont particulièrement efficaces pour représenter la colonne vertébrale à partir d'images ultrasonores. Elles utilisent trois différents plans de la colonne (*traverse*, *coronal* et *sagittal*) pour effectuer la représentation. Il s'agit d'une des techniques de rendus les plus simples et les plus efficaces en termes algorithmique. En revanche, elle souffre de perte d'informations lorsque les formes rendues ont plus de courbures

(*spinal curvature*). Cette approche semble difficilement applicable aux données ultrasonores du domaine des END. Les techniques de rendues de surface produisent une surface 3D basée sur les points segmentés en générant les triangles de surface ou les polygones associés aux techniques de rendue de surface standard fournie par interpolation [6]. Il y a deux types de rendus de surface : direct et indirect. Le rendu direct de surface est un cas particulier où la surface est rendue directement à partir du volume sans représentation géométrique intermédiaire, sans seuils de réglage ou en utilisant des étiquettes d'objet pour définir une plage d'intensités de voxels à afficher. La transparence et les couleurs sont utilisées pour une meilleure visualisation en 3D du volume. Les méthodes directes permettent de visualiser les surfaces directement à partir de leurs représentations. Celles-ci incluent *Ray-Casting*, les méthodes basées sur les particules, les techniques de rendues non réalistes et le rendu de volume.

Parmi les techniques utilisées dans le domaine de l'imagerie médicale, *Ray-Casting* est l'approche de visualisation la plus précise, mais la plus intensive à calculer [161]. Il s'agit d'une technique de rendu de volume basée sur une image. *Ray-Casting* calcule des images 2D à partir d'ensembles de données volumétriques 3D. Dans cette technique, un rayon est généré pour chaque pixel d'image souhaité. À l'aide d'un modèle de caméra synthétique simple, le rayon commence au centre de projection de la caméra et traverse le pixel de l'image sur le plan de l'image imaginaire flottant entre la caméra et le volume à restituer. Le changement de lumière qui traverse les données de volume 3D est projeté en tant que résultat de visualisation que l'opérateur peut visualiser. Le principe d'absorption de la lumière est mis en œuvre dans la technique de rendu de volume où chaque voxel possède les attributs tels que la luminosité, la transparence et la couleur. *Ray-Casting* nécessite beaucoup de ressources de la part du CPU. Plusieurs approches sont utilisées pour la visualisation, par exemple la projection par intensité maximale, la projection par intensité minimale et le rendu translucide. Bien que *Ray-Casting* soit une technique de visualisation directe qui produit les représentations de surfaces implicites les plus fidèles, c'est la technique la plus lente et, en général, n'est pas approprié pour une visualisation interactive rapide.

Les méthodes indirectes reposent sur la polygonisation pour créer une représentation discrète qui peut ensuite être utilisée à d'autres fins, telles que de la simulation ou de la modélisation. Tout comme le rendu de surface direct, le rendu de surface indirect nécessite que les points dans le volume (appartenant à la structure) soient identifiés à priori par segmentation. Des exemples de rendu de surface indirect sont le filtrage de contour (*contour filtering*) et *Marching Cubes*. Le filtrage des contours détermine la manière dont les contours de deux tranches successives doivent être connectés, les sommets des contours attribués devant être connectés pour former un maillage triangulaire.

Marching cubes semble être l'algorithme de construction 3D le plus populaire en médecine [6, 156]. Son intérêt vient du fait qu'il est très rapide et simple à utiliser par l'utilisation de *lookup table*. Plusieurs études ont amélioré l'algorithme de base depuis son lancement en 1987. L'algorithme utilise une méthode de *divide-and-conquer* dans laquelle le volume 3D est divisé en plusieurs cubes de voxels formant un tableau de voxels. Chaque cube est constitué de huit sommets, ce qui représente une valeur de voxel à partir des données de volume. Un paramètre spécifique à l'utilisateur, appelé *iso value*, est défini avant la reconstruction afin de créer une surface, également appelée *iso-surface*, en déterminant son intersection avec le cube. Le processus de *marching cubes* est déplacé de cube en cube en suivant l'ordre de gauche à droite, d'avant en arrière et de haut en bas jusqu'à la fin de l'algorithme.

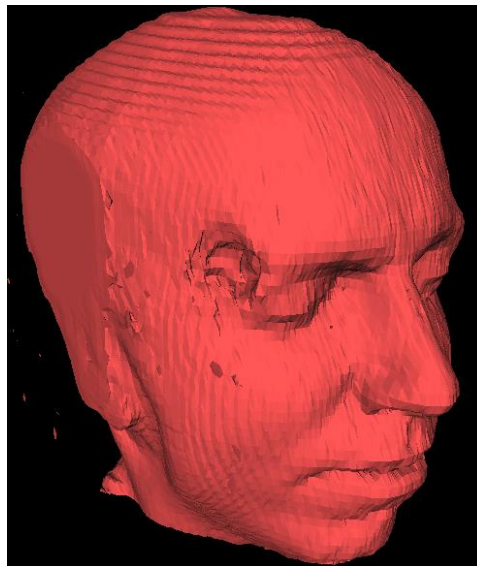


Figure 2.5 : Reconstruction par *marching cubes* d'une tête à partir de 150 images de résonance magnétique , tiré de Wikipédia : *Marching cubes*

Par rapport aux autres techniques, le maillage est à la fois rapide et facile à calculer [161]. La polygonisation des surfaces implicites nécessite la conversion d'une description mathématique continue en une approximation linéaire par morceaux discrets. Ce processus est un processus à perte, car la précision dépend de la fréquence d'échantillonnage, c'est-à-dire la taille des triangles. Ce processus doit être ajusté pour approximer fidèlement les caractéristiques de la surface et nécessite une mise à jour chaque fois que la surface implicite change. Néanmoins, les approximations polygonales permettent d'obtenir un excellent compromis entre qualité de représentation et performances interactives. La création de maillages sans ces artefacts nécessite des techniques axées sur la qualité et la régularisation des maillages. La qualité est particulièrement préoccupante lorsque le maillage doit être utilisé pour d'autres applications telles que la simulation et

la modélisation. La qualité d'un maillage peut être visualisée de deux manières : en créant des maillages réguliers ou adaptatifs. Les mailles régulières doivent avoir des triangles de tailles similaires et où les sommets ont des valences similaires. La Figure 2.6 montre un exemple de maillage à partir d'une surface implicite.

Les maillages adaptatifs utilisent des triangles dont la taille et la densité peuvent être adaptées pour refléter les propriétés de la surface, c'est-à-dire des petits triangles dans les zones de forte courbure et des triangles plus grands dans des régions plus plates. Les maillages sont décrits comme adaptatifs lorsque des triangles sont créés par rapport aux caractéristiques de la surface. La taille et la densité des polygones sont liées aux propriétés de la surface telles que la courbure ou les caractéristiques. Dans les méthodes de polygonisation adaptative, il existe un plus grand nombre de triangles dans les régions à forte courbure ou en présence de traits nets. Inversement, dans les régions plus plates, les triangles sont plus grands et moins nombreux. Il y a deux types d'approches : celles basées sur l'adaptation de maillages existants, puis celles conçues pour créer de nouvelles polygonisations adaptatives directement à partir de la surface.

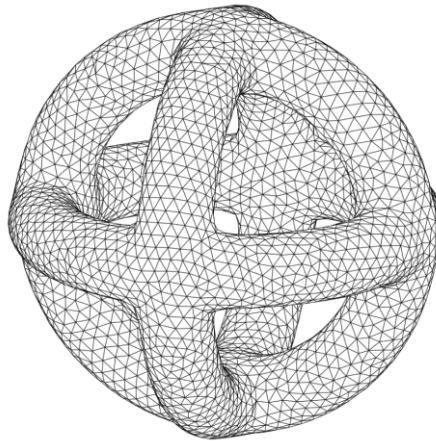


Figure 2.6 : Maillage généré à partir d'une surface implicite, tiré de Wikipédia : *Mesh generation*

La reconstruction de Poisson filtrée (*Screened Poisson Reconstruction : SPR*) est la méthode la plus couramment utilisée pour convertir un nuage de points non structuré, avec ses normales par point, en un maillage de surface [162]. *SPR* est considérée comme la méthode de reconstruction de nuage de points non basée sur les données la plus efficace à l'heure actuelle. Il s'agit d'une méthode générale de reconstruction de surface n'utilisant aucun *prior* qui produit une surface lisse. Toutefois, *SPR* peut incorrectement fermer des trous lorsque les données sont bruitées ou non uniformes. *SPR* est également susceptible de dégénérer lorsque les estimations de normales par point sont erronées.

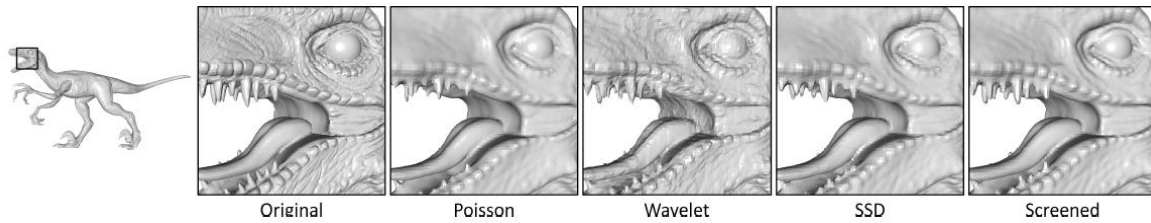


Figure 2.7 : Reconstruction par différentes techniques, dont *SPR*, d'un modèle de raptor, tiré de [162]

En résumé, le maillage par polygonisation est une excellente façon de représenter des formes en trois dimensions. Cela capture explicitement à la fois la surface de la forme et la topologie, et exploite la non-uniformité pour représenter de grandes régions plates ainsi que des éléments nets et complexes. *SPR* est la méthode qui consensuellement produit le meilleur maillage de surface et devrait.

2.3.2 Modélisation en trois dimensions à l'aide de l'intelligence artificielle

Il existe deux approches au problème de reconstruction 3D à partir de l'IA : les méthodes basées sur les voxels [163] et celles basées sur les nuages de points [87, 164]. La première approche produit des représentations en format voxels permettant de connaître par exemple le volume d'un objet. La seconde approche produit un nombre fixe de points comme extrant. Ces approches nécessitent toutefois des étapes de post-traitement, *Marching Cubes* ou *B-Splines* par exemple, pour produire une modélisation 3D finale. Ces approches intermédiaires ont toutefois deux limites majeures : dans la qualité des modélisations qui sont générées et dans le processus d'entraînement. En effet, pour obtenir une représentation intermédiaire de haute qualité, les modèles utilisent une fonction de perte intermédiaire (Chanfrein ou distance ℓ_1) qui tente de maximiser la qualité de la représentation intermédiaire. Or, cette représentation intermédiaire n'est qu'un résultat temporaire requérant une étape supplémentaire pour obtenir une surface. La fonction de perte ne peut pas jouer directement sur la qualité de la surface finale, ce qui limite la modélisation finale.

La seconde limitation provient du processus d'entraînement de ces modèles. En effet, afin d'obtenir une surface 3D, il est nécessaire de produire un produit intermédiaire (un objectif intermédiaire et temporaire), en plus de produire l'objet final désiré. Il est donc nécessaire d'obtenir des données d'entraînement annotées du produit intermédiaire qui n'ont, en soi, aucune valeur.

Le processus standard de modélisation de surface 3D à partir de données brutes, telles que voxels, nuage de points et imagerie par résonance magnétique, ne peut pas être entraîné de bout en bout (*end to end*) puisque ce processus produit généralement une solution temporaire intermédiaire desquels un maillage surfacique tridimensionnel est produit [165]. Certaines solutions émergentes

en modélisation ont permis d'automatiser davantage ce processus en intégrant de nouvelles couches aux *CNNs* afin de produire des objets maillés 3D de grande qualité comme extrants.

MeshCNN utilise les propriétés intéressantes du maillage pour créer des représentations 3D de haute qualité [166]. *MeshCNN* utilise des convolutions et opérations de *pooling* spécialisées pour effectuer les opérations nécessaires à la réalisation de maillage triangulaire. *MeshCNN* fonctionne sensiblement comme que les *CNNs* traditionnels, à savoir qu'il peut être entraîné à classifier, détecter ou segmenter ces intrants. Il est alors en mesure de classifier différents types d'objets 3D, surpassant *PointNet* par exemple, et même de segmenter les parties de corps humains. *MeshNet* est un *mesh neural network* qui apprend la représentation des formes 3D à partir de données de maillage [167]. Tout comme *MeshCNN*, cela permet de procéder aux tâches traditionnelles des *CNNs* soit la classification et la segmentation.

Deep Marching Cubes prédit la représentation de surface explicite de topologie arbitraire [165]. Il s'agit d'un modèle bout-en-bout qui produit directement une représentation de surface explicite et optimise une fonction de perte géométrique. Ceci permet d'éviter le recours à une fonction de perte auxiliaire ou de transformer le maillage souhaité en produit implicite intermédiaire. Cela résulte en une meilleure modélisation 3D, plus robuste aux bruits et aux informations manquantes. Le type d'intrant de *Deep Marching Cubes* est un nuage de points non structuré. Le réseau prend le nuage de point P et le convertit en représentation volumétrique à l'aide d'un *grid pooling*. L'opération de *grid pooling* prend en intrant un ensemble de K points avec $D = 16$ *dimensional features maps* et effectue une mise en commun maximale (*max pooling*) dans chaque cellule. Les cellules vides sont associées au vecteur zéro. Les fonctionnalités mises en commun sont traitées par un réseau encodeur-décodeur avec des *skips-connections*. Le décodeur a deux têtes : une pour les occupations O et un pour les déplacements de sommets X . L'architecture est illustrée à la Figure 2.8.

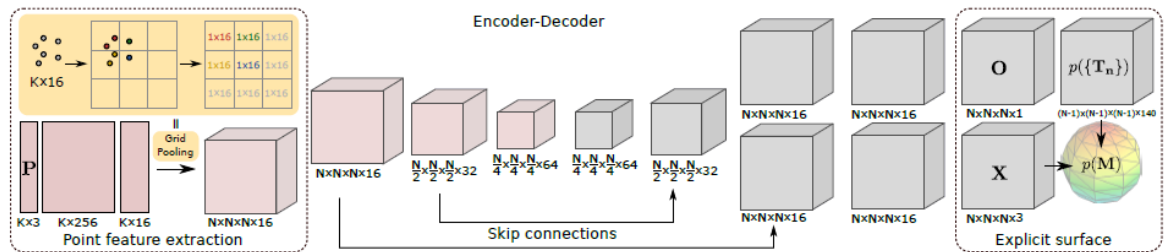


Figure 2.8 : Architecture de *Deep Marching Cubes*, tiré de [165]

Au cours de l'entraînement de *Deep Marching Cubes*, le but du modèle est de minimiser la distance entre la valeur ciblée du nuage de point et le maillage de surface prédit. Pour ce faire, le modèle a recours à une fonction de perte composée de perte géométrique, de perte d'occupation, de perte de

douceur et de perte de courbure (*geometric loss*, *occupancy loss*, *smoothness loss* et *curvature loss*). *Deep Marching Cubes* est en mesure de surpasser les études concurrentes, devenant la référence pour la modélisation de surface en 3D. La Figure 2.9 présente des résultats visuels de prédictions de *Deep Marching Cubes* à partir d'un nuage de points et les compare au *GT*.



Figure 2.9 : Résultats visuels de *Deep Marching Cubes* (DMC) à partir d'un nuage de points, adapté de [165]

Image2Mesh développe un processus de construction de maillage de surface en 3D à partir d'images individuelles [168]. Le processus débute avec une image qui est donnée à un auto-encodeur convolutif, permettant d'extraire un espace latent z qui est utilisé pour prédire la classe de l'objet dans l'image. La classe prédite est alors régressée en une paramétrisation de forme compacte à l'aide d'un réseau *feedforward*. Finalement, un *graph embedding* G qui représente de manière compacte des objets maillés 3D est utilisé pour reconstruire le modèle 3D. En détail, l'index de la classe est utilisé pour sélectionner le modèle 3D le plus près de l'image. Le modèle sélectionné est ensuite déformé pour correspondre aux paramètres estimés. Il en résulte un modèle capable de construire un modèle 3D préservant les aspects géométriques importants. Il est important de noter que les images fournies au modèle doivent être absentes de bruits pour réussir la reconstruction. *Neural 3D*

Mesh Renderer effectue un travail similaire à *Image2Mesh* [169]. Ce dernier intègre toutefois de nouvelles fonctionnalités comme le transfert de style.

Mesh R-CNN prend des images comme intrants, prédit le type d'objet dans l'image et construit sa forme 3D [170]. Il s'agit de la première tentative de joindre la perception en deux dimensions à des représentations en trois dimensions. Pour réussir ce processus, une branche de prédiction du maillage est ajoutée à *Mask R-CNN*. Le réseau prédit tout d'abord une représentation grossière en voxels qui est convertie en maillage et affinée avec un réseau de convolution de graphes opérant sur les sommets et les arêtes du maillage. Ce processus fort intéressant considérant qu'il utilise des images 2D en intrant produit toutefois des maillages 3D de qualité inférieure à *Deep Marching Cubes*.

Scan2Mesh produit un maillage 3D structuré à partir de scans 3D incomplet (*truncated signed distance field*), traduit en 5 canaux : les deux premiers stockent les valeurs de champs de distance non signées tronquées et l'espace connu/inconnu en fonction de la trajectoire de la caméra du scan tandis que les trois derniers canaux stockent les coordonnées (x, y, z) de la grille volumétrique du système de coordonnées du maillage [171]. *Scan2Mesh* propose une nouvelle formulation de *graph neural network* pour prédire la structure du maillage (vertices, arrêtes et faces).

Point2Mesh prend un nuage en intrant et reconstruit un maillage de surface 3D [172]. En fait, *Point2Mesh* débute avec un nuage de point et un maillage déformable puis rétracte (*shrink-wrap*) itérativement le maillage déformable jusqu'à obtenir une reconstruction *watertight*. Ainsi, les poids du modèle sont optimisés pour déformer un maillage déformable à l'aide d'actions *shrink-wrap* répétitives. Cette architecture utilise un *self-prior* qui encapsule les géométries répétitives d'une seule forme dans les poids d'un RDN profond. La clé du *self-prior* est la structure de partage des poids du CNN qui modélise intrinsèquement les structures récurrentes et corrélées et, par conséquent, est difficilement capable de modéliser du bruit et des valeurs aberrantes, qui ont des géométries non récurrentes. Ce principe se base sur l'hypothèse que les formes naturelles se répètent régulièrement et sont corrélés, tant au niveau détaillé que grossier, tandis que le bruit est aléatoire et non corrélé. Ceci permet de reconstruire des détails récurrents fins en éliminant le bruit. Cette approche permet d'être plus robuste aux conditions non idéales (bruits, aux données incomplètes, normales non orientées) que les méthodes de reconstruction traditionnelles. Une comparaison de *Point2Mesh* avec la reconstruction par poisson est illustrée à la Figure 2.10.

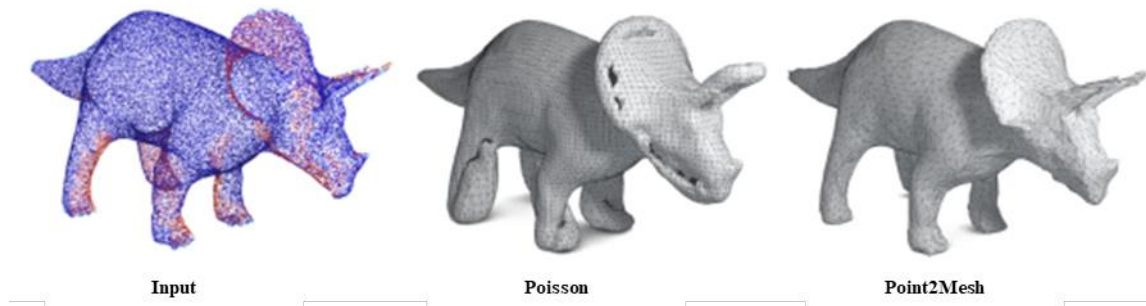


Figure 2.10 : Comparaison de *Point2Mesh* avec la reconstruction par Poisson, adapté de [172]

Points2surf (*P2S*) est un RDN ayant pour objectif de reconstruire une surface à partir d'un nuage de points non ordonné de façon à surpasser les performances des méthodes traditionnelles (par exemple la reconstruction de poisson filtrée [162]) dans des conditions difficiles [173]. Le modèle apprend un *prior* fort à partir de l'ensemble de données sur lequel il a été entraîné, ce qui permet une reconstruction robuste des surfaces même à partir de nuage de points de mauvaise qualité, tels que des nuages de points bruyants et faiblement échantillonnés. Un exemple de cette robustesse est illustré à la Figure 2.11.

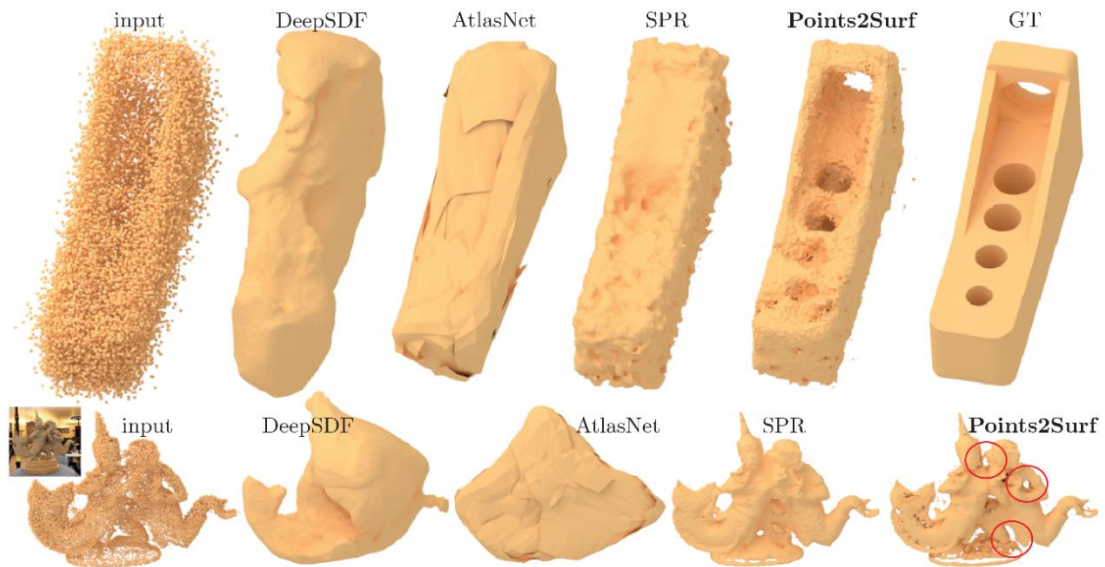


Figure 2.11 : Exemples de surfaces reconstruites par P2S à partir d'un NP bruité, tiré de [173]

L'architecture de *P2S* est illustrée à Figure 2.12. À partir d'un point x (rouge) et d'un nuage de point P (gris), *P2S* prend un échantillon local (*local patch* - jaune), et un sous-échantillon global grossier (*global coarse subsample* - mauve), du nuage de point. Ces échantillons sont ensuite encodés en

deux vecteurs de caractéristiques (*features*) qui sont transmis à un décodeur qui génère un logit de la probabilité de signe et de la distance absolue de la *Signed Distance Function* (SDF) du point $\mathbf{x}$.

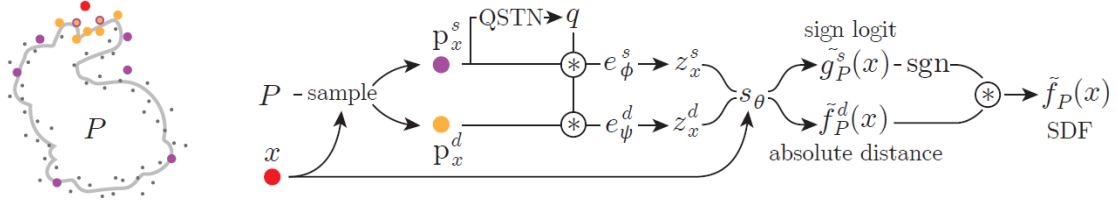


Figure 2.12 : Processus complet de *P2S*, tiré de [173]

P2S utilise la distance ℓ_2 —chanfrein pour évaluer l'erreur de reconstruction entre la surface reconstruite et les points appartenant au maillage ciblé. Un échantillonnage aléatoire du maillage est utilisé pour obtenir les points comparatifs. L'entropie croisée binaire est utilisée pour évaluer la classification logit du signe. La fonction de perte globale minimise la somme de ces deux distances.

P2S surpasse qualitativement et quantitativement autant les méthodes traditionnelles, telles que *SPR*, qui n'utilise pas de *prior* que les méthodes basées sur l'apprentissage d'un *prior* global, tels *AtlasNet* [174] et *DeepSDF* [175]. Les résultats démontrent que *P2S* possède une meilleure capacité de généralisation que les méthodes basées sur les *prior* globales et est plus robuste que *SPR*. Les comparaisons quantitatives sont basées sur la distance de chanfrein (*chamfer distance*).

Plusieurs autres études ont utilisé l'IA afin de reconstruire une surface géométrique [174-184]. La plupart de ces études partage le même processus que *P2S*, c'est-à-dire, qu'elles optimisent un RDN afin d'obtenir les valeurs d'une *Signed Distance Function*. Il est à noter que *P2S* demeure l'un des meilleurs modèles de reconstruction de surface.

Le

Tableau 2.4 résume les études ayant recours à l'apprentissage profond pour reconstruire une surface 3D à partir de données brutes. Il est possible d'y consulter l'année de publication, le type d'intrant, la technique de modélisation finale, l'architecture utilisée et les performances.

Tableau 2.4 : Résumé des études sur la modélisation 3D bout-en-bout

Études	Année de publication	Type d'intrant	Technique de modélisation	Architecture	Performance (%)
[166]	2019	Maillage irrégulier	Maillage triangulaire	<i>MeshCNN</i>	Classification = 92,16 % Segmentation de partie humaine = 92,30 %
[167]	2018	Maillage	Maillage	<i>CNN</i>	Classification = 91,9 % (<i>ModelNet40</i>)
[165]	2018	Nuage de points non structurés	<i>Marching Cubes</i>	<i>CNN</i>	Chanfrein = 0,218 Précision = 0,182 Complete = 0.254
[168]	2018	Image	Maillage	Auto-encodeur convolutif	<i>IoU</i> = 0,299 (<i>PASCAL3D+</i>) Dist3D = 0,323 (<i>PASCAL3D+</i>)
[169]	2018	Image	Maillage	Encodeur-Décodeur	60.16 % (voxel <i>IoU</i>)
[170]	2019	Image	Maillage	<i>R-CNN</i>	Chanfrein = 0,306
[171]	2019	Scan	Maillage	<i>Graph Neural Network</i>	Distance maillage = 0,0016 (<i>ShapeNet</i>) Similarité maillage normal = 0,83 (<i>ShapeNet</i>)
[172]	2020	Nuage de points	Maillage	<i>MeshCNN</i>	F-Score 81,24 % (bruité) F-Score 99,9 % (détaillé)
[173]	2020	Nuage de points	Reconstruction de surface implicite	<i>Points2surf</i>	Chanfrein : 1,80 (<i>ABC-no noise</i>)

Le

Tableau 2.4 permet de constater qu'il existe désormais plusieurs approches bout-en-bout de reconstruction de surface 3D. En fait, ces approches transforment des images 2D ou des intrants incomplets en maillage tridimensionnel de haute qualité résistante aux conditions non idéales telles que le bruit ou des intrants incomplets. Or, il est important de noter que parmi la liste d'approches présentées, seulement une minorité partage les mêmes intrants que les scans de soudure 3D provenant de *PAUT*. En effet, il s'agit à la base de voxels (ou de nuage de points si quelques opérations de transformations sont appliquées), ce qui limite le nombre de modèles possibles.

2.4 Suivi périodique des soudures

Il est possible de reconnaître des objets à partir de certaines caractéristiques bien précises. Les empreintes digitales en sont un bon exemple. Un lecteur d'empreinte est en mesure de reconnaître chacun des intrants qui lui sont montrés grâce à quelques méthodes d'identification comme *K-Plots Neighborhood descriptors*. *K-Plots* est une technique de détection d'objet dérivée de *Local Binary Patterns*. *K-Plots* a été appliqué avec succès à la reconnaissance de troncs d'arbres à partir d'images des caractéristiques visuelles des arbres (nœuds, signes de vieillesse, signe de coupe) [185]. L'algorithme débute en déterminant les coordonnées du centroïde des nœuds des troncs d'arbres. Le problème se résume ainsi à faire correspondre des points entre différentes images. Les distances et les angles des *K-Plots* voisins pour chaque pièce de bois sont stockés dans une matrice. Pour retrouver une pièce de bois dans la base de données, les *K-Plots* de la pièce à retrouver sont comparés aux *K-Plots* de chaque pièce de la base de données. La meilleure correspondance est obtenue en comptant le nombre de *K-Plots* correspondant et en déterminant quel panneau a donné la valeur la plus élevée. La méthode s'est montrée très robuste aux rotations des troncs d'arbres et aux autres aléas possibles au cours d'une chaîne de production de bois.

La recherche de *ANN* avec des données de grandes dimensions est une tâche fastidieuse, tant au niveau de la quantité de mémoires nécessaire que la complexité de calculs nécessaire [186]. Un bon algorithme de *ANN* réduit la liste de candidats potentiels à un seuil acceptable, et ce, sans avoir à calculer de mesure de similarité. Dans la famille des *ANN*, il existe plusieurs options qui pourraient fonctionner [61]. La liste inclut entre autres *LSH* et les autres techniques de hachage, les arbres kd [187], les arbres M [188] et les graphes [189].

Une étude effectuée la comparaison quantitative de plusieurs algorithmes de *ANN* depuis 2017 [61]. Leurs résultats permettent de connaître les performances des algorithmes sur plusieurs bases de données différentes. Il est ainsi possible d'observer des tendances grâce aux différents indicateurs de performance. En somme, les méthodes basées sur les graphes semblent surpasser les autres méthodes recensées tant au niveau des performances que de la robustesse. En revanche, l'étude a noté que la construction de l'index prend beaucoup de temps pour les bases de données ayant des

requêtes difficiles. Également, il est important de noter que les bases de données utilisées ont une dimensionnalité maximale de 960 alors que cette étude utilise des données largement plus importantes (de 10 000 à 30 000 points). Les performances des algorithmes testées sont donc grandement affectées.

Le concept d'*Image Similarity* correspond à l'utilisation d'une mesure de la similitude entre deux images. L'*Image Similarity* quantifie le degré de similitude entre deux images. Une façon de construire un modèle de similarité d'image consiste à extraire des fonctionnalités à l'aide de certaines techniques telles que *local binary patterns* ou les auto-encodeurs, puis de les utiliser pour calculer la similarité entre les fonctionnalités. L'autre méthode consiste à utiliser un modèle d'apprentissage profond pour faciliter l'identification d'images similaires pour une image donnée. *Deep Ranking* apprend la similarité des images en caractérisant les *patterns* à partir d'un ensemble de triplets [66]. Un triplet contient une image de recherche, une image positive et une image négative où l'image positive est similaire à l'image de recherche. Un modèle d'apprentissage profond est utilisé pour apprendre les similarités directement des images. Le réseau est constitué de 3 parties identiques ayant une architecture et des paramètres partagés : l'échantillonnage de triplets, le classement convolutif et le classement des similarités des images. Ce réseau prend en intrant les triplets d'images. Le triplet alimente indépendamment les trois RDN profonds. Les résultats de *Deep Ranking* surpassent les méthodes traditionnelles (*Wavelet*, *HOG*, *Golden Features*, etc.) d'*Image Similarity* en termes de précision.

La détection de copie de données géométriques 3D a fait l'objet de quelques études récemment [190]. Bien qu'il existe des méthodes reconnues et efficaces de détection d'images en 2D ou de texte, les techniques de détection de données géométriques 3D se font plus rares. La complexité de ce problème de détection consiste à déterminer si un modèle est une version manipulée d'un modèle *GT*. Les manipulations les plus fréquentes sont la rotation, l'ajout de bruit et le recadrage. Les méthodes traditionnellement utilisées pour résoudre ce problème empruntent le concept de filigrane (*watermarking*) au filigrane numérique [191]. Ce concept correspond à l'intégration d'informations reconnaissable aux données numériques et la détection de ces filigranes pour la reconnaissance. Les marques ajoutées sont généralement imperceptibles, mais peuvent être détectées. Il existe différents algorithmes de *watermarking*. Le *watermarking* est largement utilisé pour détecter la copie de contenus numériques. Ce concept est toutefois vulnérable à l'ajout de bruit et au recadrage.

Des techniques de *watermarking* spécifiques au maillage triangulaire 3D sont maintenant disponibles [192]. Le *watermarking* de maillage triangulaire 3D fait face à des deux problèmes spécifiques importants : (1) les maillages sont par nature non organisés ce qui fait qu'il est difficile de définir une analyse systématique de la surface et (2) il est difficile d'anticiper tous les types d'attaques que les maillages peuvent subir considérant tous les outils de manipulation de maillage disponibles.

Les techniques proposées modifient la géométrie ou la topologie des maillages de façon à laisser une empreinte détectable. Les vertices, les faces et les normales peuvent donc être modifiés par ces techniques. D'autres techniques modifient la distribution des normales des vertices. Ils existent plusieurs variations à ces techniques. Il est à noter que la reconnaissance des maillages est faite en calculant la distance de chanfrein.

Bien que ces techniques soient intéressantes pour détecter des fraudes de droits d'auteur, elles ne s'appliquent pas au problème de comparaison périodique des modèles géométriques des soudures. En effet, les modèles géométriques sont générés par un processus de segmentation et de reconstruction de surface automatisé. Le modèle de base (requête) et celui recherché (la même soudure déjà scannée et modélisée) ne sont pas des copies numériques altérées d'un même modèle, mais deux modèles géométriques pris de la même soudure. Il est donc impossible d'ajouter des filigranes indicatifs dans ce processus.

Une autre solution au problème de détection de modèles géométriques similaires est d'extraire la forme de l'objet et de comparer la similitude des deux formes 3D. Une étude s'attarde à améliorer la comparaison de formes 3D pour le cas de fraude de droit d'auteur [190]. La comparaison débute par l'alignement des modèles et est suivie par le calcul de la similitude par 3 différents indicateurs. Les trois indicateurs sont basés sur les matrices de probabilités. Il s'agit de la mesure *low-rank*, de kurtosis et de corrélation. Cette technique est en mesure de surpasser le *watermarking*. Il est à noter que le calcul de la similitude est très lourd algorithmiquement. À cet effet, les auteurs suggèrent de l'utiliser pour les cas où la durée du calcul n'est pas importante. Bien que cette technique semble intéressante pour effectuer la comparaison de modèles 3D, elle ne résout pas le problème de recherches des plus proches voisins, c'est-à-dire de retrouver les candidats qui proviennent potentiellement du même objet.

À cet effet, il est important de noter que les techniques discutées à la section 2.1 peuvent également s'appliquer à la reconnaissance des soudures. Il suffit de modifier le type d'intrants et modifier les configurations des algorithmes de recherche des plus proches voisins. Ainsi, *LSH* pourrait être utilisé comme algorithme de recherche des plus proches voisins et la comparaison pourrait être faite avec la combinaison des trois indicateurs.

2.5 Positionnement de la recherche

La revue de littérature a présenté plusieurs études liées aux 4 objectifs de cette thèse. Elle a permis de constater qu'il existe sûrement plusieurs approches de solution au problème d'analyse des soudures à partir de fichiers ultrasons. Par contre, il ne semble pas y avoir de processus d'analyse complètement intégré qui valide la qualité et l'authenticité des données, localise la soudure et les

défauts, en assure un suivi périodique, le tout dans un logiciel infonuagique. Un tel processus constituerait une contribution substantielle puisqu'il permettrait d'automatiser plusieurs tâches manuelles qui actuellement augmentent les délais et les coûts d'analyse. Il en résulterait une réduction considérable des délais depuis l'acquisition des données jusqu'à l'analyse de la soudure, avec un potentiel de suivi de son évolution dans le temps.

De façon plus détaillée, la revue de la littérature a premièrement permis de constater qu'il n'existe aucune méthode publique pour identifier une soudure afin de la reconnaître et éviter des cas de fraudes ou des erreurs de manipulation (objectif a). En effet, la littérature des END ne contient aucun exemple de méthodes permettant l'évaluation d'un fichier de scan pour des fins de contrôle de son authenticité.

Deuxièmement, l'amélioration des architectures IA et l'augmentation de la puissance des ordinateurs ont conduit au développement de détecteurs de défauts de soudure nettement plus performants, ce qui permet maintenant de localiser et de dimensionner les défauts. Les récentes architectures développées depuis 2012 sont capables de détecter correctement les défauts de soudure jusqu'à une MAP de 85 % [86]. Cependant, aucun des réseaux développés jusqu'à présent ne s'attarde à la segmentation *de la géométrie de la soudure en plus des défauts* (objectif b), et aucun ne semble *utiliser les scans 3D* comme intrants.

Troisièmement, il n'existe à l'heure actuelle aucune recherche qui se soit attardée à la représentation géométrique 3D de la géométrie de la soudure et de ses défauts possibles (objectif c), résultant de la segmentation effectuée a priori (objectif b).

Il semble donc évident que les recherches futures devraient se concentrer sur l'utilisation de modèles 3D tant pour la segmentation que la modélisation géométrique 3D d'une soudure et ses défauts. En effet, les réseaux actuels ne peuvent pas fournir d'informations exactes concernant la position et la taille des défauts identifiés en 3D. L'industrie bénéficierait grandement de ces connaissances supplémentaires. Qui plus est, le développement d'une méthode pour reconnaître une soudure et comparer son évolution temporelle à partir de son « ADN géométrique » en 3D est également un aspect qui démarque cette recherche. Il sera alors possible de suivre l'évolution d'un défaut ou la dégradation de la soudure générale en mettant en évidence les différences entre deux scans de la même soudure à des moments différents (objectif d).

Soulignons finalement l'absence totale de solutions infonuagiques d'analyse de soudure à partir de données ultrasonores. Cet aspect est également au cœur des solutions proposées dans cette thèse. L'une des contributions potentielles de cette recherche, liée à son implantation pratique cette fois,

consisterait donc en la création d'un processus infonuagique automatisé d'analyse des soudures, qui en somme reprend chacun des objectifs de cette recherche, encore ici répétés :

- a) Assurer l'intégrité et l'authenticité des données ultrasonores soumises pour la première fois ;
- b) Identifier les différentes parties d'une soudure ;
- c) Reconstruire la surface géométrique 3D d'une soudure ;
- d) Assurer un meilleur suivi des analyses périodiques d'une même soudure.

L'atteinte de chacun de ces objectifs et l'implantation infonuagique des processus résultants pour fins d'audits constituent autant de contributions originales décrites dans les chapitres suivants de cette thèse.

3 MÉTHODOLOGIE DE LA RECHERCHE

Ce chapitre est divisé en cinq sections. La première section décrit les données disponibles. La section 3.2 décrit les différents types de doublons et présente en détail le processus complet de détection des doublons ainsi que les processus individuels de détection des *ED* et des *ND*. La section 3.3 présente le choix du modèle de segmentation sélectionné ainsi que le processus de sélection de ses HP. La section 3.4 présente le processus de reconstruction de surface développée dans cette étude et l'algorithme de reconstruction de surface choisi ainsi que ses HP. Finalement, la section 3.5 décrit le processus de suivi périodique choisi ainsi que de ses paramètres importants.

Ce chapitre présente non seulement les différents choix méthodologiques possibles de chacun des objectifs, mais il permet également d'établir la méthodologie complète de cette étude. Pour ce faire, certaines expériences et calculs sont nécessaires afin de sélectionner le processus optimal permettant de résoudre les quatre objectifs spécifiques. Ainsi, le prochain chapitre 4 se concentrera seulement sur les résultats du processus complet d'analyse des soudures.

3.1 Description des données

Dans plusieurs domaines de recherche, des ensembles de données publiques et annotées sont disponibles pour aider les chercheurs à parfaire leur RDN profond [2]. Les chercheurs peuvent utiliser ces ensembles de données publiques pour améliorer la qualité des réseaux développés. Les performances, utilisant des mesures prédéfinies, sont publiées en même temps que l'architecture du réseau, ce qui permet aux autres chercheurs de reproduire leurs travaux. Ces mesures peuvent être comparées et classées, car elles sont évaluées sur le même ensemble de données. Un tel processus permet d'améliorer en permanence la qualité des réseaux de neurones profonds et de réduire le temps de développement. Or, il n'existe aucune base de données publique de données ultrasonores contenant des défauts de soudure, ce qui rend ce processus impossible à reproduire. Les résultats de cette étude sont donc produits à partir d'une base de données privée de scans ultrasonores de soudures.

Les données actuelles, fournies par le partenaire industriel (Ondia), sont des données réelles que Ondia a obtenues de clients et de collaboration avec des partenaires industriels, tel que *The Welding Institute*. Les données ultrasonores sont des séquences de points qui balayent la soudure d'un bout à l'autre. Il est possible d'associer le fichier à une séquence de points dans un très grand nombre de coupes 2D successives. Le nombre de points peut excéder plusieurs millions et chaque point comporte au moins quatre coordonnées : X, Y, Z, Amplitude. Les données proviennent de différents types de scan (par exemple ultrasons multiéléments : A-Scan digitalisé et S-Scan digitalisé), tel que présenté à la Figure 3.1 [15]. La Figure 3.1 présente un exemple de scan 3D *PAUT* où un défaut est

mis en évidence à l'aide d'une boîte englobante. Le même scan est illustré en format S-Scan. Dans (a), l'entière du scan est représentée alors que dans (b) il s'agit d'un seul *frame*. Les coordonnées utilisées pour les deux représentations sont les coordonnées cartésiennes x, y, z.

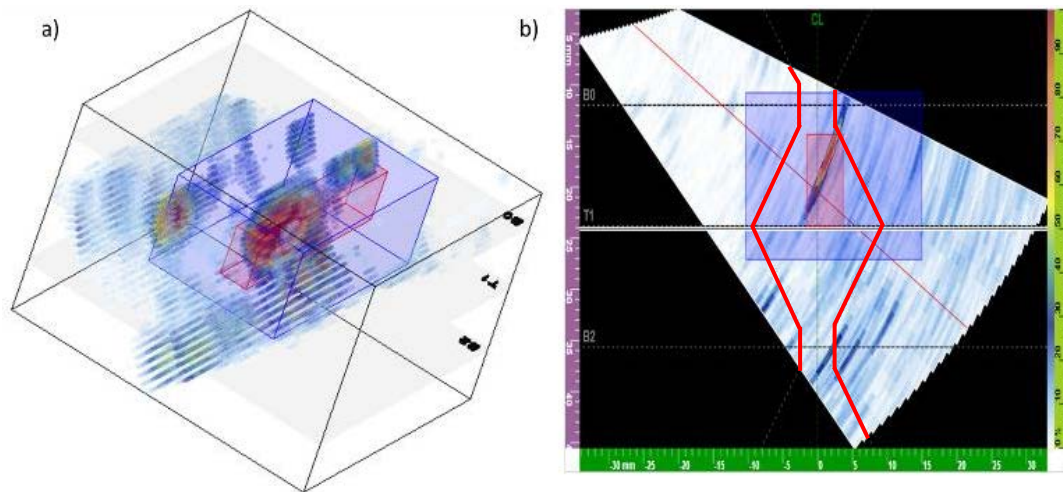


Figure 3.1 : Un scan représenté sous deux formats.(a) Un scan 3D *PAUT* mettant en évidence un manque de fusion des parois latérales (b) le même scan affiché comme un S-Scan, mettant à nouveau en évidence un manque de fusion des parois latérales. La forme théorique de la soudure (simple-V) a été ajoutée en superposition rouge pour une meilleure compréhension.

Parmi l'ensemble des scans ultrasonores disponibles, 106 ont été annotés par deux experts en *PAUT* de Ondia (niveau III). Les annotations correspondent à chacune des parties de la soudure : matériel de base, soudure en V, soudure en double-V, reflet incomplet supérieur et inférieur. Parmi les 106 annotations, 59 appartiennent à une soudure en double-V et 47 en V, ce qui équivaut à 56 % de soudure en double-V et 44 % en V. Parmi les 106 scans annotés, les défauts annotés correspondent à une inclusion, un manque de fusion, une pénétration incomplète de la racine, une fusion incomplète de la racine et à une fissure (au centre, à la racine, au renforcement, à la paroi latérale). Il est à noter que les annotations des deux experts ont été combinées. Tous les cas où les annotations des deux experts étaient différentes ont été analysés et discutés avec les experts. Des correctifs ont ensuite été apportés. La Figure 3.2 présente un ensemble de S-Scans annoté en 3 D. La première colonne représente un S-Scan brut, la seconde est l'annotation par l'analyste de Ondia des défauts, et la dernière correspond à l'annotation de la soudure. La première ligne est une soudure en V et la deuxième est une soudure en double-V. Les couleurs des annotations des défauts (S-Scan du centre) correspondent à l'intensité des défauts (du bleu (aucun défaut), à vert (faible), à jaune (moyenne) et à rose (défaut le plus intense). Il faut rappeler que les représentations des S-scans de la Figure 3.2 utilisent la valeur moyenne sur l'entière des *frames*. Les couleurs des annotations de la soudure

(S-Scan de droite) correspondent au matériel de base (en bleu), à la soudure en V (en vert), à la soudure en double-V (en jaune) et à la réflexion incomplète de la soudure (en rose).

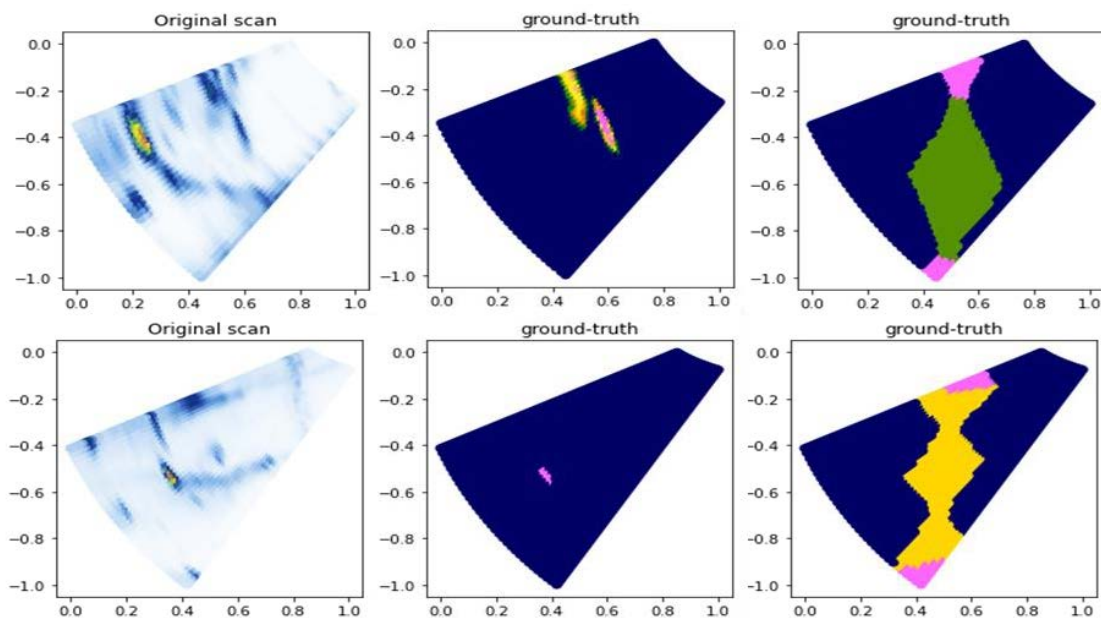


Figure 3.2 : Exemples de données brutes S-Scan (original scan - gauche), de défauts annotés

(GT - milieu) et de parties de la soudure annotée (GT - droite)

Les soudures ont été réalisées à l'aide de différentes méthodes de soudage : MIG, TIG, arc submergé, soudage automatisé et soudage manuel. Les soudures sont de tailles variables, en fonction de l'épaisseur des plaques (entre 8 et 75 millimètres). Les données ultrasonores sont obtenues à l'aide d'appareil *PAUT* utilisant des scans sectoriels (S-Scans) angulaires en mode onde de cisaillement (*shear wave*), de résolutions angulaires variables et de pas de digitalisation variable. Les S-Scans ont généralement une plage angulaire entre 35 ° et 70 ° dans l'acier. La fréquence des sondes utilisées varie de 2 MHz à 10 MHz. Aucun filtre de fréquence n'est utilisé. Il n'y a qu'un seul matériau considéré dans cette recherche, il s'agit de l'acier au carbone. C'est l'un des matériaux offrant le meilleur rapport signal sur bruit et c'est l'un des plus utilisés dans l'industrie.

Afin de valider de la qualité des annotations, une comparaison des annotations est effectuée. Cette comparaison est résumée au

Tableau 3.1. On peut résumer les annotations grâce à quelques indicateurs : la position de la soudure, l'épaisseur de la soudure et le type de la soudure.

Tableau 3.1 : Comparaison quantitative des annotations par les experts

Catégorie	Moyenne Expert 1	Moyenne Expert 2	Différence moyenne	Écart-type de la différence
Position	0,28 mm	-0,03 mm	0,91 mm	0,79 mm
Épaisseur	18,54 mm	18,98 mm	0,41 mm	0,38 mm

On peut constater que les différences moyennes entre les annotations des experts sont faibles au niveau de l'épaisseur de la soudure. En effet, la différence moyenne est de seulement 0,41 mm alors que les soudures ont une épaisseur d'environ 18 mm. L'écart type est de 0,38 mm, ce qui confirme que les différences sont faibles, peu nombreuses et peu importante. En effet, la plus importante différence est de 1,2 mm alors qu'un expert évalue l'épaisseur d'une soudure de 19,0 mm alors que le second expert l'évalue à 20,2 mm. Au niveau de la position de la soudure (par rapport au centre du scan), la différence absolue moyenne est de 0,91 mm et l'écart-type est 0,79 mm. Considérant la largeur d'un scan et d'une soudure, il s'agit d'une différence très faible. Sachant que la Zone Affectée Thermiquement (ZAT) est de plusieurs millimètres de chaque côté de la soudure, une différence moyenne de 0,91 mm est acceptable. Il est à noter que la différence la plus importante est de seulement 2,5 mm. Pour ce qui est des types de soudures, les experts ont obtenu des résultats identiques (56 % de soudure en V et 44 % de soudure en double-V). Somme toute, les annotations par les deux experts sont très similaires.

3.2 Détection des doublons

En END, un *ED* représente le cas où deux scans ultrasonores (A-Scan, B-Scan, C-Scan, S-Scan) sont identiques à 100 %. La Figure 3.3 présente 4 scans distincts identifiés de (a) à (d). Les scans (a) et (b) représentent un S-Scan provenant d'un scan ultrasonore en trois dimensions. Les scans (c) et (d) représentent une image ultrasonore représentant un seul *frame* d'un scan ultrasonore tridimensionnel. Nous pouvons observer que les deux graphiques (a et b) sont des *ED* puisqu'ils sont complètement identiques, même chose avec les images (c et d). Les images sont donc identiques. Par conséquent, tous les scans proviennent du même fichier original.

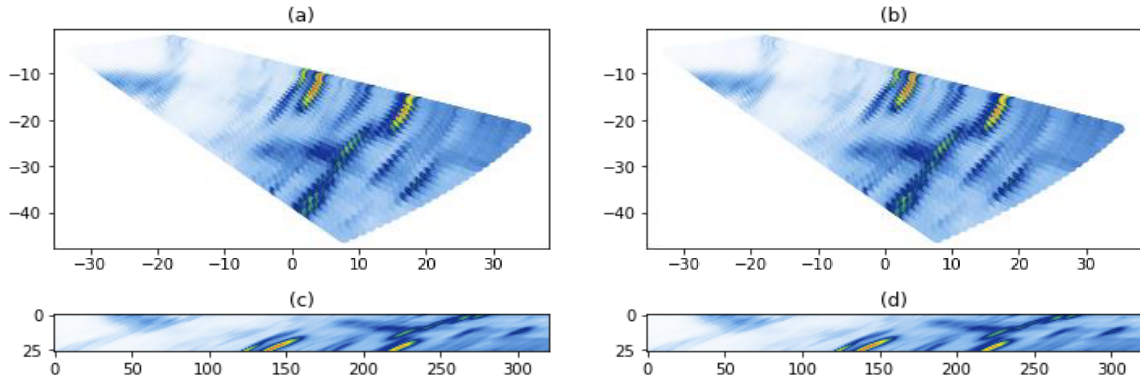


Figure 3.3 : Exemples de *ED* : S-Scan (a et b) et C-Scan (c et d)

Un *ND* correspond à une copie d'un scan original avec de légères différences, fréquemment générées lors de l'acquisition du scan. En général, ces différences proviennent de petites transformations des données de l'original dues à la modification de certains paramètres de l'appareil *PAUT* (nombre d'échantillons, filtres, mise au point, angle angulaire ou départ/fin du balayage) alors que le matériau testé reste le même. D'autres transformations peuvent inclure des décalages dans les données, la suppression d'une partie du scan original, la normalisation de l'amplitude, et bien plus encore. Les *ND* ont une très grande similitude avec les données originales, mais ils ne peuvent pas être considérés comme des *ED*.

Pour illustrer ce que représente un *ND*, la Figure 3.4 montre à nouveau quatre scans notés de (a) à (d). Les scans (a) et (c), respectivement un S-scan et un C-scan, ont été générés à partir des données originales tandis que les scans (b) et (d) ont été générés à partir des mêmes données ultrasonores, mais sous des paramètres différents. Bien que les scans (a) et (b), ainsi que (c) et (d), se ressemblent grandement, ils sont en fait très différents. Mais ces différences sont suffisamment faibles pour tromper la plupart des logiciels actuellement utilisés en END.

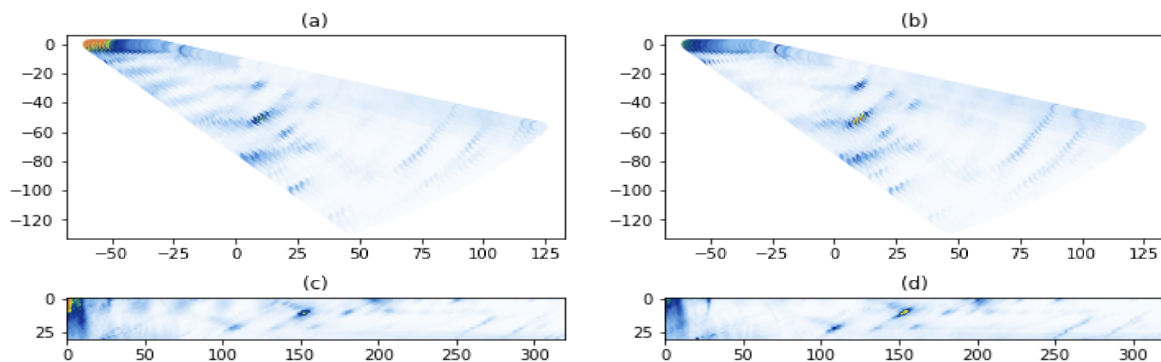


Figure 3.4 : Exemples de *ND* : S-Scan (a et b) et C-Scan (c et d).

Les méthodes mises au point pour trouver les *ED* peuvent facilement valider la correspondance exacte de deux fichiers, mais dès qu'ils ne sont pas identiques, les méthodes deviennent inefficaces [193]. Ainsi, une méthode de détection des *ED*, différente de celle des *ND*, est nécessaire.

À partir d'un fichier ultrasonore, le processus de détection des doublons procède donc en deux étapes parallèles : une pour *ED* et une autre pour *ND* (Figure 3.5).

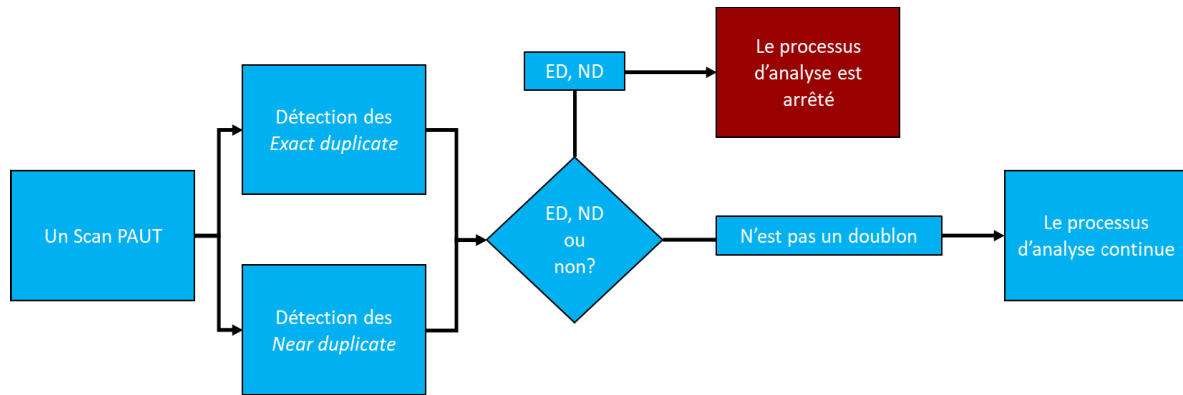


Figure 3.5 : Processus de détection des doublons

La tâche de trouver des *ED* peut se faire soit en comparant chaque point de deux fichiers ultrasonores, soit en échantillonnant les données au hasard et en comparant ces échantillons. Ces méthodes de force brute sont cependant très inefficaces et coûteuses en termes de calcul. Les données ultrasonores recueillies par un appareil *PAUT* peuvent contenir plusieurs millions de points (en 4 dimensions). L'analyse de points échantillonnés aléatoirement réduit la charge de calcul du logiciel et est statistiquement valable. Toutefois, cette technique ne permet pas de trouver les *ND*, car la modification d'une seule donnée ferait échouer ce processus. Une légère modification telle que l'ajout d'un point dans un fichier ultrasonore tromperait la méthode de détection des *ED*. La recherche de *NDs* n'est pas une tâche triviale puisqu'elle nécessite une importante puissance de calcul et une complexité algorithmique.

Une première validation de l'authenticité et de l'intégrité des données doit être effectuée rapidement dans le processus global, avant même de procéder à une étape d'analyse. Cette validation doit se faire immédiatement après la transformation des données en format neutre. C'est à ce moment qu'il faut vérifier que les données sont uniques, c'est-à-dire qu'elles ne proviennent pas de fichiers déjà analysés, que ce soit par erreur ou par fraude.

La stratégie proposée consiste à générer une empreinte numérique d'un fichier de données ultrasons pour la stocker dans une base de données pour fin de comparaison et de validation de son authenticité. Avant chaque analyse, l'empreinte générée à partir d'un nouveau fichier sera comparée

avec la base de données. Les fichiers ayant des empreintes numériques identiques seront détectés et une notification sera émise par le logiciel.

La comparaison point par point des données étant très lourde et peu efficace, la méthode de comparaison préconisée sera le hachage. À partir d'un scan ultrasonore de très grande taille, les fonctions de hachage pourraient être utilisées efficacement. En effet, les fonctions de hachage ont déjà démontré leur efficacité à identifier un très grand ensemble de données sans grande difficulté [193]. Ainsi, elles ne devraient pas avoir de problèmes à générer un *hash* des données ultrasonores, et ce malgré la quantité importante de points (plusieurs millions) à considérer.

Une seconde technique est étudiée. Il s'agit de *LSH* qui est généralement utilisée pour retracer les images les plus proches d'une image [64]. Il s'agit d'une technique de recherche des plus proches voisins largement plus efficaces que K-NN par exemple. Dans le contexte de cette recherche, la technique sera adaptée pour permettre de retracer les scans ultrasonores similaires.

Étant donné la nature des doublons, il est nécessaire de diviser le processus de détection en deux étapes parallèles. Puisqu'un *ED* partage 100 % de ses attributs avec un autre fichier, il n'est pas nécessaire que le fichier identifié en tant que *ED* poursuive le processus d'analyse complet. L'analyse a déjà été faite sur le fichier original. En outre, le *ED* peut être facilement identifié, même dans une base de données de plusieurs millions de scans ultrasonores. Ainsi, la détection d'un *ED* permet d'économiser du temps et de la puissance de calcul. Compte tenu du nombre de scans que l'industrie des END crée et traite, cela représente un avantage important.

Puisque les deux méthodes répondent à des cas spécifiques, elles seront implémentées successivement et intégrées au logiciel infonuagique afin d'offrir davantage de sécurité et de robustesse. La première méthode doit être efficace et rapide, tandis que la seconde doit être robuste face à de légères modifications des paramètres de l'appareil d'acquisition (*PAUT*). Les deux méthodes doivent être en mesure de fonctionner sur des données de grande dimension et des ensembles de données volumineux. La méthode utilisée pour détecter les *ED* est illustrée à la Figure 3.6.

Le processus commence par un fichier ultrasonore. Un *hash* est généré pour ce fichier. Ensuite, une requête est envoyée à la base de données pour déterminer si cette valeur de hachage est déjà associée à un autre fichier. Un *ED* est détecté si la requête renvoie une correspondance positive.

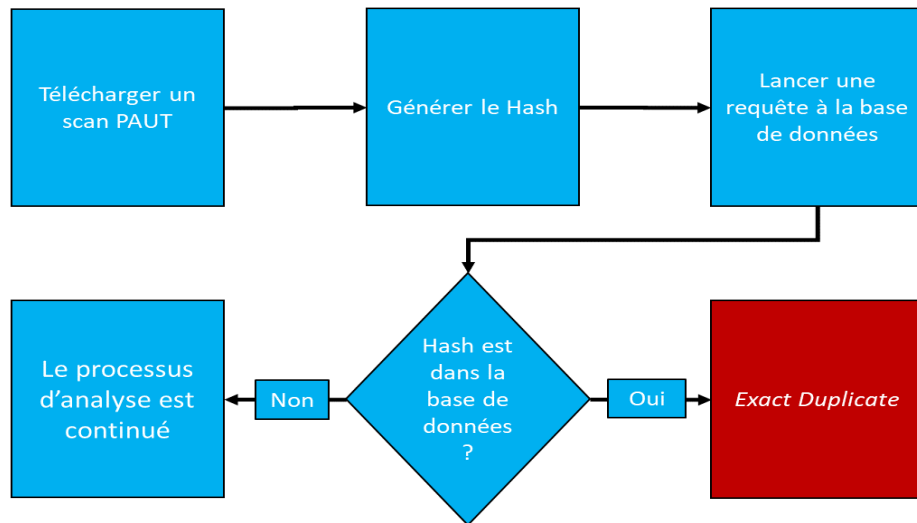


Figure 3.6 : Processus de détection des *ED*

Le processus de détection des *ND* est illustré à la Figure 3.7 [62]. Le processus commence par un ensemble d'enregistrement *R* qui est suspecté d'inclure des *ND*. Un algorithme est défini pour choisir les candidats prometteurs parmi toutes les paires possibles. La mesure de similarité est ensuite calculée pour tous les candidats. La dernière étape de ce processus consiste à appliquer un seuil de similarité pour décider si le candidat est effectivement un *ND* ou non.

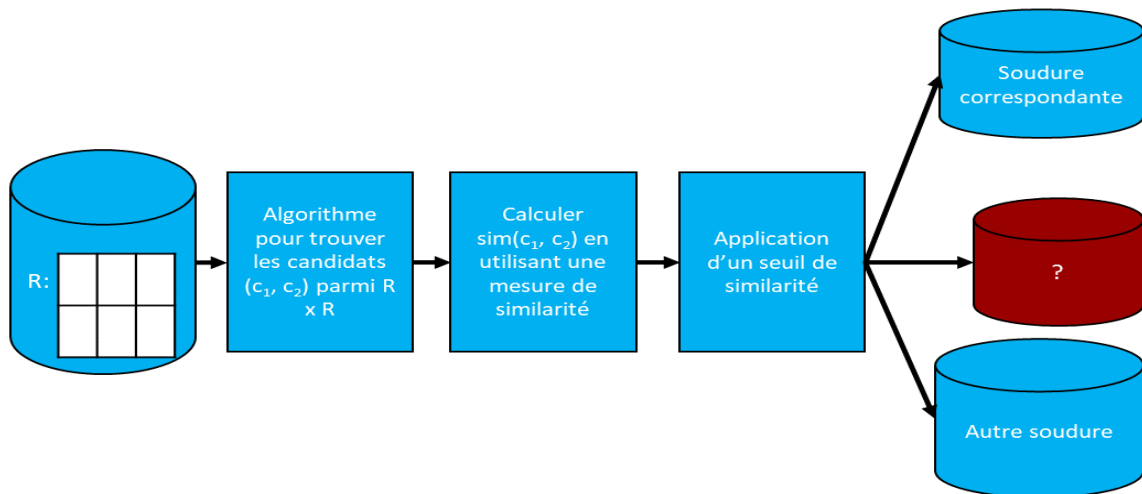


Figure 3.7 : Un processus standard de détection des doublons, adapté de [51]

Afin de détecter les *ND*, il est nécessaire de quantifier la ressemblance entre deux fichiers ultrasonores. Les fonctions de similarité prennent deux fichiers en intrant et déterminent si un candidat est un doublon ou non [51]. Considérant le seuil de similarité τ , une fonction de similitude

permet de classer deux candidats c et c' en utilisant une mesure de similarité, en tant que doublons ou non. La similarité du cosinus est une mesure de similarité très populaire qui est utilisée dans diverses situations, telles que la recherche d'informations. Elle mesure la différence d'orientation entre deux vecteurs. La similarité du cosinus est calculée par le produit scalaire de deux vecteurs divisés par leur amplitude. Par conséquent, deux vecteurs perpendiculaires ont une similarité en cosinus de 0 alors que les vecteurs parallèles ont une similarité en cosinus de 1. La distance euclidienne, également appelée *distance* L^2 , est une mesure utilisée en statistique et en *machine learning*. Elle représente la somme des erreurs au carré entre deux vecteurs, où les erreurs sont la distance en ligne droite entre deux points dans l'espace euclidien. L'avantage de cette mesure de distance est qu'elle pénalise davantage les erreurs importantes. Une mesure de distance plus précise est la *SSIM* [194, 195]. Les mesures basées sur la distance ℓ^p calculent les distances par pixel (par voxel en 3D) en assumant que les pixels voisins sont indépendants les uns des autres.

L'algorithme choisi pour trouver les candidats de *ANN* est *LSH*, compte tenu de son efficacité algorithmique et de son efficacité potentielle. Il faut rappeler que les deux méthodes de détection des doublons sont développées pour être intégrées dans un logiciel infonuagique d'analyse END actuellement disponible, expliquant ainsi le besoin de méthodes de détection légères, mais efficaces.

3.3 Identification de la soudure

Pour répondre à cet objectif, on procède en plusieurs étapes distinctes, soit le choix du modèle de segmentation, l'optimisation de ses hyperparamètres et les explications justifiant ces choix. Chacune de ces étapes est présentée individuellement.

3.3.1 Apprentissage multitâche

Il existe des modèles de segmentation qui sont en mesure de produire plus d'une carte de segmentation ; ce type de modèle se nomme l'apprentissage multitâche [196]. Dans la littérature, un bel exemple de ce type de réseau a réussi à produire deux cartes de segmentation distinctes (segmentation des immeubles et prédiction de la distance) à partir d'une image aérienne [196]. Ce *FCN* appelé *SegNet* consiste en un encodeur-décodeur similaire à *U-Net* et *V-Net*. Une dernière couche de convolution est ajoutée pour prédire la distance des immeubles et une autre est ajoutée pour produire la carte de segmentation des immeubles. Le résultat de cette étude montre que le modèle multitâche a de meilleures performances que le même modèle entraîné à seulement segmenter des immeubles.

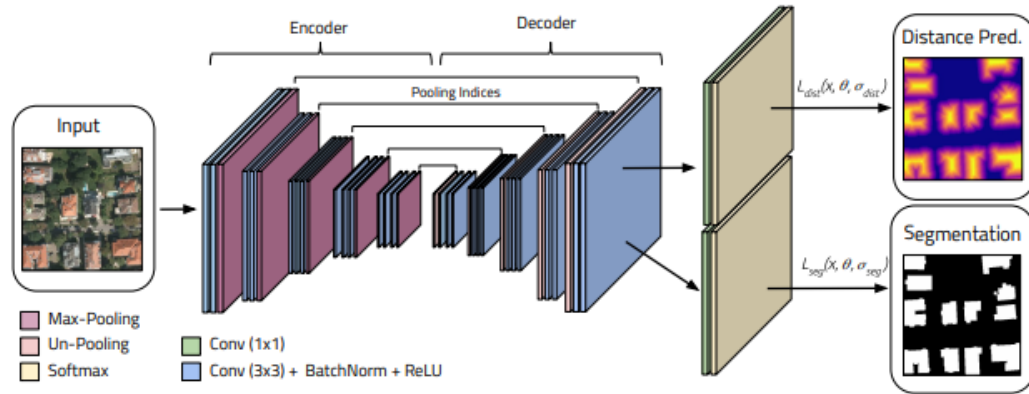


Figure 3.8 : Exemple de *CNN* multitâches, tiré de [196]

Ce dernier résultat justifie l'ajout de la segmentation des défauts à l'objectif b) (identification de la soudure), qui ne portait initialement que sur la segmentation de la frontière entre le matériel de base et le métal fondu. Le modèle de segmentation présenté à la Figure 3.8 peut segmenter à la fois la soudure et ses défauts. Pour ce faire, une deuxième branche décodeur est ajoutée au modèle de segmentation à partir de l'espace latent. La seconde branche génère une carte de segmentation binaire permettant de savoir si chaque voxel inclut ou non un défaut. Cette branche est entraînée à partir de données de soudure annotées par de Ondia. En plus de potentiellement produire de meilleurs résultats de segmentation, ce modèle amélioré permet de superposer les défauts détectés à la représentation 3D de la soudure, qui à son tour permet d'améliorer la qualité des analyses des scans ultrasonores. Il devient possible de considérer à la fois la localisation, la forme et la dimension de la soudure en plus de ses défauts.

3.3.2 Choix du modèle de segmentation

Le type de modèle sélectionné dans cette étude est un *FCN* 3D [102]. Il s'agit d'un modèle en deux étapes qui encode les intrants dans une représentation latente, une version compressée des intrants, et reconstruit la représentation latente dans son format d'origine. Le réseau essaie de minimiser l'erreur entre la segmentation prédite et la segmentation annotée par un expert, qui correspond à la cible [98]. Le modèle prend en entrée des scans *PAUT* 3D normalisés de taille (voxels) (1, 80, 32, 80), l'encode dans un espace latent de taille prédéfinie et le reconstruit à une taille de (c, 80, 32, 80) où c est le nombre total de classes (4 pour la soudure et 2 pour les défauts).

Aucune technique de débruitage n'est appliquée. Les points appartenant à la soudure peuvent être considérés comme étant du bruit. Alors, il serait risqué de perdre de l'information précieuse lors du débruitage. De plus, la revue de littérature permet de constater que les meilleurs résultats de

classification des défauts de soudure à partir de scans ultrasonores proviennent d'études qui n'utilisent aucune technique de débruitage [13, 85].

Il existe de multiples modèles de *FCN* permettant la segmentation de données en trois dimensions. Or, ces modèles ont des particularités leur permettant d'obtenir de meilleures performances dans des conditions bien précises. Parmi la liste des *FCN* intéressants recensés au chapitre 2, il y a *U-Net*, *V-Net*, *DenseNet*, *ESPNet*, *DeepLab* ainsi que les réseaux entraînés de façon contradictoire. Le choix final du modèle a été obtenu à l'aide d'un algorithme d'optimisation. La sélection du modèle final et de ses hyperparamètres est discutée à la section suivante. Le Tableau 3.2 présente quelques modèles sélectionnés pour effectuer la segmentation de la soudure.

Tableau 3.2 : Modèles de segmentation 3D sélectionnés

Modèles sélectionnés	Année de sortie	Attraits particuliers	Nombre de paramètres (millions)
<i>U-Net</i> [99]	2016	Bonnes performances lorsqu'entraîné avec très peu de données	2,37
<i>V-Net</i> [102]	2016	Bonnes performances lorsqu'entraîné avec très peu de données	45,61 ou 12,25
<i>DeepLabv3</i> [104]	2020	Très profond.	44,37
Contradictoire [114, 115]	2019	Résultats visuels très précis grâce à l'entraînement contradictoire.	46,50 (44,37 [G] + 2,13 [D])
<i>ESPNet</i> [197]	2018	Modèle léger, efficace pour un ensemble de tâches varié	3,61
<i>DenseNet</i> [100, 105]	2017	Résout le problème de <i>vanishing gradient</i> pour les modèles profonds grâce à une meilleure connexion des couches	1,55

Bien qu'il existe une multitude d'autres modèles ou d'architectures potentiellement intéressantes pour effectuer la segmentation, le choix d'un *FCN* est justifié par ces caractéristiques. En effet, les *FCN* n'ont pas besoin de beaucoup de données d'entraînement annotées pour atteindre des performances de segmentation importantes, et ce même en 3D [99, 102]. En effet, dans la publication originale, *V-Net* a démontré obtenir d'excellentes performances de segmentation en ayant été entraîné avec moins de 50 données, et ce même s'il s'agit d'un modèle 3D. Il faut toutefois faire appel à l'augmentation de données. Bien que les autres modèles répertoriés requièrent davantage de données pour atteindre de bonnes performances, leur besoin est moindre que d'autres types d'architectures comme les *Generative Adversarial Networks*. Un autre avantage des *FCN* est l'absence de couche linéaire (*Flatten* + *Linear*) ce qui permet une excellente contraction des données et ce qui limite le nombre de paramètres à apprendre. De plus, les *FCN* sont en mesure d'utiliser des intrants de tailles variables, pour autant que la taille des intrants soit plus grande que le ratio de contraction et qu'un multiple du même ratio. Le fait de ne pas redimensionner les intrants à une taille

fixe est un avantage intéressant lorsqu'on utilise des données ultrasonores issues de *PAUT*. En fait, la taille des scans varie de 1 million de points (*frames*beams*samples*) à plus d'un milliard.

Évidemment, chacun des modèles possède des particularités intéressantes. Or, il est difficile de sélectionner le meilleur modèle puisque le choix des hyperparamètres a un grand impact sur les performances du modèle, d'où la nécessité d'utiliser un algorithme d'optimisation des hyperparamètres.

3.3.3 Sélection du modèle optimal et de ses hyperparamètres

La sélection des hyperparamètres n'est pas une tâche triviale et peut affecter grandement les performances du modèle. Pour sélectionner l'ensemble optimal d'hyperparamètres, l'algorithme d'hyperband est utilisé. Il s'agit d'un algorithme de réduction de moitié successive [30]. Il permet de tester plusieurs configurations d'hyperparamètres et de déterminer le sous-ensemble optimal. *Hyperband* fonctionne comme ceci : pour un budget initial donné R , *Hyperband* détermine aléatoirement R sous-ensemble d'hyperparamètre et les exécute tous pour ce budget (c'est-à-dire une *epoch*) ; ensuite, *Hyperband* choisit une fraction ($1/\eta$) des algorithmes les plus performants, augmente le budget η fois et répète ce processus successivement jusqu'à ce qu'il ne reste plus qu'un seul sous-ensemble (le meilleur) [198]. Il a été démontré que *Hyperband* est une technique d'optimisation des hyperparamètres qui performe bien, en termes de temps d'exécution et d'itérations nécessaires, en comparaison des techniques similaires [199]. *Hyperband* surpasse les techniques d'optimisation basées sur l'allocation uniforme des ressources (budget), sur la recherche aléatoire (*random search*) et sur la théorie de Bayésien. Un exemple d'algorithme de *Hyperband* est illustré à la Figure 3.9.

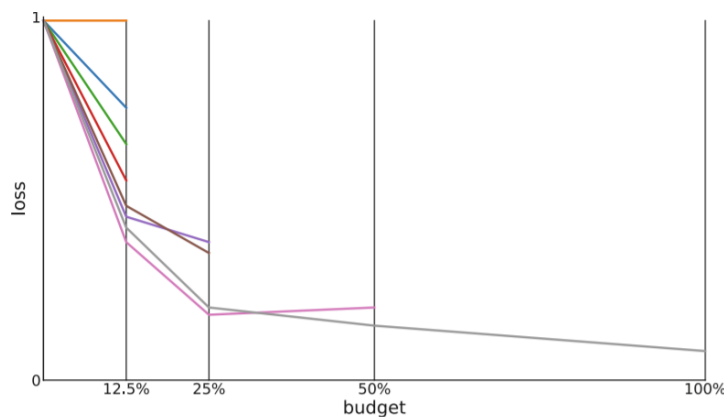


Figure 3.9 : Illustration de *Hyperband* avec un budget de 8, tiré de [198]

3.3.3.1 Résultats de Hyperband

Le Tableau 3.3 résume les choix de hyperparamètres possibles et indique la combinaison ayant obtenu les meilleures performances. L'explication de chacun des hyperparamètres est présentée à l'Annexe 1. *Hyperband* a été exécuté en utilisant $R=81$, ce qui correspond au nombre maximal de ressources autorisées, et $\eta=3$, le nombre de branches considéré. La Figure 3.10 montre l'exécution de *Hyperband*. Chaque ligne des deux graphiques est un essai distinct. Le graphique de gauche présente l'évolution de chaque essai en fonction de la précision. Le graphique de droite montre l'évolution de chaque essai en fonction de la valeur du *DSC*. Au total, 81 différentes combinaisons d'hyperparamètres ont été testées pour un total combiné de 1875 *epochs*, ce qui représente 25,5 heures d'entraînement sur un ordinateur équipé de trois *GPU* Nvidia GeForce RTX 2080.

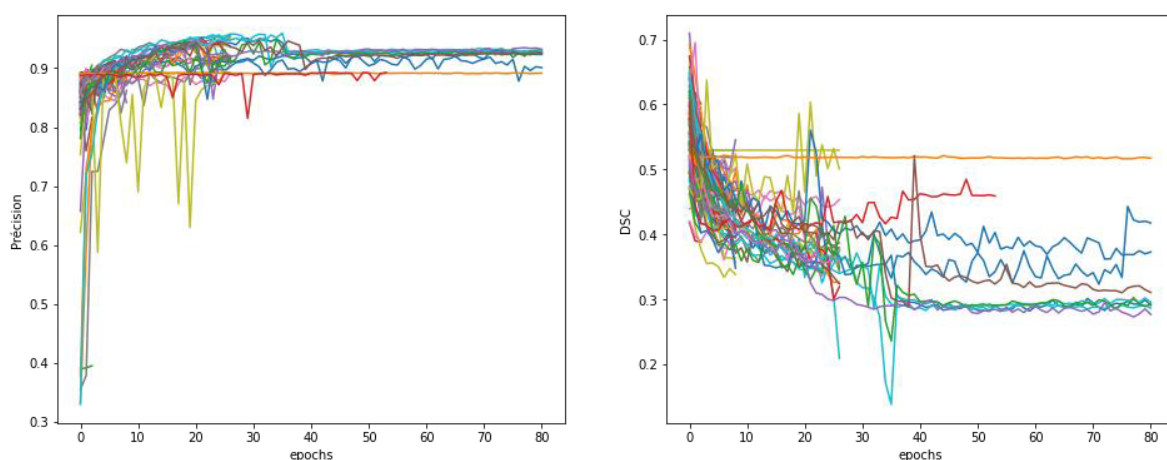


Figure 3.10 : Illustration de l'exécution de *Hyperband* avec $R=81$ et $\eta=3$

La meilleure combinaison d'hyperparamètres a obtenu un *DSC* de 86,6 et une précision de 95,9 %, tel qu'illustré à la Figure 3.10. Les résultats optimaux ont été obtenus avec *V-Net* comme modèle de segmentation, un taux *dropout* de 0,30 un *LR* de 0,00099363, un *weight decay* de 0,00210985, *AdamW* comme fonction d'optimisation, la fonction de perte *dice*, *ELU* comme fonction d'activation. Les résultats sont résumés dans le Tableau 3.3.

Tableau 3.3 : Résultats de *Hyperband* pour la sélection des hyperparamètres optimaux

Hyperparamètres	Sélection	Valeurs possibles
Modèle	<i>V-Net</i>	<i>U-Net</i> , <i>V-Net</i> , <i>DeepLabv3</i> , <i>DenseNet</i> et <i>ESPNet</i>
<i>Dropout</i>	0,30	Sans <i>dropout</i> , valeur de <i>dropout</i> de 0,1 à 0.7
<i>LR</i>	0,00099363	0.00008 à 1
<i>Weight decay</i>	0.00210985	10^{-2} à 10^{-10}
Fonction optimisation	<i>AdamW</i>	<i>Adadelata</i> , <i>Adaptative Gradient</i> , <i>Adam</i> , <i>Adamax</i> , <i>AdamW</i> , <i>Root Mean Square Propagation (RMSProp)</i> , <i>SGD</i>
Fonction de perte	<i>DSC</i>	<i>DSC</i> , <i>SoftDice</i> , <i>Generalized Dice</i> , <i>Focal</i> , <i>IoU</i> , <i>LogCoshDice</i>
Fonction d'activation	<i>ELU</i>	<i>ReLU</i> , <i>SeLU</i> , <i>ELU</i> , <i>LeakyReLU</i> , <i>Mish</i> , <i>Swish</i> , <i>aReLU</i> , <i>Slaf</i> , <i>Maxout</i> , <i>PReLU</i>

Normalisation	Groupe	<i>batch</i> , instance, groupe
---------------	--------	---------------------------------

Considérant la faible quantité de données d'entraînement disponible, il est peu surprenant que *V-Net* soit le modèle qui obtient les meilleurs résultats puisqu'il s'agit de l'un des modèles les plus légers. De plus, *V-Net* a fait ses preuves : il est en mesure de bien performer avec très peu de données d'entraînement [102]. Le *FCN* multitâche sélectionné est illustré à la Figure 3.11. On voit un exemple de la taille des *features maps* pour un intrant de taille (1, 80, 32, 80).

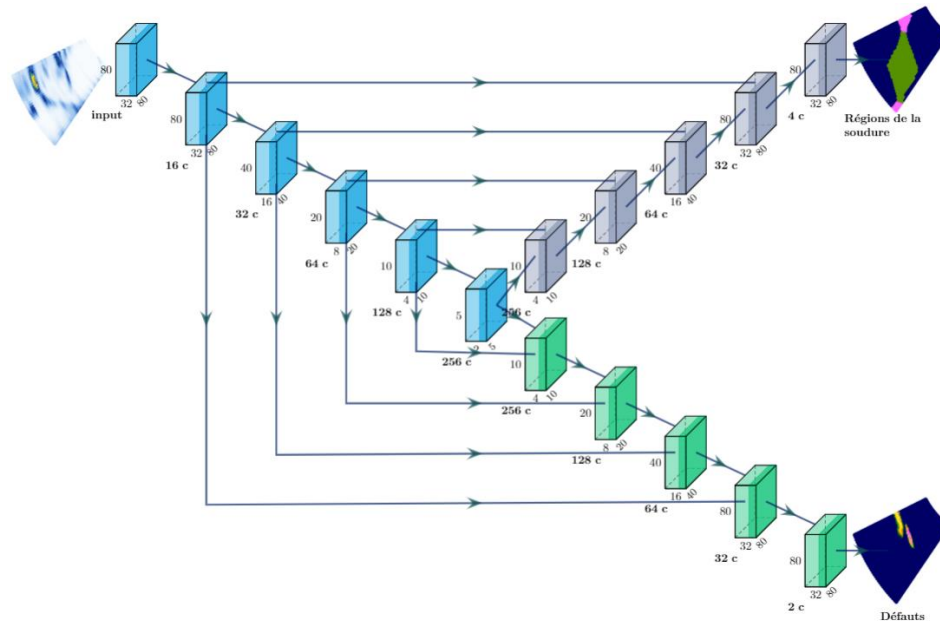


Figure 3.11 : Modèle de segmentation multi-objectif

L'utilisation de *dropout* et la normalisation par *batch* combinées peuvent entraîner une réduction des performances des RDN [200]. Cela résulte d'une modification de la variance (*variance shift*) occasionnée par *dropout* au moment de l'inférence. La normalisation par *batch* n'occasionne pas de modification de la variance. L'incohérence des variances dans *dropout* et la normalisation par *batch* provoque une instabilité numérique à l'inférence qui conduit à des prédictions erronées. Des stratégies de combinaison des deux techniques ont été développées. Il est donc recommandé d'utiliser une valeur de *dropout* faible ($<0,50$) et de ne pas utiliser *dropout* dans la partie *bottleneck* du réseau. Il est également possible de modifier *dropout* pour réduire la modification de la variance, ce qui augmente la performance des RDN même lorsque *dropout* est utilisée dans la partie *bottleneck*. La valeur optimale de *dropout* déterminée par *Hyperband* est de 0,30. Le *FCN* développé dans cette étude suit donc les recommandations proposées sur l'utilisation combinée de *dropout* et la normalisation par *batch*.

L'utilisation de technique de gestion du *LR* aide à augmenter la performance des RDN. L'utilisation de technique de gestion du *LR*, comme *CyclicLR*, *ReduceLROnPlateau*, n'a pas été testée par *Hyperband*. *CyclicLR* et *ReduceLROnPlateau* sont utilisées pour l'entraînement final du modèle, une fois que les hyperparamètres optimaux ont été trouvés. *Hyperband* a permis de trouver la valeur du *LR* et du *weight decay* qui maximise les performances de segmentation du modèle *FCN*. La valeur finale de *LR* est de 0,00099363 tandis que le *weight decay* est de 0,00210985.

La fonction d'optimisation optimale sélectionnée par *Hyperband* est *AdamW* qui est une fonction d'optimisation adaptative utilisant une nouvelle implémentation du *weight decay*. Il est important de noter que la régularisation l_2 et le *weight decay* sont équivalents pour *SGD*, mais ce n'est pas le cas pour les fonctions d'optimisation adaptative, comme *Adam*. Les implémentations communes des fonctions d'optimisation adaptative emploient simplement la régularisation l_2 , ce qui occasionne une réduction de la performance. Une simple modification, en découplant le *weight decay* des étapes d'optimisation, permet de retrouver la formulation originale de la régularisation *weight decay*. Ceci a pour effet d'augmentation significativement les performances de généralisation de *Adam* au même niveau que *SGD* pour un problème de classification d'images.

La fonction d'activation optimale sélectionnée par *Hyperband* est *ELU* (*Exponential Linear Unit*) qui est une amélioration de *ReLU* [201]. Son comportement est similaire à celui de *LeakyReLU*, tout en étant plus *smooth*. L'un des avantages de *ELU* est qu'il ne souffre pas du problème de neurones morts, comme *ReLU*, car *ELU* peut produire des valeurs négatives. Ce problème provient du fait que *ReLU* ne peut pas produire de valeur négative, ce qui fait que son gradient pour les sommes négatives est également zéro. Ainsi, lorsqu'un neurone devient négatif, ce neurone arrête d'apprendre. Un second avantage de *ELU* est la présence de deux zones distinctes dans son comportement : une zone de linéarité et une de non-linéarité. Ceci a pour avantage de pouvoir approximer autant des fonctions linéaire (*piecewise* par exemple) et non linéaire. Un troisième avantage de *ELU* est son efficacité algorithmique qui permet une réduction de la durée d'entraînement [202]. Cet avantage provient du fait que *ELU* peut avoir des valeurs négatives, ce qui fait tendre la moyenne d'activation des neurones près de zéro, réduisant la complexité algorithmique et augmentant la vitesse d'apprentissage. En somme, *ELU* possède une meilleure capacité d'apprentissage et de généralisation que *ReLU* et *LeakyReLU*. Il n'est pas nécessairement considéré comme la fonction d'activation de référence. Toutefois, les résultats de *Hyperband* justifient son utilisation dans cette étude.

La valeur optimale du *batch size* est limitée à 4 pour des raisons de mémoire. La taille des intrants est très importante, pouvant atteindre jusqu'à 60 millions de points. Or, le modèle optimal étant un *FCN*, il est possible de faire varier la taille des intrants en respectant un facteur de 8 pour chaque

dimension. Il a été déterminé que les résultats sont au mieux lorsque la compression des données est le plus faible possible.

Considérant le faible *batch size* utilisé, il est tout à fait prévisible que la normalisation par *batch* ne soit pas considérée comme étant le meilleur choix par *Hyperband*. En effet, la normalisation par *batch* réduit la valeur des intrants par la moyenne de la *batch* puis divise le résultat par l'écart-type de la *batch* [203-205]. Ensuite, le résultat est multiplié par un coefficient γ et un second coefficient β est additionné. Ce calcul est refait pour chaque *batch*. La normalisation utilisée par la meilleure configuration de *Hyperband* est la normalisation par groupe. Cette normalisation est spécifiquement conçue pour les modèles convolutifs comme le *FCN* utilisé dans cette étude [206]. Les auteurs de [206] montrent que la normalisation par *batch* introduit des problèmes. L'erreur augmente rapidement quand le *batch size* est petit, comme c'est le cas dans cette étude. Ceci est causé par une estimation erronée des statistiques de la *batch*. Ce problème a pour effet de réduire l'utilité et l'efficacité de la normalisation par *batch* pour les modèles profonds. La normalisation par groupe divise les canaux en groupes et calcule la moyenne et la variance de chaque groupe pour effectuer la normalisation. Chaque filtre d'un même groupe devrait se spécialiser dans une même tâche. La normalisation par groupe est donc invariable au *batch size* et son efficacité est stable pour une plage de *batch size* importante. La normalisation par groupe surpasse la normalisation par *batch* de 10,6 % en utilisant un *batch size* de 2 et performe de façon comparable lorsque le *batch size* est plus important. Il est aussi intéressant de noter qu'il surpasse aussi les autres types de normalisation, dont la normalisation par instance. Finalement, la normalisation par groupe a réussi à démontrer sa supériorité dans la segmentation d'images à l'aide la base de données coco [206]. Ces explications ainsi que les résultats de *Hyperband* justifient amplement l'utilisation de la normalisation par groupe dans cette étude.

Il est à noter que pour comparer les fonctions de pertes, il est nécessaire d'utiliser une (ou plusieurs) mesure universelle comparative. Les mesures de performance communément utilisée pour évaluer la performance d'un modèle de segmentation sont la précision pixel par pixel (ou voxel par voxel en 3D), le *DSC* et le score F1 [207]. Le calcul de la précision va comme suit : tous les pixels d'une image sont évalués, et le nombre de pixels correctement identifiés est divisé par le nombre de pixels total de l'image. Le *DSC* mesure la similitude de deux échantillons. Le *DSC* équivaut à deux fois le nombre d'éléments communs aux deux échantillons divisés par le nombre d'éléments dans chaque ensemble. Le score F1 est une mesure de performance pour évaluer la performance des modèles de classification. En segmentation 3D, elle peut être utilisée en considérant chaque voxel comme une classification, c'est-à-dire à quelle classe appartiennent chaque voxel. Le score F1 est égal à :

$$Score\ F1 = \frac{2 * précision * rappel}{précision + rappel} \quad (3.1)$$

Le score F1 peut aussi être formulé à partir des éléments de la matrice de confusion :

$$\text{Score F1} = \frac{\text{Vrai Positif}}{\text{Vrai Positif} + \frac{1}{2}(\text{Faux Négatif} + \text{Faux Positif})} \quad (3.2)$$

Un score F1 de 50 % équivaut à une bonne prédiction (vrai positif) pour deux mauvaises prédictions (Faux Négatif + Faux Positif). Lors de la classification multi-classe, le score F1 est calculé pour chaque classe. Le score F1 pondéré en fonction de l'importance de chacun est utilisé. Il convient de préciser que le score F1 et le *DSC* sont étroitement liés. En fait, *DSC* et le score F1 sont identiques, si on ne prend en compte la pondération de chaque classe comme c'est le cas dans cette recherche.

Différentes fonctions de perte sont évaluées afin de trouver la fonction de perte optimale. Pour ce faire, chaque fonction de perte, présentée au Tableau 3.4, est utilisée pour un entraînement complet et trois indicateurs de performances sont calculés pour effectuer une comparaison quantitative des fonctions de perte. Les hyperparamètres choisis pour effectuer les entraînements sont les hyperparamètres optimaux sélectionnés par *Hyperband*. *Earlystopping* avec une patience de 40 *epochs* a été utilisé pour chaque entraînement. Les résultats de cette expérience sont présentés au Tableau 3.4. Les performances en rouge sont les pires performances par colonne alors que les performances en vert sont les meilleures performances par colonne.

Tableau 3.4 : Comparaison des fonctions de perte

Fonction de perte	DSC (%)			Précision (%)			Score F1 (%)			Moy.
	Soudure	Défaut	Total	Soudure	Défaut	Total	Soudure	Défaut	Total	
<i>Generalized Dice</i>	67,9	80,5	74,2	90,2	99,7	95,0	88,0	64,1	76,0	81,7
<i>Sensitivity-Specifity</i>	60,7	65,1	62,9	87,7	98,6	93,1	88,3	27,8	58,2	71,4
<i>Soft Dice</i>	51,3	78,1	64,7	81,5	99,8	90,6	79,8	57,6	68,7	74,7
<i>IoU</i>	55,9	53,7	54,7	77,8	95,8	86,8	67,8	9,8	38,8	60,1
<i>Focal-Tversky</i>	65,5	80,3	72,9	81,8	99,9	90,8	86,8	65,0	75,9	79,9
<i>Log Cosh Dice</i>	46,3	47,0	46,7	77,2	88,9	83,0	65,9	1,7	33,0	54,4
<i>Asymmetric</i>	79,0	81,3	80,2	92,2	99,8	96,0	93,4	67,4	80,4	85,5
<i>Dice</i>	56,5	51,5	54,0	73,4	92,3	83,8	83,0	6,3	44,6	60,8
Moyenne	60,4	67,2	63,8	82,7	96,8	89,9	81,6	37,5	59,5	71,0

On remarque au Tableau 3.4 que les meilleurs résultats ont été obtenus avec les fonctions de perte *Asymmetric* (moyenne de 85,5 %) et *Generalized Dice* (moyenne de 81,7 %) alors que les pires résultats proviennent de *Log Cosh Dice* (moyenne de 54,4 %). Il est peu surprenant que les meilleurs résultats aient été obtenus avec ses fonctions de perte, car les deux sont des améliorations à la fonction de perte *dice* afin de régler les problèmes de déséquilibre des données. Il est important de rappeler que les données disponibles dans cette recherche sont très déséquilibrées. En effet, le

matériel de base représente 76,2 % des données, la soudure en V 10,5 %, la soudure en double-V 8,2 % et la réflexion incomplète 5,1 %. Dans le cas des défauts de soudure, il y a un défaut dans seulement 0,28 % des voxels alors que 99,72 % des voxels n'en ont pas.

On note également que les indicateurs de performance *DSC* et précision sont plus élevés pour les défauts que les soudures. On observe le contraire pour le score *F1*. Il est surprenant d'observer une telle divergence entre ces indicateurs. Il semble que pondérer le score *F1* par l'importance de chacun permet de mieux évaluer la capacité du modèle à segmenter les défauts que *DSC* et la précision.

Pour la suite de cette recherche, la fonction de perte asymétrique est utilisée, car elle est celle qui offre les meilleures performances de segmentation, comme démontré au Tableau 3.4.

3.3.3.2 Augmentation des données

L'un des plus importants problèmes en END est le manque de données représentatives [7]. En effet, il est difficile d'obtenir en quantité suffisante des données annotées considérant le faible nombre de défauts par soudure. Certains défauts plus rares sont ainsi disponibles en très faible quantité, ce qui est problématique au niveau de l'entraînement de modèles d'IA à apprentissage supervisé, tel que le modèle de segmentation multi-objectif. Pour pallier ce problème, les études ont recours à l'augmentation des données. Les techniques les plus fréquentes en vision par ordinateur sont la rotation, la translation, la modification des couleurs et l'ajout de bruits [208]. Il existe des techniques d'augmentation des données spécifiques aux scans *PAUT*. Ces techniques reproduisent des phénomènes présents dans les scans. La Figure 3.12 montre quelques-unes des techniques utilisées dans cette étude.

Le premier trio d'images montre un exemple de S-scan brute et les annotations correspondantes de la soudure et des défauts. Le deuxième trio d'image illustre une rotation de 180° (*flip*) afin de reproduire un scan pris dans le sens opposé (90° vs 270°). Le troisième trio montre un exemple où un certain nombre de *beams* (pour un maximum de 5), déterminé au hasard, a été supprimé. Le quatrième trio de S-scans représente la combinaison de la rotation de 180° et la suppression aléatoire de *beams*. Selon [209], une rotation de 180° procure une augmentation de 1,8 % de précision top 1 et 2,9 % de précision top 5 en comparaison avec un entraînement de base, ce qui est une augmentation importante.

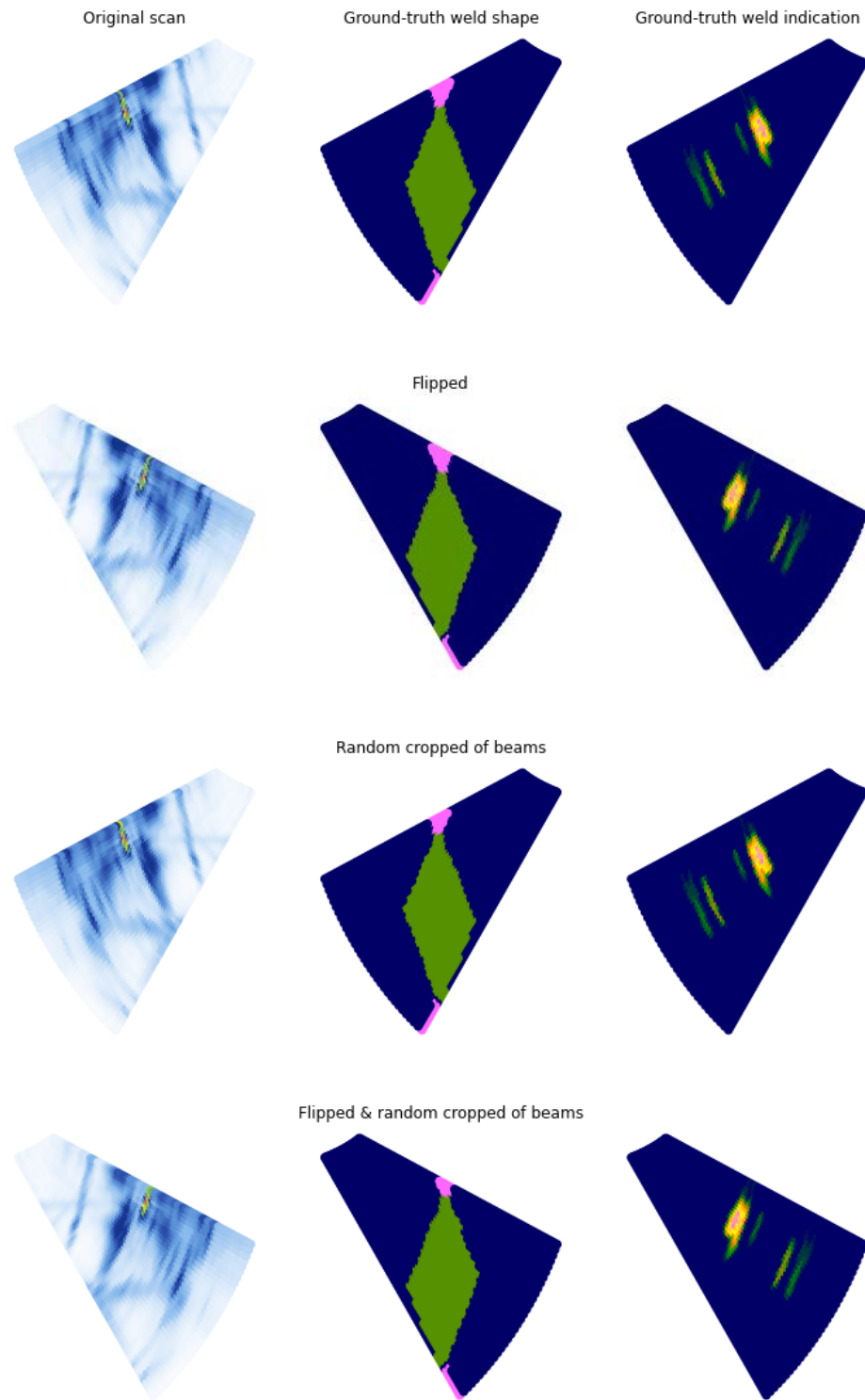


Figure 3.12 : Exemples de technique d'augmentation des données utilisées

Des techniques d'augmentation des données utilisées en image 2D et 3D sont également utilisées. Ces techniques proviennent de la famille de l'injection de bruit. Il s'agit de correction gamma et de

l'ajout de bruit aléatoire (sel et poivre, gaussien et *speckle*) et l'ajout de gain aléatoire. L'ajout de bruit aléatoire consiste en l'injection d'une matrice de valeur aléatoire tirée d'une distribution choisie. L'ajout de gain aléatoire est une technique spécifique au domaine des END qui consiste en l'augmentation de l'amplitude globale du scan selon certains critères. Ces techniques aident les CNNs à apprendre des caractéristiques plus robustes [208]. La Figure 3.13 illustre les différentes techniques d'augmentation des données basées sur l'injection de bruit. Dans l'ordre horaire, on y voit : (1) le S-scan original, (2) l'ajout de bruit sel et poivre, (3) l'ajout de bruit gaussien, (4) l'ajout de gain aléatoire, (5) la correction gamma et (6) l'ajout de bruit *speckle*.

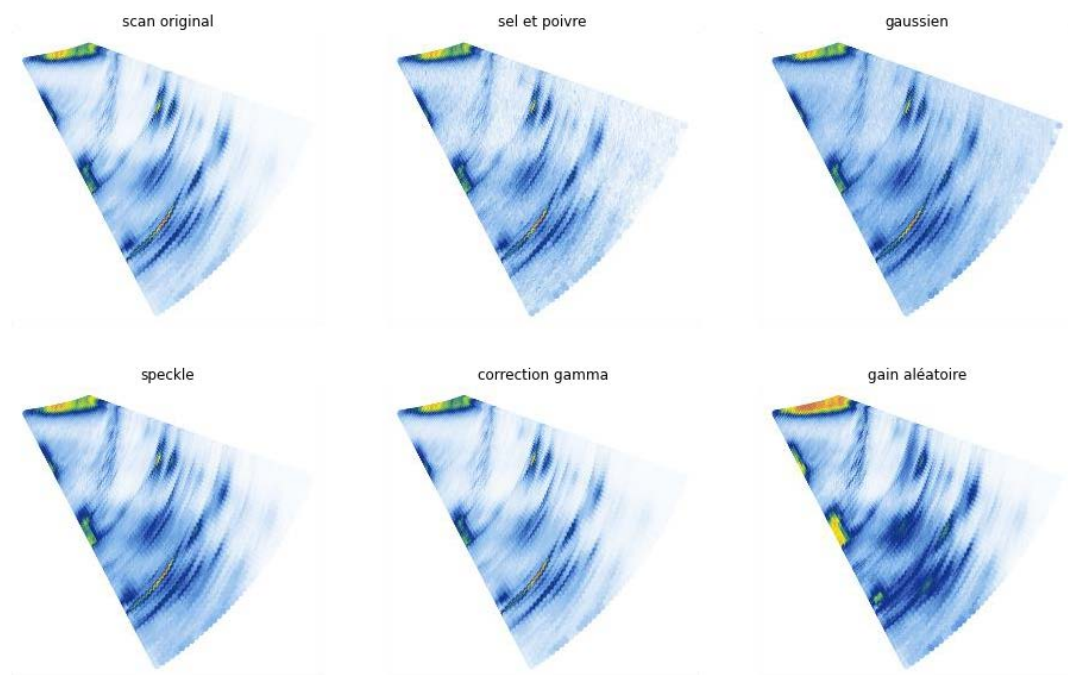


Figure 3.13 : Exemples de techniques d'augmentation des données utilisées de la famille de l'injection de bruit

Bien qu'il existe plusieurs autres techniques d'augmentation des données, comme l'effaçage aléatoire de *patch* (*random erasing*), ces techniques ne s'appliquent pas au problème de segmentation d'objets, particulièrement de petite taille comme les défauts de soudure. Elles ne sont donc pas utilisées ici.

3.4 Modélisation géométrique en trois dimensions

Le modèle de segmentation de la soudure développé dans cette recherche comporte une deuxième branche qui génère une carte de segmentation binaire permettant de savoir si chaque voxel inclut ou non un défaut. Cette branche est entraînée à partir de données de soudure annotées par un expert

de Ondia (*PAUT* niveau III). Chaque voxel est identifié comme ayant une absence ou une présence de défaut. Il est alors possible, tout comme pour la soudure, de créer un modèle géométrique 3D du défaut et d'en faire l'affichage.

Les voxels identifiés précédemment par le *FCN* permettent de procéder à une modélisation des différentes parties appartenant à la soudure. Chaque classe d'objets identifiée par le *FCN* peut être modélisée pour générer un modèle géométrique 3D représentatif de la soudure. La construction de ce modèle utilise une technique de modélisation 3D traditionnelle et une technique basée sur l'apprentissage profond. La méthode ayant les meilleurs résultats est finalement choisie. Le choix final du processus est fait à partir des résultats et de certains critères de performance. Selon [173], l'algorithme de reconstruction de surface traditionnelle offrant la meilleure qualité et efficacité est *SPR* ce qui justifie son expérimentation dans cette recherche. La technique d'IA testée est *P2S* considérant qu'il s'agit du modèle ayant les meilleures performances et étant la plus robuste aux bruits [173]. Le processus de reconstruction de la surface de la soudure et des défauts est illustré à la Figure 3.14. Ce processus est répété pour chaque méthode de reconstruction de surface.

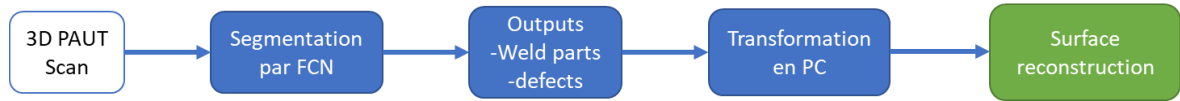


Figure 3.14 : Processus de reconstruction de la surface de la soudure et des défauts

Afin de mesurer l'erreur de reconstruction pour chaque méthode testée, la surface reconstruite et la surface du *GT* sont échantillonnées pour sélectionner 10 000 points et la distance de chanfrein (d_{ch}) entre les deux échantillons sont calculés :

$$d_{ch}(A, B) = \frac{1}{|A|} \sum_{p_i \in A} \min_{p_j \in B} \|p_i - p_j\|_2^2 + \frac{1}{|B|} \sum_{p_i \in B} \min_{p_j \in A} \|p_i - p_j\|_2^2 \quad (3.3)$$

Où A et B sont les ensembles de points échantillonnés des deux surfaces. La distance de chanfrein pénalise autant les faux négatifs (parties manquantes) que les faux positifs (parties en trop).

Les modèles géométriques de la soudure et des défauts doivent être affichés sous une représentation 3D simplifiée. Il est à noter que l'affichage sera intégré au logiciel infonuagique de Ondia. Il est nécessaire que l'affichage soit interactif, permettant notamment le zoom et la rotation du modèle lors de l'analyse. Les modèles géométriques sont générés en Python à partir des segmentations. Ils sont enregistrés dans un format reconnu, par exemple : .STL ou .PLY, et des libraires graphiques compatibles effectuent la visualisation interactive [210]. Il est à noter que les critères de performance

pour la sélection de la librairie graphique et du format d'enregistrement sont la taille des fichiers générés, la qualité de la visualisation produite et la qualité de l'infrastructure de visualisation.

Afin de faire une comparaison juste entre les deux techniques de reconstruction de surface, il est nécessaire de déterminer les hyperparamètres optimaux de *P2S* et de faire ensuite un entraînement complet en utilisant ces hyperparamètre. En effet, tout comme le modèle de segmentation multi-objectif, le modèle de reconstruction de surface possède des hyperparamètres qui ont un effet sur la qualité de l'extrait du modèle.

Certains hyperparamètres spécifiques à *P2S* ont un impact considérable sur les performances du réseau, autant au niveau de la qualité des résultats obtenus, du délai d'exécution que du besoin en mémoire. La liste inclut, entre autres, la taille du réseau, le nombre de points par *patch* et le nombre de *patch* par soudure. La taille du réseau fait référence à la taille maximale du sous-réseau *QSTN* de *P2S* et de la première couche des encodeurs. La taille du réseau a un effet direct sur la taille de l'espace latent de l'auto-encodeur. Or, l'espace latent a un effet sur la qualité des prédictions [211]. En fait, un auto-encodeur stocke toutes les caractéristiques pertinentes des données dans l'espace latent, afin qu'il y ait suffisamment d'informations dans cette forme comprimée pour que le modèle puisse les reconstruire avec précision. Une étude sur l'effet de la taille de l'espace latent sur la qualité des images synthétiques créées par un *Generative Adversarial Networks* a permis de mieux comprendre la complexité de cet hyperparamètre [211]. L'étude a prouvé que les *Generative Adversarial Networks* sont capables de générer des images de qualité acceptable avec un espace latent significativement plus petit (16) que les formats couramment utilisés qui sont de 100 ou 512. Elle a aussi réussi à déterminer qu'après un certain point, l'augmentation de la taille de l'espace latent n'a pas d'effet visible sur la qualité des images créées et n'a pas d'effet quantitatif sur la capacité du modèle à mieux généraliser. Il y a donc une plage de valeur optimale pour la taille de l'espace latent qui se situe entre 16 et 256 dans l'étude. Il est donc nécessaire d'ajouter la taille de l'espace latent à *Hyperband* afin de trouver la taille optimale ou la plage de valeur optimale permettant d'obtenir des résultats quantitatifs de qualité.

La variable points par *patch* fait référence au nombre de points extraits de chaque *patch*. Plus il y a de points extraits par *patch*, plus la représentation de cette *patch* est précise. En revanche, cela augmente la quantité de mémoire nécessaire et le nombre de calculs à effectuer. Il est donc nécessaire de trouver le bon compromis.

La variable patches par soudure correspond au nombre de patches considéré par soudure modélisée. Plus il y a de *patches* par soudure, plus la représentation globale de la soudure est précise. Toutefois, cela augmente la quantité de mémoire nécessaire et le nombre de calculs à effectuer. Il est important de trouver le bon ajustement.

Hyperband est donc appliqué à *P2S* afin de trouver les hyperparamètres suivants : la taille du réseau, le nombre de points par *patch* et le nombre de *patch* par soudure, fonction d'optimisation, fonction d'activation *LR* et *weight decay*.

3.5 Comparaison périodique des soudures

Dans le cycle de vie d'un assemblage mécanique, une soudure déjà déclarée conforme (connue par les analystes) est souvent appelée à être scannée une nouvelle fois dans le cadre de son suivi périodique (par exemple une fois par année). De nouvelles données ultrasonores de la même soudure sont donc envoyées pour être analysées. Il faut alors s'assurer que le nouveau fichier correspond à une soudure déjà connue. Cependant, même si la soudure n'avait absolument pas changé (vieilli) entre les deux scans, ces derniers ne donneraient jamais deux fichiers identiques pour les raisons suivantes :

1. Le technicien peut avoir changé certains paramètres des appareils ultrasons entre les deux scans (par exemple les fréquences, les puissances, l'utilisation d'un scanneur différent, etc.)
2. Même si les paramètres des appareils demeurent identiques, les coordonnées (X, Y, Z, amplitude) de chaque point vont nécessairement varier d'un scan à l'autre à cause de la trajectoire différente du scanneur (il est impossible pour un technicien de reproduire une trajectoire parfaitement identique pour deux scans de la même soudure)

La Figure 3.15 montre un exemple d'une même soudure issue de deux *PAUT* différents. La soudure de gauche a été prise à 90° tandis que celle de droite à 270° . On peut distinguer plusieurs éléments correspondants entre les deux scans comme les défauts de soudure. Il existe toutefois des éléments qui ne se répètent pas d'une soudure à l'autre. Ce phénomène provient du fait que les deux scans ne proviennent pas du même côté de la soudure. Ainsi, certains éléments sont reflétés plus facilement d'un côté que de l'autre.

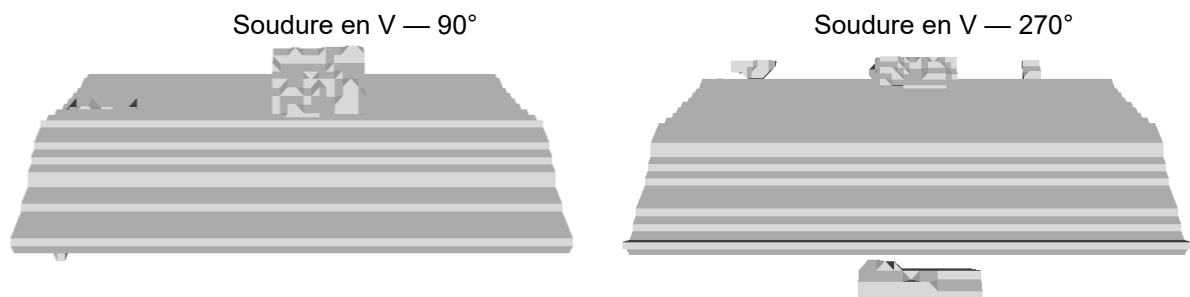


Figure 3.15 : Exemples d'une même soudure issue de deux scans différents

Pour assurer le suivi périodique d'une soudure, il devient donc nécessaire de faire appel à des algorithmes de recherche des *ANN* afin de trouver les possibles associations (c.-à-d. retracer des scans antérieurs de la même soudure) [212]. Contrairement à l'objectif a), on ne désire pas ici vérifier si un même fichier a déjà été soumis, mais plutôt si c'est la même soudure qui a été scannée. On désire donc obtenir la signature de la soudure. Une avenue des plus intéressante consiste à capitaliser sur les identifications géométriques issues des objectifs spécifiques b) et c). En effet, pour générer une identité unique (signature) d'une soudure, on peut certainement analyser certains de ses paramètres géométriques. L'idée consiste donc à utiliser les informations du modèle géométrique 3D comme intrant à l'algorithme de recherche des *ANN*. Plus spécifiquement, des points peuvent être extraits de la surface géométrique et être utilisés comme intrants. La méthode de recherche des *ANN* permet alors de retrouver des candidats qui sont potentiellement issus de la même soudure. À l'aide d'une mesure de similarité, il est alors possible de connaître la ressemblance de chaque candidat. Le(s) candidat(s) ayant une ressemblance supérieure à un certain seuil seront considérés comme issus de la même soudure.

Le processus de reconnaissance des modèles géométriques est illustré à la Figure 3.16. On constate que ce processus est similaire à celui de la Figure 3.7, à la différence que l'extrant de ce processus est la décision à savoir s'il s'agit d'une même soudure (modèle géométrique) ou non. Il est à noter que les actions sont toutes légèrement différentes de celles du processus de détection des doublons. Chaque action est adaptée pour augmenter la capacité de reconnaissance de processus.

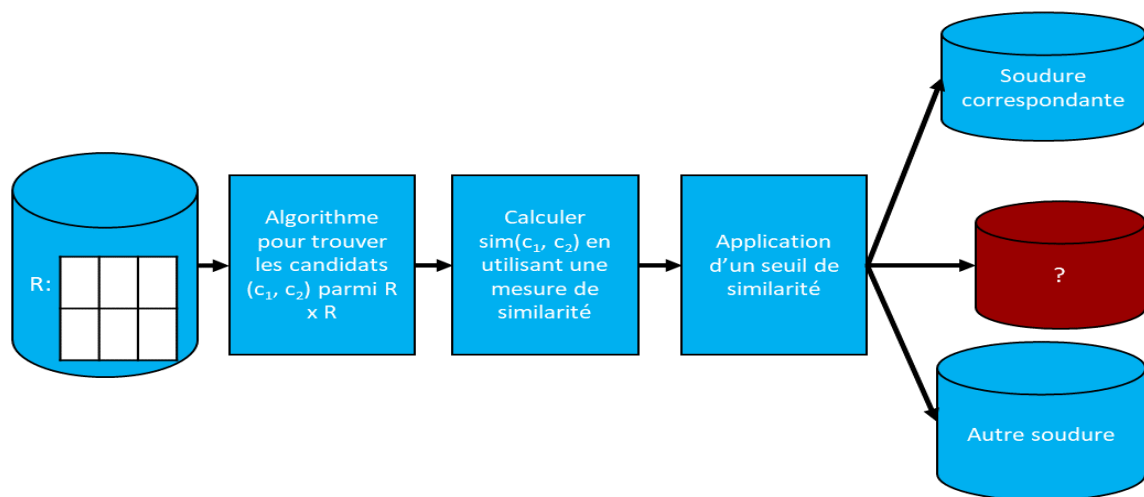


Figure 3.16 : Processus de reconnaissance des modèles géométriques 3D

Il est nécessaire de déterminer les trois éléments cruciaux à la bonne exécution de ce processus : (1) un algorithme de *ANN* qui fonctionne bien avec des données en 3D issues d'une surface reconstruite (2) une mesure de similarité qui caractérise efficacement la similitude de deux modèles

géométriques et (3) un seuil fiable permettant de réellement déterminer si deux modèles géométriques sont issus de la même soudure.

Le choix de la méthode de recherche des *ANN* s'arrête donc sur *LSH* pour la raison principale que cette technique est déjà sélectionnée pour faire la détection des doublons de *PAUT*, ce qui est un grand avantage pour son implémentation dans le logiciel infonuagique de Ondia et pour sa maintenance éventuelle. Une seconde raison qui explique ce choix est que *LSH* est capable de prendre des intrants de taille importante et de bien fonctionner dans des bases de données de grande taille. L'efficacité algorithmique de *LSH* lui permet de retrouver les candidats en quelques secondes seulement alors que ce serait beaucoup plus long pour les méthodes basées sur les graphes. Il est simplement nécessaire de modifier le processus pour l'adapter à la reconnaissance des modèles géométriques des soudures, tel qu'illustré à la Figure 3.16.

Le second élément à déterminer est la mesure de similarité. Il est essentiel d'utiliser une mesure de similarité qui quantifie bien la ressemblance entre deux modèles géométriques. La mesure de similarité la plus utilisée pour la comparaison de modèles géométriques 3D et de nuage de points est la distance de chanfrein, qui correspond au coût minimal de tous les chemins de longueur finie entre les deux nuages de points [172]. La distance de chanfrein est considérée comme la mesure de référence, notamment en apprentissage profond. Plus précisément, la distance de chanfrein correspond à la somme quadratique de la distance minimale de chaque point des deux nuages de points.

Il existe d'autres mesures de distance comme la distance de Wasserstein. Mathématiquement, la distance de Wasserstein (WD) est formulée ainsi :

$$WD(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n f_{i,j} d_{i,j}}{\sum_{i=1}^m \sum_{j=1}^n f_{i,j}} \quad (3.4)$$

Où il est assumé que P a m clusters et Q a n clusters. $f_{i,j}$ est le flow entre p_i et q_j . $d_{i,j}$ est la distance entre le cluster p_i et q_j . Cette distance est largement utilisée en IA. Les *Wasserstein Generative Adversarial Networks* sont un type de *Generative Adversarial Networks* utilisant la distance Wasserstein comme fonction de perte qui surpasse les performances des *Generative Adversarial Networks* traditionnels [213].

La distance de Hausdorff est une mesure de l'éloignement de deux sous-ensembles d'un espace métrique sous-jacent [214, 215]. Elle se définit comme ceci :

$$DH(P, Q) = \max\{\sup_{p \in P} d(p, Q), \sup_{q \in Q} d(P, q)\} \quad (3.5)$$

Où *sup* représente le supremum (c.-à-d. borne supérieure). La distance de Hausdorff est couramment utilisée pour mesurer la similarité de maillage 3D, notamment pour mesurer l'erreur géométrique de simplification entre un maillage original et le maillage simplifié. Elle est également utilisée en IA comme fonction de perte [216].

Il est également possible de déterminer la similarité de deux nuages de points à l'aide d'une combinaison de différents indicateurs, comme suggéré dans [190]. Dans cette étude, la comparaison des nuages de points est effectuée en quelques étapes : échantillonnage des deux surfaces, alignement, segmentation en trois parties, calcul de la mesure de similarité et mise en commun des mesures de similarité afin d'obtenir la mesure de similarité globale. La mesure de similarité individuelle utilisée est une combinaison de la mesure *low-rank*, qui est une mesure basée sur l'analyse en composantes principales, la mesure de kurtosis, qui mesure la similarité en termes de distribution des probabilités de chaque point individuel et la mesure de corrélation qui se concentre sur la correspondance des points dans l'espace géométrique. Il faut noter que cette mesure est d'abord élaborée pour le *watermarking*. Cette technique n'est pas retenue considérant qu'elle implique des étapes superflues (par exemple, alignement, segmentation, mise en commun), ce qui aurait pour conséquence de complexifier et alourdir le processus de comparaison périodique.

Les mesures de similarité Chanfrein, Wasserstein et Hausdorff sont testées afin de déterminer laquelle quantifie le mieux la similarité des modèles géométriques 3D. Une comparaison quantitative permet de déterminer quelle mesure est choisie.

Le troisième élément à déterminer est la valeur du seuil au-delà duquel un modèle géométrique est considéré comme provenant de la même soudure que la requête. Il va de soi que la valeur du seuil dépend de la mesure de similarité choisie. Ainsi, la valeur du seuil est déterminée après la sélection de la mesure de similarité à l'aide d'expériences.

En résumé, le processus de reconnaissance des modèles géométriques pour comparaison périodique débute avec un intrant, soit un nouveau modèle géométrique. La signature de cet intrant est alors calculée. Une fois la signature obtenue, une requête à la base de données permet de retrouver la liste des *ANN*. La similarité de chaque *ANN* avec le modèle géométrique intrant est calculée. Une soudure correspondante est trouvée si la valeur de la mesure de similarité est supérieure au seuil prédéterminé. Ce processus permet de comparer les modèles géométriques 3D d'une même soudure à des moments différents pour des fins de suivi.

4 RESULTATS ET DISCUSSION

Ce chapitre présente les résultats associés à chacun des objectifs spécifiques de cette étude. Une discussion sur les résultats, leur mise en pratique et leur implication est présentée pour conclure ce chapitre.

4.1 Détection des doublons

Cette section démontre l'efficacité de la détection des doublons en l'appliquant à la base de données privée discutée en section 3.1. Trente *ED* ont été intentionnellement ajoutés à la base de données pour tester l'efficacité des méthodes de détection de doublons. Les *ND* ont été générés en scannant une même soudure sous différentes conditions. Les modifications apportées aux paramètres de l'appareil *PAUT* incluent la focalisation, les filtres, le nombre d'échantillons, la longueur de balayage et la résolution angulaire différente.

4.1.1 Détection des ED

Les 30 *ED* ajoutés intentionnellement à la base de données ont tous été détectés avec succès en suivant le processus illustré à la Figure 4.1.

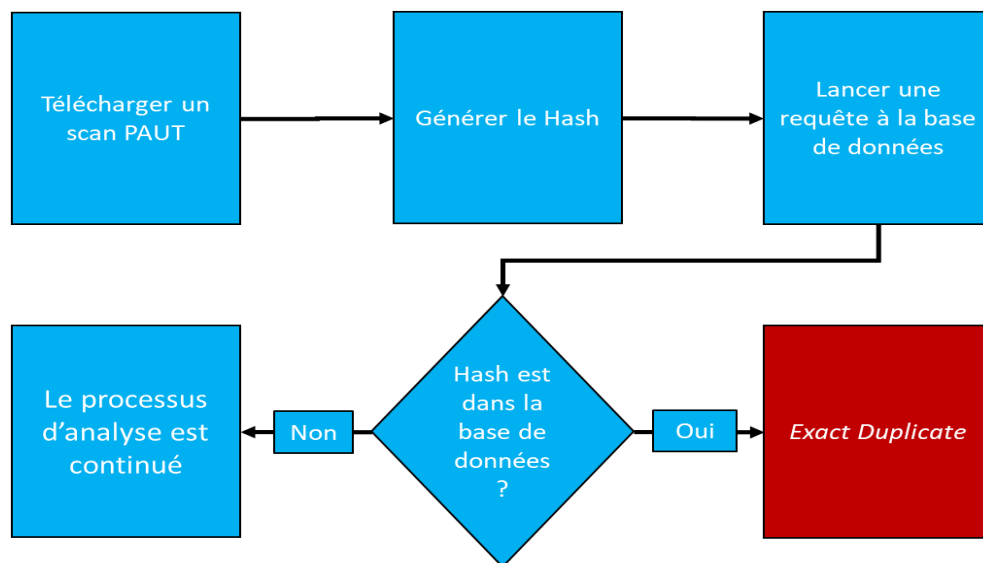


Figure 4.1 : Processus de détection des *ED*

Le Tableau 4.1 montre cinq exemples de scan ultrasonore et leur valeur de hachage associée.

Tableau 4.1 : Exemples de hash associés aux fichiers scannés

	hash	csv_file_path
396	d4dc17d0360b939e477bb635e81a0efb19a92db1c8790c...	UUT_DV25mm_OL_comp_21_indexdrift.opd2.ondia
397	1c14fc8b79832e1ea1d55ec5e013777d3a8afdb78a6822...	UUT_DV25mm_OL_SECTweld_22.opd1.ondia
398	c8a360d348b75109c92df5f78693e92e9d73785a6d988a...	UUT_DV25mm_OL_SECTweld_22.opd2.ondia
399	bdc7a34480debd93dd2d1f48419b332c7eaf7327b89449...	UUT_DV25mm_OM_comp_28.opd1.ondia
400	ffb86ce6722a806c0560f945be2452fb8fd4ca0a432986...	UUT_DV25mm_OM_comp_28.opd2.ondia

Pour effectuer le processus illustré à la Figure 4.1 pour un seul fichier, il faut environ 0,20 seconde avec un ordinateur équipé d'un processeur de 1,90 GHz. Ainsi, le processus de détection des *ED* est très efficace. La Figure 4.2 illustre un exemple de *ED*.

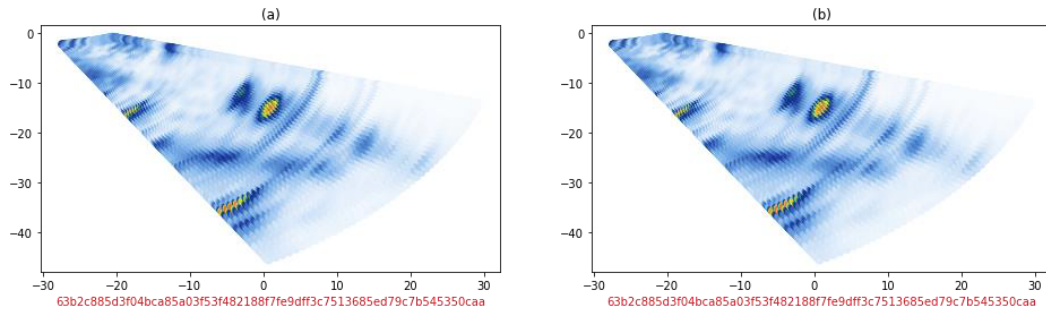


Figure 4.2 : Exemple de *ED* et leur *hash* respectif

Il est possible de constater que visuellement ces deux S-scans sont similaires. Le *hash* de chacun de ces scans permet de confirmer que ce sont bien des *ED*.

4.1.2 Détection des *ND*

L'évaluation de la méthode de détection des *ND* a été réalisée à l'aide de quatre expériences différentes. Pour chacune des expériences, la même méthodologie est utilisée. La méthode est d'abord appliquée à tous les scans pour créer la base de données. Ensuite, la méthodologie de la Figure 4.3 est suivie. Tous les scans passent de manière itérative à travers ce processus, ce qui donne une liste de *ANN* et leur distance en fonction de toutes les mesures de similitude choisies pour tous les scans intrants. Enfin, tous les candidats au-dessus (ou en dessous) du seuil sont considérés comme *ND*.

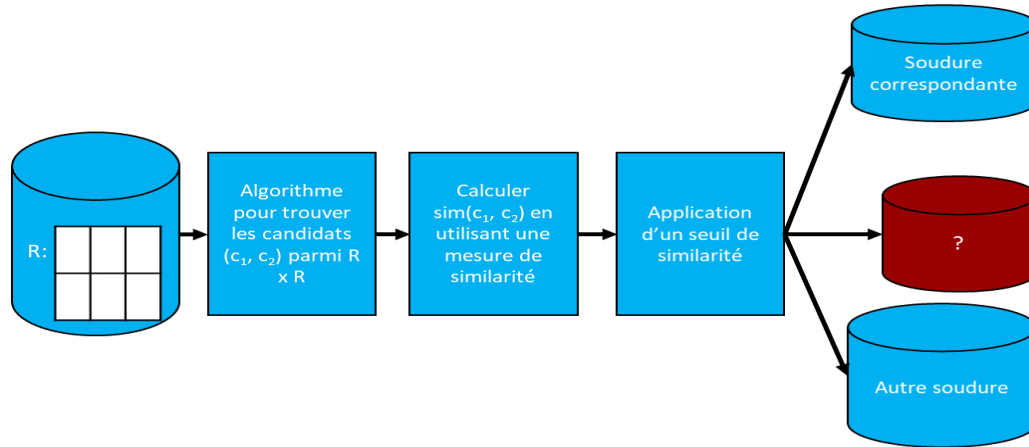


Figure 4.3 : Processus de détection des doublons utilisé, adapté de [51]

4.1.2.1 Capacité de détection

La première expérience consiste à tester la capacité de détection de la méthode sur des *ND* spécialement identifiés. Ce premier test commence par diviser l'ensemble de données en deux sous-ensembles différents : l'ensemble d'entraînement et l'ensemble de test. L'ensemble d'entraînement contient la grande majorité des scans *PAUT* tandis que le second sous-ensemble n'est composé que de 15 fichiers *ND* générés en scannant certaines soudures à plusieurs reprises à partir de différents paramètres de l'appareil d'acquisition (*PAUT*). Les modifications apportées aux paramètres d'acquisition comprennent des modifications du focus, des filtres, du nombre d'échantillons, de la longueur du balayage et de la résolution angulaire. Par conséquent, chaque scan de test a entre un et sept doublons dans l'ensemble d'entraînement. Un détecteur de *ND* efficace devrait être capable de détecter tous ces *ND*.

LSH est d'abord appliqué à l'ensemble d'entraînement pour créer la base de données. Ensuite, des requêtes pour tous les fichiers de test sont générées. Les résultats sont résumés à la Figure 4.4. Le nombre d'*ANN* au-dessus du seuil et la performance de détection (précision) à différentes valeurs de τ sont illustrés pour aider à sélectionner la valeur optimale du seuil.

La précision de la méthode de détection est maximale à $\tau = 0,86$ à $0,92$ et commence à diminuer pour atteindre une précision minimale de 67 % à $\tau = 0,96$. La performance optimale a été obtenue à $\tau = 0,92$ puisque la méthode détecte tous les *ND* tout en ayant le plus petit nombre d'*ANN* à environ 9. La Figure 4.4 montre que le nombre d'*ANN* diminue presque linéairement avec l'augmentation de τ alors que la précision ne commence à diminuer qu'à $\tau = 0,92$, mais le taux de diminution est important. Le choix de la valeur de τ affecte à la fois la précision de détection et l'efficacité algorithmique de la méthode. Le meilleur scénario serait une précision élevée tout en ayant un faible

nombre d'ANN au-dessus du seuil, permettant ainsi d'identifier uniquement les *ND* réels. Pour la suite de cette recherche, une valeur de 0,92 est utilisée pour τ . Cette valeur a été validée visuellement par un analyste certifié (*PAUT* niveau III par *Certification Scheme for Weldment Inspection Personnel*).

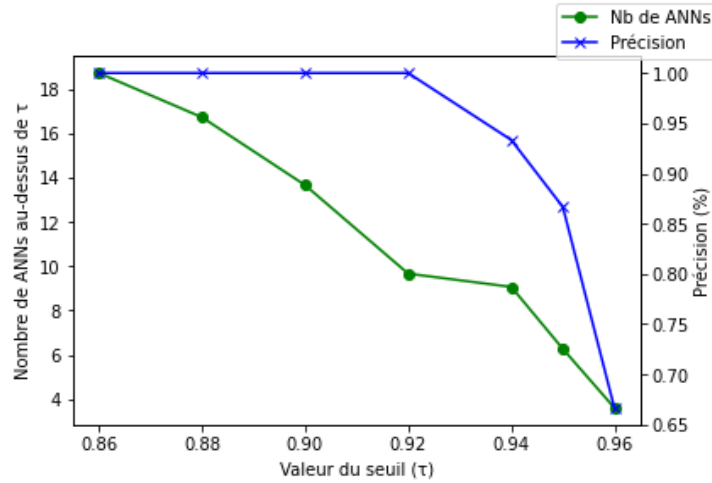


Figure 4.4 : Performance de détection des *ND* à différentes valeurs de τ

Il est à noter que la valeur de τ doit être validé fréquemment par un expert tel qu'un analyste certifié. En effet, à mesure que de nouvelles données sont ajoutées à la base de données, il est nécessaire de valider la valeur de τ , car la distribution des données peut avoir suffisamment changée pour produire des décisions erronées.

4.1.2.2 Précision de LSH

La deuxième expérience évalue la précision de différentes configurations de *LSH*. Le nombre d'ANN identifié par la méthode est également considéré étant donné son importance sur l'effort de calcul et le temps nécessaire à l'exécution de la méthode de détection. Considérant que la méthode développée doit être efficace dans les grands ensembles de données, le nombre d'ANN identifié par *LSH* doit être maintenu aussi bas que possible. L'objectif est donc de trouver la configuration qui allie une grande précision et un nombre d'ANN aussi bas que possible. La performance de chaque configuration testée est présentée à la Figure 4.5.

Les modifications apportées aux configurations comprennent la longueur du *hash*, l'augmentation des données (du fichier requête) et la fonction de similarité. Les résultats optimaux sont obtenus à l'aide de la configuration 5 qui est un excellent compromis entre précision et efficacité de calcul. Dans la configuration 5, le nombre moyen de candidats est faible tandis que la précision de détection est maintenue élevée. La configuration 5 se compose d'une longueur de hachage de 30, sans augmentation des données et *MSSSIM* comme mesure de similarité.

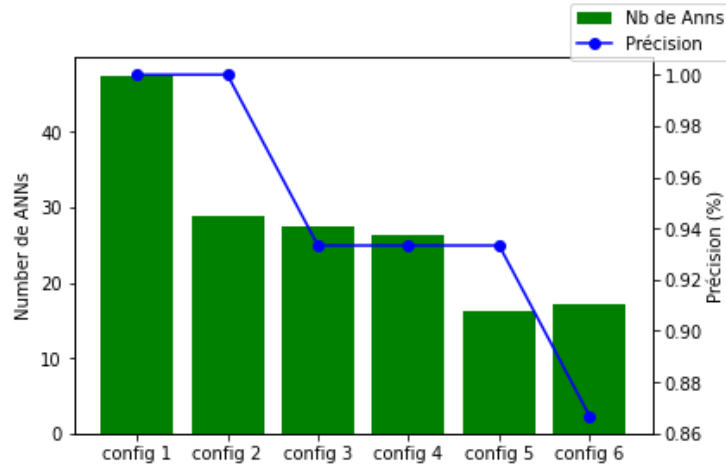


Figure 4.5 : Performance de détection de chacune des configurations testées

4.1.2.3 Performances sur des scans inédits

Dans la troisième expérience, un analyste certifié a créé 30 *ND* qui ont été ajoutés à l'ensemble de données original. Les transformations appliquées aux scans sont le redimensionnement des scans, l'inversion des scans (90 ° vs 270°), la modification des paramètres d'acquisitions et autres transformations similaires.

La technique a détecté, en moins de trois secondes par scan, les 30 *ND* créés intentionnellement parmi l'ensemble de données composé de 400 scans. Aucun algorithme de réduction de dimensionnalité ni de techniques d'échantillonnage n'est utilisé. La technique de détection des *ND* est rapide et suffisamment robuste pour trouver plusieurs types de transformations des données d'origine. Le Tableau 4.2 présente un exemple de détection en utilisant la méthode proposée. Il montre les candidats identifiés par l'algorithme et montre la mesure de similarité associée à chaque doublon potentiel. Selon le seuil, la décision finale est fournie dans la dernière colonne.

Tableau 4.2 : Mesure de similarité des *ANN* pour la soudure #282 270°

Fichier	<i>MSSSIM</i>	Cosine	<i>MAPE</i>	<i>ND</i>
Soudure #280 (270 °)	0,9902	0,9884	0,9251	Oui
Soudure #278 (270 °)	0,9757	0,9627	0,8417	Oui
Soudure #283 (90 °)	0,9555	0,9493	0,6924	Oui
Soudure #281 (90 °)	0,9462	0,9365	0,8185	Oui
Soudure #167	0,8568	0,8134	0,6325	Non

La Figure 4.6 montre les images ultrasonores et les S-scans des fichiers listés dans le Tableau 4.2. On voit clairement que les fichiers identifiés par *LSH* sont effectivement des *ND*, à l'exception du dernier (Soudure #167).

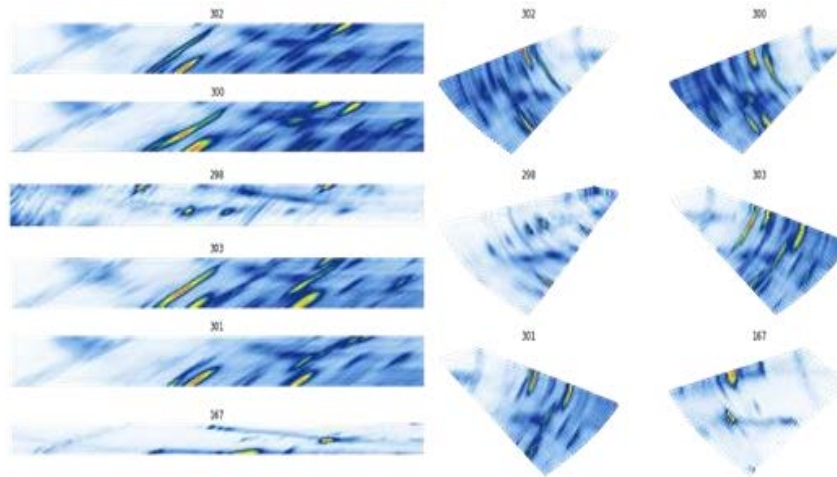


Figure 4.6 : Étude de cas de détection de *ND* d'un fichier test. Sur la gauche, la vue C-scan du fichier testé est en haut et le C-scan des cinq plus proches voisins identifié est en dessous. Sur la droite, les S-Scans résumés des mêmes résultats.

La Figure 4.7 compare deux scans *PAUT* 3D et affiche la similarité entre ces deux scans (S-scan du centre) à l'aide d'une échelle de couleur conçue pour le logiciel infonuagique de Ondia. Il est à noter que les S-scans de gauche et de droite sont des S-scans max où l'amplitude maximale à chaque couple (*beam*, *sample*) est utilisée. L'échelle va comme suit : les parties similaires sont en rouge alors que les points divergeant sont en blanc. Deux scans qui sont des *ED* auraient un S-scan de similarité complètement rouge alors que deux scans complètement différents auraient un S-scan de similarité complètement blanc. Cet outil visuel a été développé afin de rapidement constater la similarité de deux scans, particulièrement pour confirmer visuellement les résultats du processus de détection des *NDs*. Les deux scans comparés sont les scans #282 270° et #280 270° de la troisième expérience. Il convient de rappeler que ces deux scans sont des *ND*, comme montré par les mesures de similarité *MSSSIM*, Cosine et *MAPE*.

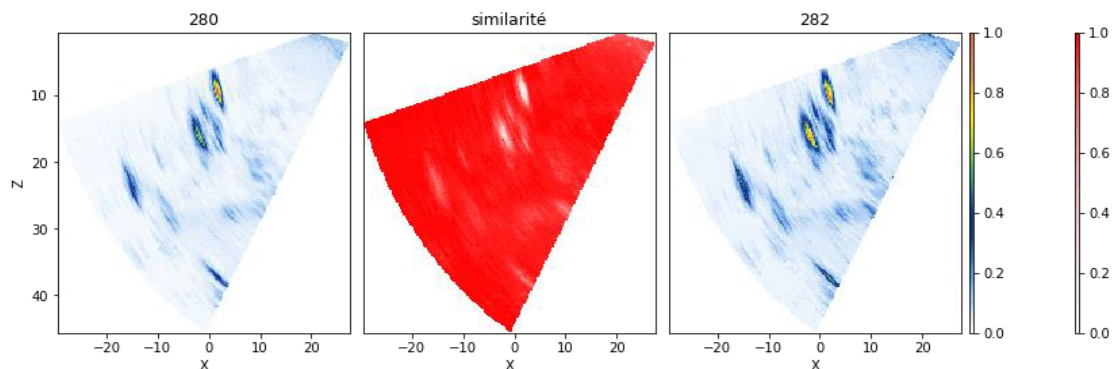


Figure 4.7 : Comparaison de deux S-scans maximales similaires (#280 270° et #282 270°)

4.1.2.4 Robustesse au bruit

La dernière expérience consiste à bruite 15 scans selon un niveau de bruit prédéterminé pour évaluer la robustesse de la méthode. Le bruit correspond à une augmentation du gain suivant $N(\mu, \sigma^2)$ où μ est le niveau de bruit introduit et σ^2 est fixé à 0,01. Le résultat est présenté à la Figure 4.8. On peut observer que la méthode de détection est robuste au bruit puisqu'elle a atteint une grande précision malgré des niveaux de bruit gaussien aussi élevés que 6 %. Lorsque le niveau de bruit devient plus élevé, les *ANN* identifiés par *LSH* pour chaque fichier convergent vers le même petit sous-ensemble de scans très bruités, qui deviennent plus proches du scan modifié que du scan d'origine.

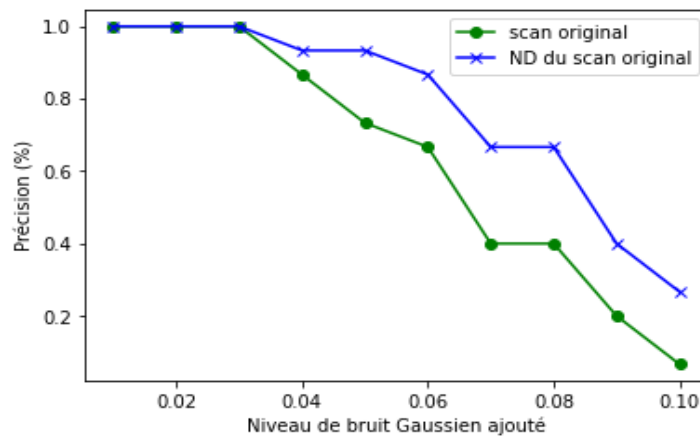


Figure 4.8 : Précision de détection des *ND* à différents niveaux de bruit gaussien.

4.1.2.5 Résumé des quatre expériences

Un résumé des quatre expériences est présenté au Tableau 4.3. Il rassemble les informations présentées dans cette section. Comme on peut le constater, la méthode de détection proposée permet de détecter avec précision les *ND* même lorsque les transformations sont élaborées, telles que le redimensionnement du scan, l'inversement du scan ou les modifications des paramètres d'acquisitions *PAUT*, y compris lorsque du bruit gaussien est ajouté. Il est toutefois important de noter que dans son application pratique, les opérateurs ne passeraient pas plus de quelques minutes à essayer de modifier un scan pour tromper le système.

Bien que les configurations de *LSH* puissent être choisies pour obtenir une précision de détection parfaite, cela entraînerait un processus de détection très lent et inefficace. Par conséquent, cette recherche a testé différentes configurations et situations pour déterminer les paramètres optimaux de *LSH*, permettant une méthode précise et efficace pour détecter les *ND* des scans *PAUT*.

Tableau 4.3 : Résumé des expériences

Expérience	Variables testées	Conclusion
1	Valeur du seuil τ	$\tau = 0,92$
2	Configurations	La configuration 5 est choisie
3	Précision sur de nouveau <i>ND</i> en utilisant la configuration optimale et τ	Résistant aux modifications des paramètres <i>PAUT</i> du scan
4	Robustesse contre le bruit gaussien	Robuste au bruit gaussien jusqu'à 6 %

4.2 Identification de la soudure

Cette sous-section présente tout d'abord le processus d'entraînement du modèle de segmentation. Ensuite, les résultats de la segmentation des quatre parties principales de la soudure sont présentés et sont suivis par les résultats de la segmentation des défauts de soudure.

4.2.1 Entraînement du modèle de segmentation

Le modèle de segmentation multi-objectif a été entraîné à segmenter les quatre parties principales des soudures et les défauts à l'aide de 106 scans ultrasonores annotés par deux experts en ultrasons de niveau III. Les techniques d'augmentation des données présentées à la section 3.3.3.2 ont été utilisées. L'entraînement est illustré à la Figure 4.9.

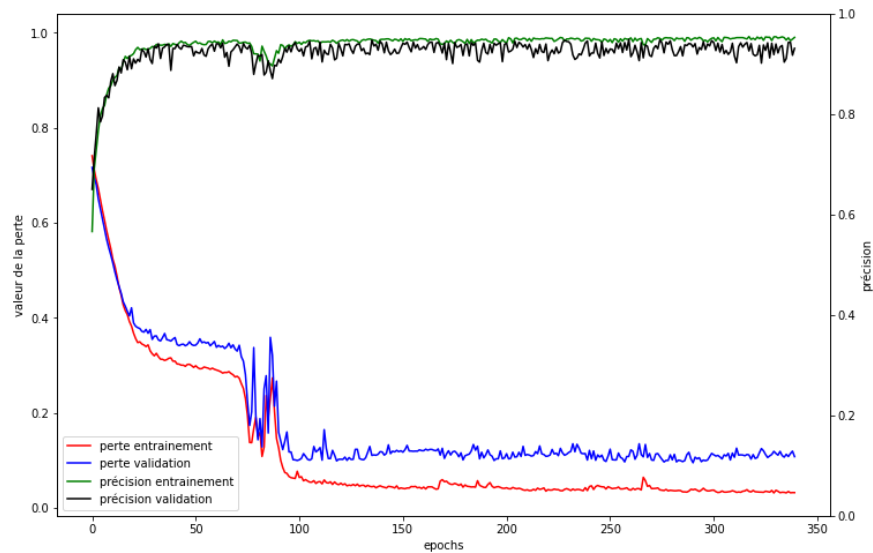


Figure 4.9 : Évolution de la valeur de la perte et de la précision au fil des *epochs*

On peut voir que le modèle atteint une précision maximale de 97,6 % à l'*epoch* 259 et qu'il atteint un *DSC* maximal de 90,47 % ($1-0.0953$) à l'*epoch* 290. L'entraînement a été en trois phases. La première sans planificateur de *LR*. *CyclicLR* a été ajouté à l'entraînement à partir de l'*epoch* 169 pour

76 *epochs*. Ensuite, *ReduceLRonPlateau* a été utilisé pour terminer l'entraînement. *Early stopping* a été utilisé dans les trois phases de cet entraînement (sans planificateur de *LR*, *CyclicLR*, *ReduceLRonPlateau*). L'entraînement a nécessité un total de 340 *epochs* pour une durée de 283 minutes (4 heures et 43 minutes) sur un ordinateur équipé d'un *GPU* Nvidia GeForce RTX 2080.

4.2.2 Segmentation des quatre principales parties de la soudure

4.2.2.1 Analyse des résultats

Le modèle de segmentation multi-objectif atteint une précision totale de 97,7 % au niveau des voxels et un *DSC* de 90,47 % sur l'ensemble de données de test. En détail, le modèle atteint une précision de 95,6 % et un *DSC* de 90,55 % pour la segmentation des parties de la soudure. La Figure 4.10 montre une matrice de confusion qui montre la qualité de la classification de chaque voxel en fonction du *GT*.

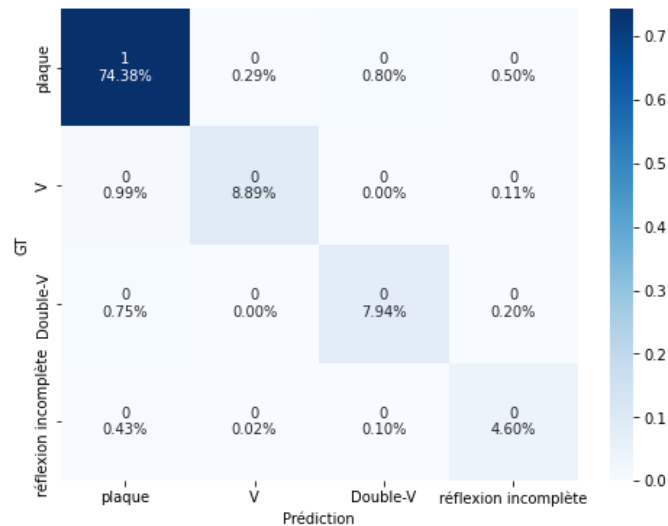


Figure 4.10 : Matrice de confusion de la segmentation de chaque voxel de la soudure

On peut constater qu'environ 75 % des voxels correspondent au matériel de base. Le modèle a réussi à bien classer 74,38 % des 74,67 % des voxels appartenant au matériel de base. En revanche, le modèle a classé 0,99 % des voxels comme appartenant à une soudure en V alors qu'ils appartiennent au matériel de base. Il s'agit de la plus importante erreur du modèle. La deuxième plus importante erreur est la classification de 0,80 % des voxels comme étant une soudure en double-V alors qu'il s'agit du matériel de base. Une troisième erreur similaire de classification est présente lorsque le modèle identifie 0,75 % des voxels comme étant le matériel de base alors qu'il s'agit de soudure en double-V. Il est intéressant de souligner que le modèle réussit très bien à distinguer les types de soudure. L'erreur de classification entre les classes soudure en V et double-V est nulle. La plupart

des erreurs sont liées au matériel de base (3,76 % des 4,20 % d'erreur total). Cette part d'erreur s'explique en grande partie par la difficulté de catégoriser chacun des voxels dans la ZAT. Les voxels dans la ZAT peuvent pratiquement être catégorisés comme appartenant autant en matériel de base que faisant partie de la soudure considérant la frontière floue entre ces zones. Puisque le matériel de base est majoritaire dans un scan, il est aussi possible que le modèle favorise cette classe par rapport aux autres. La section 4.2.2.2 est entièrement dédiée à la présence d'une ZAT et ses effets sur l'algorithme de segmentation.

Le modèle obtient un score F1 moyen de 91,7 %, grâce à une précision moyenne de 91,4 % et un rappel moyen de 92,2 %. Le score F1 de chaque classe est disponible dans le Tableau 4.4. Le score F1 moyen et pondéré en fonction de l'importance de chaque classe est également disponible. L'importance de chaque classe est la suivante : 76,2 %, 10,5 %, 8,2 % et 5,1 %.

Tableau 4.4 : Tableau des scores F1 pour chaque classe de la soudure

Classe	Précision (%)	Rappel (%)	Score F1 (%)
Matériel de base	97,9	97, 2	97,5
Soudure en V	89,0	96,6	92,7
Soudure en double-V	89,3	89,8	89,6
Réflexion incomplète	89,3	85,0	87,1
Moyenne	91,4	92,2	91,7
Pondéré	95,8	95,8	95,8

Il est intéressant de souligner la différence de score F1 entre la version pondérée et la version basée sur la moyenne. La version pondérée prend davantage en considération l'importante représentation du matériel de base sur l'ensemble des données alors que la version moyenne considère chaque classe également. Le score F1 moyen est préféré à la version pondérée, car la version moyenne est plus critique envers les classes associées aux soudures et c'est majoritairement le résultat de la segmentation de ces deux classes qui sont utilisées pour la suite du processus d'analyse développé dans cette étude.

Un score F1 moyen de 91,7 % est un bon indicateur pour démontrer que le modèle est en mesure de bien remplir son objectif de segmenter les différentes parties de la soudure. Le score F1 le plus faible appartient à la classe réflexion incomplète qui a obtenu un score de 87,1 %. Bien que ce soit le score le plus faible, cette valeur n'est pas critique. Il s'agit d'une classe difficile à segmenter puisque la forme de la réflexion incomplète peut grandement varier d'une soudure à l'autre, particulièrement entre une soudure en V et une en double-V. La forme peut également varier en fonction des différents paramètres de l'appareil d'acquisition. Il est également possible de constater que le modèle possède un meilleur score F1 pour les soudures en V (92,7 %) en comparaison aux soudures en double-V (89,6 %). Cette différence s'explique majoritairement par un rappel de seulement 89,8 % pour les

soudures en double-V alors qu'il est de 96,6 % pour la soudure en V. Cette différence est causée par une l'erreur de classification de la soudure en double-V quand il s'agit du matériel de base (0,80 %). Ce type d'erreur n'est que de 0,20 % pour la soudure en V. Cette erreur s'explique en partie par la plus forte complexité de la forme de la soudure en double-V par rapport à une soudure en V. Le faible rappel de la classe réflexion incomplète (85,0 %) est causé en grande partie par une erreur de classification importante au niveau du matériel de base. L'erreur atteint 0,50 % alors que seulement 5,15 % des voxels appartiennent à cette classe.

À la Figure 4.11, un exemple du processus de segmentation de la soudure est représenté pour deux scans ultrasonores faisant partie de l'échantillon de validation. Pour simplifier la visualisation, les scans 3D sont représentés par des S-Scans 2D. Il faut noter que le modèle utilise bien des scans 3D *PAUT* en entrée et non des images ultrasonores. Les S-Scans de la colonne de gauche sont les données d'entrée transmises au *FCN* après redimensionnement. La colonne du centre représente les cartes de segmentation prédites par le réseau. Les S-scans de la colonne de droite représentent les cartes de segmentation identifiées par l'analyste certifié Ondia. La première ligne est une soudure en V alors que la deuxième ligne est une soudure en double-V. La troisième ligne correspond à un exemple difficile où la soudure est décalée de près de 20 mm vers la droite par rapport au centre du centre, où les soudures sont généralement positionnées. Les couleurs des colonnes du centre et de droite correspondent à : bleu/vert foncé=matériel de base, vert=soudure en V, jaune=soudure en double-V et rose=réflexion incomplète de la soudure.

Pour ces trois prédictions, le modèle atteint un *DSC* de 94,8 %, de 94,7 % et de 90,3 %. La Figure 4.11 démontre clairement que le réseau de segmentation reconnaît la forme de la soudure et peut distinguer des parties importantes de la soudure (matériel de base, soudure en V, soudure en double-V). Le modèle est en mesure de segmenter la soudure de façon précise (*DSC* de 90,3 %) alors que la soudure est située près de 20 mm vers la droite de la position habituelle des soudures. Cela montre bien que le *FCN* a bien appris à localiser les soudures. Il s'agit d'un indicateur visuel de l'absence de surapprentissage.

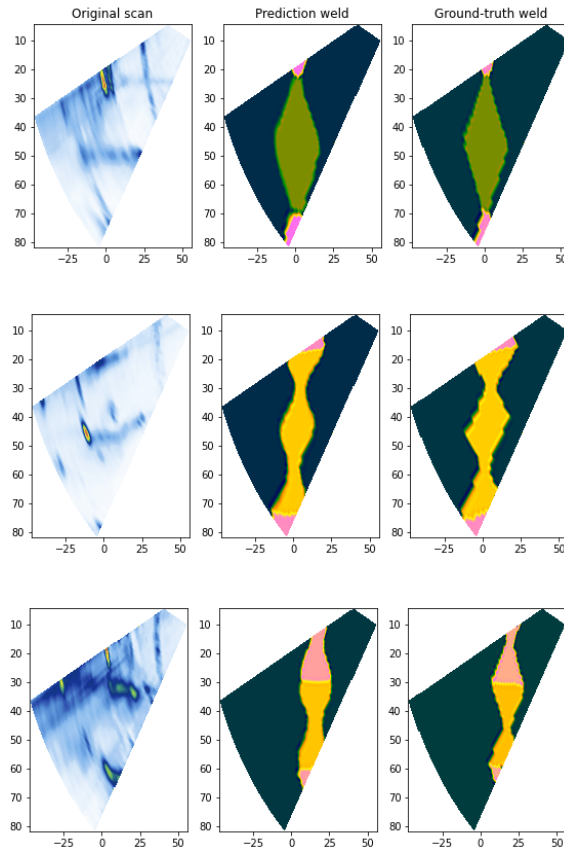


Figure 4.11 : Exemples des intrants, prédictions et *GT* du modèle de segmentation.

La Figure 4.12 montre la distribution du *DSC* de l'échantillon de validation pour les soudures.

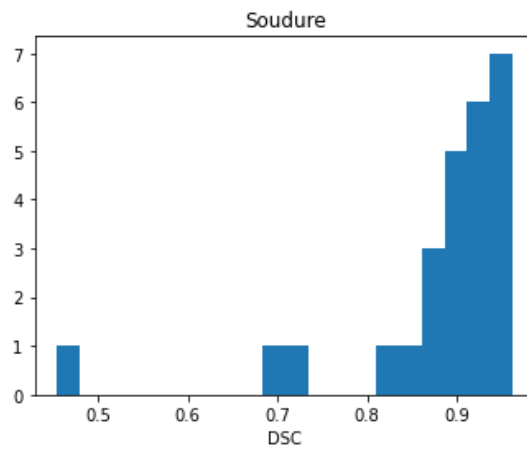


Figure 4.12 : Distribution des *DSC* de l'échantillon de validation pour la soudure

On peut voir que 24 des 26 prédictions ont un *DSC* supérieur à 80 %. Les deux pires prédictions ont un *DSC* de 45,5 % et de 73,3 %. La pire prédiction est présentée à la Figure 4.13. On peut y observer qu'il s'agit d'une soudure en V de forme particulière qui est très peu représentée dans la base de données d'entraînement (ayant peu de S-scans similaires). Il s'agit de la raison principale pour laquelle la prédiction est aussi ratée. À noter, la prédiction du type de soudure est fausse (V vs Double-V). Bien que deux prédictions soient insatisfaisantes, les résultats sont intéressants, car l'ajout de données d'entraînement et l'ajout de variabilité dans la base de données devraient permettre de réduire les mauvaises prédictions.

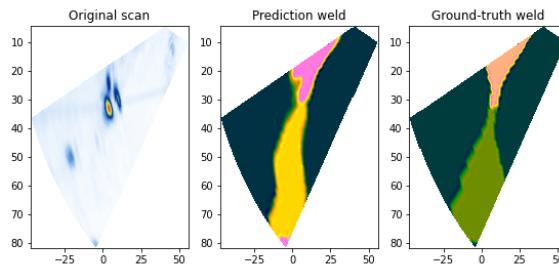


Figure 4.13 : Pire prédiction de l'échantillon de validation

Pour évaluer davantage la capacité du modèle à reconnaître la forme des soudures, une légère modification du *FCN* pour générer une prédiction binaire de la forme de la soudure a été effectuée. Le modèle de classification des soudures a été entraîné à l'aide de 281 scans *PAUT* 3D en entrée. Il a été testé sur 70 scans, atteignant une précision de classification de 92 %. Cela met en évidence la capacité du modèle à distinguer le type de soudure dans un scan 3D *PAUT*.

4.2.2.2 Effets de la ZAT

La frontière entre la soudure et le matériel de base est une zone de transition qui se nomme la ZAT. La ZAT est la zone adjacente à la soudure qui a été chauffée suffisamment pour avoir subi un changement dans sa microstructure (dû à son exposition à de très hautes températures), mais pas suffisamment pour avoir fondue [90]. La chaleur créée lors du soudage et le refroidissement subséquent causent un changement entre l'interface de la soudure et la fin de la zone où la température a augmenté dans le matériel de base. La composition chimique dans la ZAT est la même que pour le matériel de base, mais les propriétés et la structure ont changé par le traitement thermique. Ces changements peuvent être significatifs et peuvent causer une défaillance sous la forme de craques, de corrosion, de distorsion ou autres. Les effets sur les propriétés mécaniques dans la ZAT sont généralement négatifs [9]. L'étendue et l'ampleur du changement de propriétés dépendent du matériau de base, du métal d'apport, la distance de la zone de fusion, la durée du traitement thermique, le processus de refroidissement et la quantité et la concentration de la chaleur

apportées par le processus de soudage. La diffusion thermique est l'un des facteurs qui jouent un grand rôle dans la création de la ZAT. Il s'agit de la mesure qui qualifie la capacité du matériel de base à transmettre la chaleur. Le calcul correspond au rapport entre la conductivité thermique du métal divisée par sa densité et sa capacité thermique à pression constante. Si la capacité de diffusion thermique est importante, le matériel de base est en mesure de transmettre la chaleur plus rapidement, ce qui se traduit par une ZAT plus petite. Le matériel de base et le cycle thermique (chauffage et refroidissement dus au soudage) ainsi que tout traitement thermique post-soudage sont les facteurs déterminants du changement dans la ZAT. Identifier la ZAT est donc une tâche importante. C'est en général dans la ZAT que les défaillances de la soudure occurrent. Ainsi, un grand nombre de défauts s'y retrouve, notamment des craques.

Pour démontrer l'importance de localiser la ZAT, un exemple est discuté en considérant l'acier au carbone. Lors d'un refroidissement rapide de l'acier au carbone, il se forme de la martensite. La martensite est une solution dure et très cassante du carbone solide dans le fer. La martensite est souhaitable dans certains cas, car elle est extrêmement dure, mais peut en même temps présenter une faible ductilité. Elle est donc idéale pour les outils, mais pas pour l'acier de construction. Le danger d'avoir une ZAT fragile dans l'acier au carbone est la susceptibilité à la fissuration induite par l'hydrogène. Cette microstructure fragile cède sous la pression exercée par l'hydrogène qui se diffuse hors de la soudure et de la ZAT. Ce phénomène est dangereux, car, contrairement à la fissuration à chaud qui se produit juste après le soudage, la plupart des fissures à froid (comme dans le cas de la fissuration assistée par l'hydrogène) ne se produisent que plusieurs heures après le refroidissement de la pièce. C'est pourquoi les procédures de soudage appropriées prévoient l'inspection des soudures susceptibles de subir une fissuration à froid 48 heures après la fin du soudage. La ZAT étant sensible à tous ces problèmes, le fait de la localiser géométriquement aide les analystes dans leur diagnostic.

Les figures présentées précédemment utilisent la classe ayant le plus fort niveau de confiance pour chaque voxel pour créer un masque 3D. Or, il est possible de visualiser la prédiction brute pour chaque classe afin de voir le niveau de confiance du modèle à chaque voxel. La Figure 4.14 montre la valeur moyenne de la prédiction pour chaque point en XZ. En d'autres termes, il s'agit de la visualisation de la probabilité d'appartenance pour chaque classe. Il est possible de constater pour les deux exemples de la Figure 4.14 que le modèle est très confiant en ce qui a trait à la classe 1 (matériel de base). Dans les trois autres classes, il semble y avoir une zone d'incertitude entre la soudure, que ce soit en V, en double-V ou la réflexion incomplète, et le matériel de base. Cette zone d'incertitude peut atteindre plusieurs millimètres. Nous pouvons émettre l'hypothèse qu'il s'agit, du moins en partie, de la ZAT. La plupart des erreurs du modèle, discutées précédemment, proviennent de cette zone. Il est possible de constater qu'il existe une zone de transition entre la soudure et le matériel de base où la confiance des prédictions passe de forte à faible, de la soudure à la ZAT, et

redevient plus forte en atteignant le matériel de base. Ce phénomène est observé pour les deux prédictions (soudure en V ou double-V).

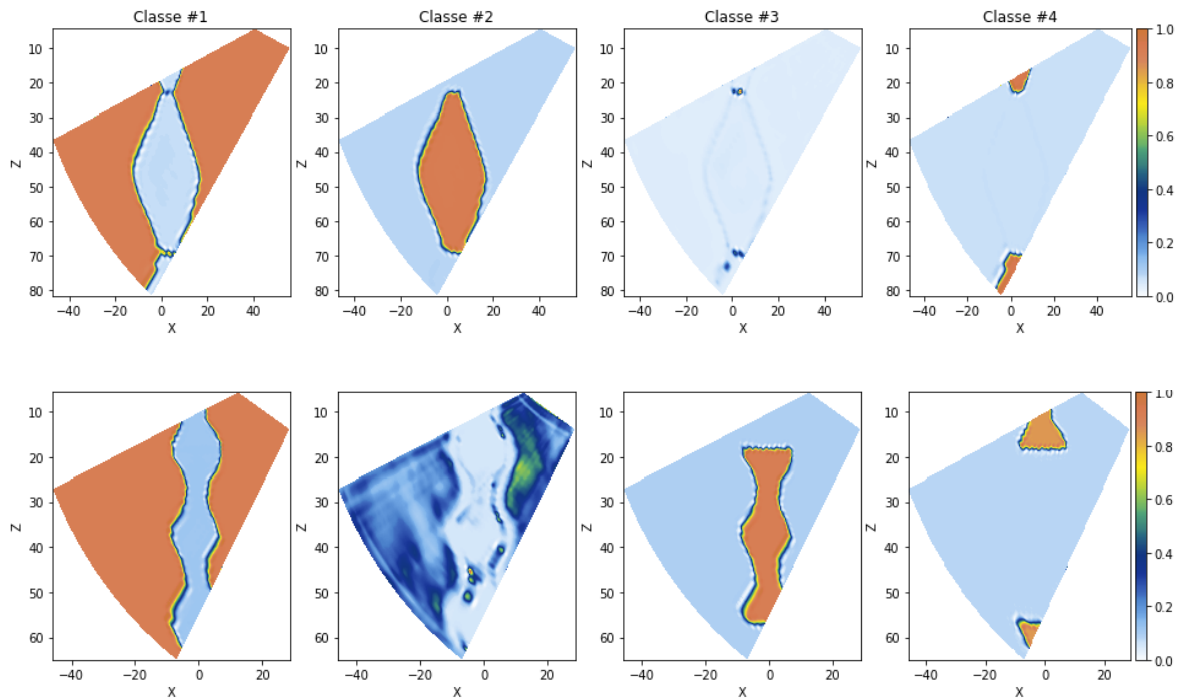


Figure 4.14 : Visualisation de chaque classe de la soudure pour deux prédictions

À la Figure 4.15, on peut voir deux prédictions du modèle de segmentation pour la classe soudure en V (à gauche) et soudure en double-V (à droite). Afin de visualiser la ZAT, un contour de 8 mm (en rouge) a été ajouté à partir de la soudure. À partir de cette représentation, on peut alors avoir une idée de la forme, grosseur et localisation de la ZAT dans le scan fourni pour analyse. Puisque les défauts de soudure sont également segmentés, il est alors possible de localiser les défauts par rapport à la ZAT. Cela devrait se traduire par une augmentation de la qualité des analyses effectuées par les analystes certifiés. Par exemple, tous les défauts segmentés se retrouvant dans la ZAT devraient être analysés précisément, car ils sont à risque plus élevé. Il faut préciser qu'il est facilement possible de changer la taille de la ZAT théorique en fonction des conditions de la soudure (matériel, type de soudure, etc.).

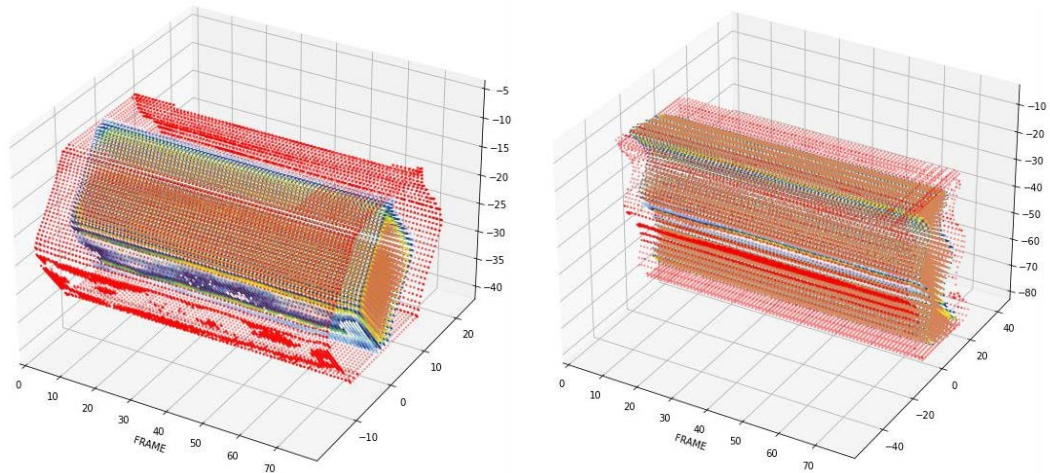


Figure 4.15 : Illustration en 3D de la ZAT potentielle de deux prédictions (Soudure en V à gauche et double-V à droite)

4.2.3 Segmentation des défauts

Le modèle atteint une précision de 99,8 % et un *DSC* de 79,09 % pour la segmentation des défauts. La segmentation des défauts est une tâche difficile pour le modèle de segmentation multi-objectif puisque 99,72 % des voxels ne contiennent pas de défauts. Le modèle réussit donc à bien classer 99,79 % des voxels, ce qui est légèrement supérieur au pourcentage de voxels n'ayant pas de défaut. L'erreur totale est de 0,21 % seulement. Or, en considérant seulement les voxels qui n'ont pas de défaut, le modèle réussit à bien classer 0,17 % des 0,28 % de voxels avec défaut, ce qui représente 61 %. En somme, il est possible de considérer que le modèle de segmentation des défauts est suffisamment performant puisqu'il ne produit que très peu de faux positifs. Les positifs correspondent à l'identification positive (défaut) d'un voxel alors que le vrai résultat (*GT*) est négatif (sans défaut). Les faux négatifs, qui correspondent à l'identification négative d'un voxel (sans défaut) alors qu'il y a un défaut, sont un problème encore plus important en END. Les faux négatifs représentent 0,11 % des prédictions. Ce pourcentage est faible et donc acceptable bien qu'il soit non négligeable considérant la rareté des défauts (0,28 % seulement des voxels.) Une réduction du taux de faux négatifs serait une amélioration intéressante pour une application dans un logiciel infonuagique. La matrice de confusion est présentée à la Figure 4.16.

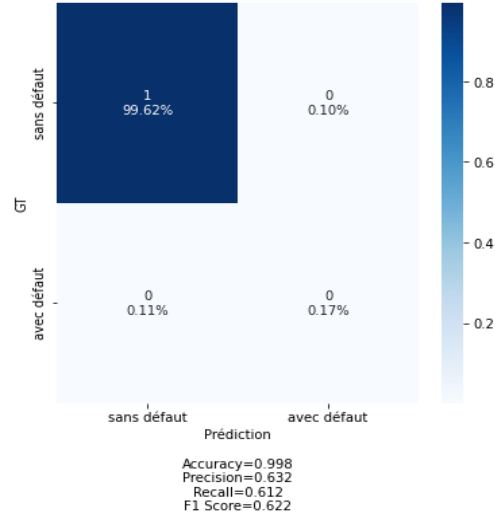


Figure 4.16 : Matrice de confusion de la segmentation de chaque voxel des défauts

Le modèle obtient un score F1 de 62,2 %, grâce à une précision de 63,2 % et un rappel de 61,2 %. Cela correspond à une précision de :

$$Précision = \frac{VP}{VP + FP} = \frac{0,17}{0,17 + 0,10} = 63,2 \% \quad (4.1)$$

Le rappel équivaut à :

$$Rappel = \frac{VP}{VP + FN} = \frac{0,17}{0,17 + 0,11} = 61,2 \% \quad (4.2)$$

Finalement, le score F1 est calculé ainsi :

$$Score\ F1 = \frac{2 * précision * rappel}{précision + rappel} = \frac{2 * 0,632 * 0,612}{0,632 + 0,612} = 62,2 \% \quad (4.3)$$

Un score F1 de 62,2 % est dû au fait que cette mesure de performance ne considère pas les vrais négatifs dans son évaluation. Puisque 99,62 % des voxels ont été classés comme des vrais négatifs, cette mesure n'est basée que sur les 0,38 % restants. Il est toutefois intéressant de noter que le modèle possède une certaine marge d'amélioration au niveau de la segmentation des défauts. Cette marge d'amélioration peut être expliquée par le fait que la fonction de perte est fortement pondérée en faveur de la segmentation de la soudure au cours de l'entraînement. La pondération de la valeur de perte est une décision prise afin de mettre l'accent sur la segmentation de la soudure, qui est au cœur de l'objectif général de cette recherche. Afin d'augmenter les performances de la segmentation des défauts, l'ajout de données ou la modification de la pondération, pour réduire l'accent sur la segmentation de la soudure, devrait permettre une amélioration des indicateurs de performance.

Il faut rappeler que le modèle a été entraîné avec seulement 106 scans *PAUT* alors que les modèles de segmentation 3D sont généralement entraînés avec des milliers de données. Par exemple, la base de données Semantickitti, utilisée pour la segmentation 3D, consiste en plus de 40 000 annotations [217].

À la Figure 4.17, un exemple du processus de segmentation est représenté pour deux scans faisant partie de l'échantillon de validation. Les trois premiers scans de gauche représentent un S-scan original, la prédiction et le *GT*. Les trois S-scans de droite sont un autre exemple, suivant le même ordre.

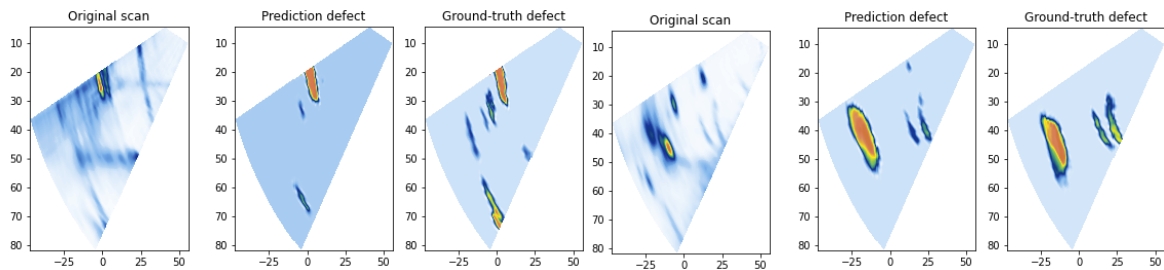


Figure 4.17 : Exemples des intrants, prédictions et annotations du modèle de segmentation

Pour ces deux prédictions, le modèle atteint un *DSC* de 88,3 % et 91,7 %. Il est à noter que l'échelle de couleur débute par le blanc pour les faibles amplitudes, passe ensuite par le blanc et termine dans l'orange pour les fortes amplitudes. Il s'agit de l'échelle de couleur standard en ultrason.

La Figure 4.18 montre la distribution du *DSC* de l'échantillon de validation pour les défauts. On peut remarquer que les prédictions ont toutes un *DSC* supérieur à 88 %, ce qui est intéressant. Il ne semble pas y avoir de mauvaises prédictions.

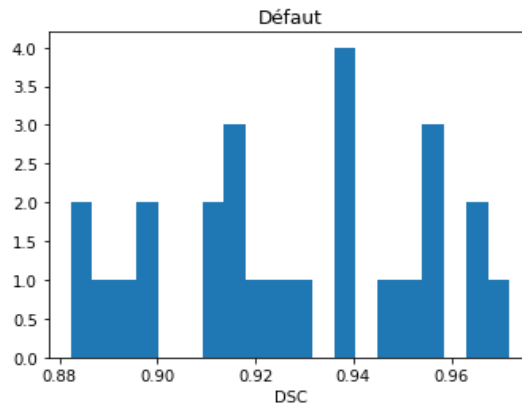


Figure 4.18 : Distribution des *DSC* de l'échantillon de validation pour les défauts

La Figure 4.19 montre la valeur moyenne de la prédiction pour chaque point en XZ. Pour les deux exemples affichés, on peut voir un S-scan moyen de chaque classe de la prédiction des défauts. Le S-scan de gauche (Classe #1) de chaque exemple correspond à la classe sans défaut alors que celui de droite est la classe des défauts (Classe #2). L'échelle de couleur est en lien avec le degré de confiance de la prédiction du modèle. En orange, la prédiction est fiable alors que plus la couleur tend vers le blanc, en passant par le jaune puis le bleu, moins la confiance envers cette prédiction est élevée. En observant attentivement les deux S-scans de la Classe #2, on constate qu'il semble y avoir une légère zone d'incertitude encerclant les défauts (zone de couleur bleu). Cette zone semble toutefois être plus petite que pour les parties de la soudure.

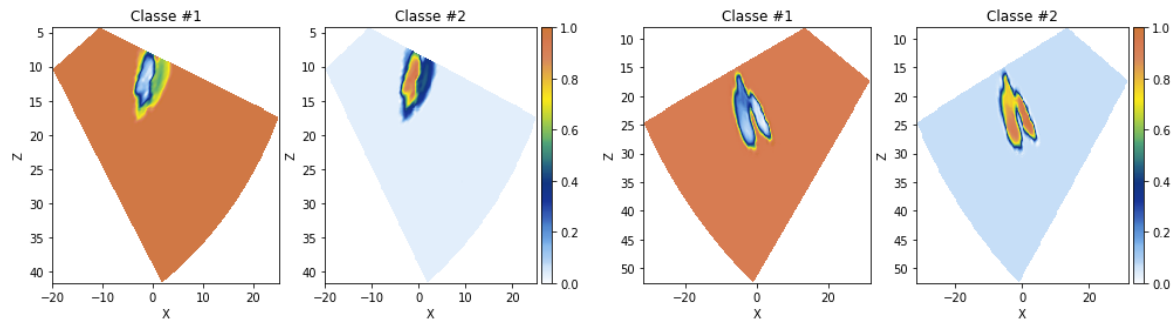


Figure 4.19 : Visualisation pour deux prédictions des classes sans défaut et défaut

4.2.4 Segmentation intégrée de la soudure et des défauts

La méthode développée permet une analyse approfondie de la forme de la soudure. La Figure 4.20 représente visuellement le processus de segmentation de la soudure. Ce processus débute avec un scan 3D (NP de gauche), le segmente à l'aide du modèle de segmentation multi-objectif, sélectionne les points associés à la soudure et transforme finalement ces points en NP pour faire l'affichage 3D de la soudure (NP du centre). Le NP de droite est le *GT* de la soudure. Dans le processus global de reconstruction de la surface, le NP obtenu est ensuite transmis à *P2S* pour reconstruire la surface et ainsi créer le modèle géométrique 3D de la soudure et des défauts.

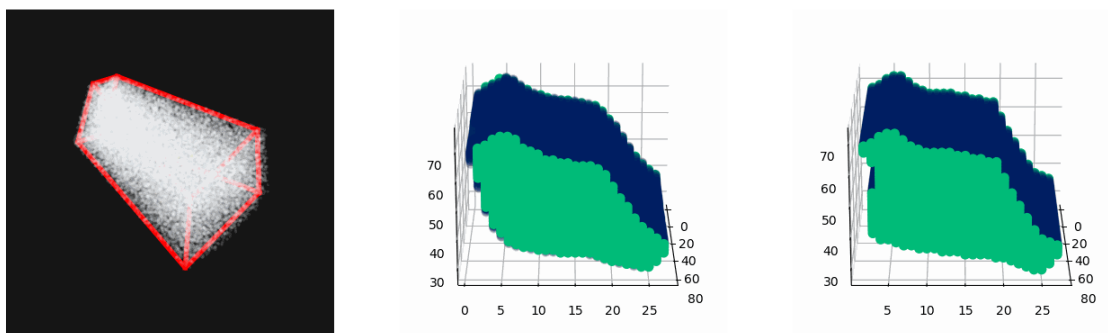


Figure 4.20 : Exemple 3D des intrants, prédictions et extrants du modèle de segmentation : entrée (à gauche), extrant prédit (au milieu) et forme réelle de la soudure en V (à droite)

L'analyse des soudures peut être considérablement améliorée en fournissant de telles informations géométriques. Par exemple, on peut estimer la longueur réelle de la soudure. Plus important encore, l'interprétation des informations géométriques pourrait également aider à déterminer si l'analyse elle-même a été effectuée correctement. Le même processus peut être utilisé pour générer un NP des défauts. La Figure 4.21 illustre le processus pour des défauts de soudure : à gauche : un scan 3D intrant, au centre : l'affichage 3D du défaut par un NP et à droite : NP du *GT* des défauts.

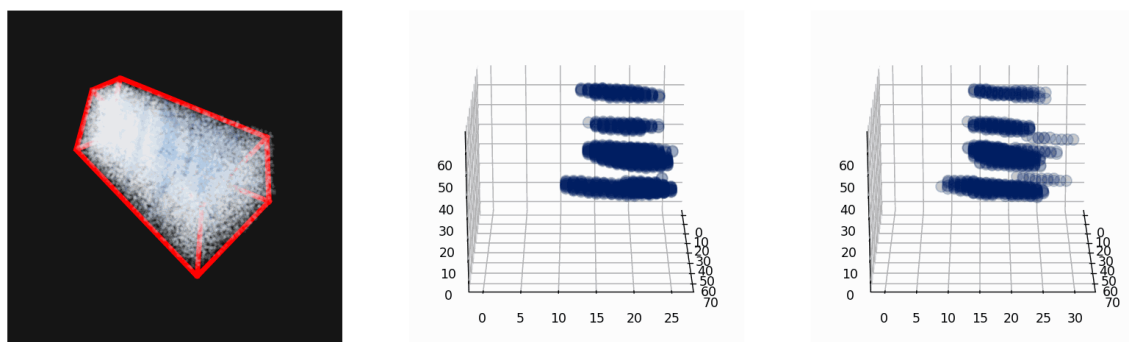


Figure 4.21 : Exemple 3D des intrants, prédictions et extrants du modèle de segmentation : entrée (à gauche), extrant prédit (au milieu) et forme réelle des défauts (à droite)

La création d'un NP en 3D de la soudure et des défauts permet de mettre en contexte chacun des éléments par rapport à l'autre. En effet, il est ainsi possible de connaître la forme, la grosseur et la localisation relative des défauts et d'en déduire la gravité. Le diagnostic de l'analyste ne peut qu'être meilleur s'il possède ces informations.

À la Figure 4.22, on peut voir trois exemples de prédictions du modèle de segmentation où les deux cartes de segmentation (soudure et défaut) produites ont été superposées. Cette superposition des prédictions permet de localiser géométriquement les défauts par rapport à la soudure. Les deux

premières lignes sont des soudures en double-V, identifiées en jaune, et la dernière ligne est une soudure en V, identifiée en bleu.

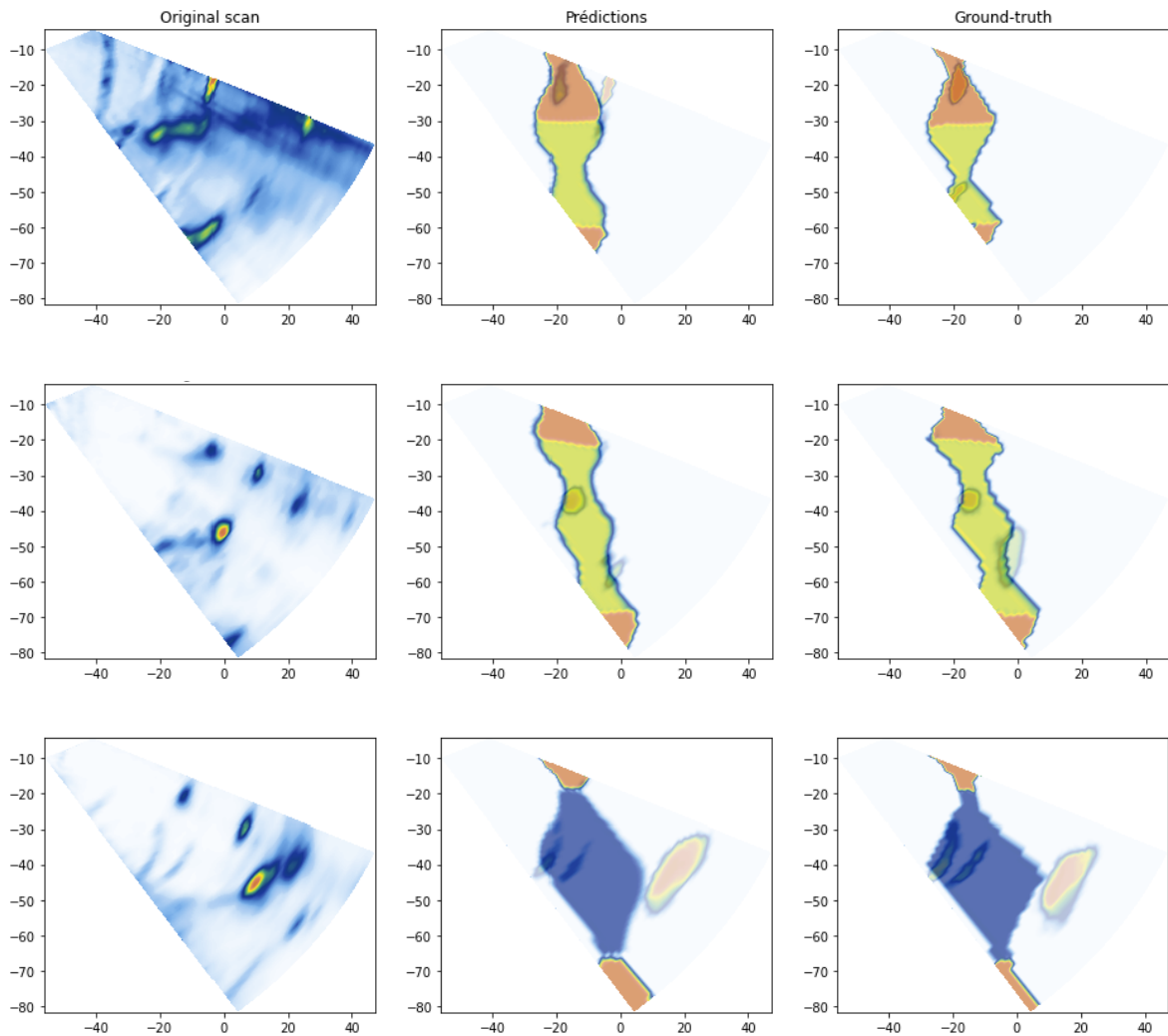


Figure 4.22 : Exemples des intrants, prédictions et annotations du modèle de segmentation superposés

Il est possible de constater que les prédictions, autant de la soudure que des défauts, sont très proches du *GT*. Le principal attrait de ces graphiques est de localiser les défauts par rapport aux différentes parties de la soudure. Par exemple, dans le deuxième exemple, on peut constater qu'un défaut chevauche la soudure et la ZAT, aux environs de $x = 0$ mm et $z = -60$ mm. On peut également constater que les autres défauts identifiés dans cet exemple sont à l'intérieur de la soudure ($x = -15$ mm et $z = -37$ mm).

En somme, la représentation de la soudure et des défauts de la Figure 4.22 permet aux analystes de confirmer leur diagnostic grâce à la connaissance précise de la position, de la forme et du type de soudure ainsi que de la ZAT.

4.2.5 Comparaison avec un auto-encodeur convolutif

La rareté des défauts dans les soudures est un problème bien documenté [38]. Un modèle de segmentation basée sur l'apprentissage non supervisé, comme un auto-encodeur convolutif, pourrait également segmenter les défauts de soudure. Ce type de modèle est entraîné à partir de scans sans défauts pour apprendre à reconstruire les scans normaux (sans défaut). À l'inférence, le modèle est en mesure de reconstruire les voxels sans défaut, permettant l'identification des voxels avec défaut grâce à l'erreur de reconstruction.

Considérant les avantages de l'apprentissage non supervisé, un auto-encodeur convolutif a été testé. Les performances de ce modèle ont été comparées au modèle de segmentation supervisé. L'auto-encodeur convolutif a été entraîné à partir des mêmes 106 scans ultrasonores. Les *frames* ayant des défauts ont été retirés. La fonction de perte *SSIM* a été utilisée. *SSIM* a atteint un sommet de 85,5 % de ressemblance entre les prédictions et les scans intrants. Au total, l'entraînement a duré 877 *epochs*. Les résultats sont résumés au Tableau 4.5.

Tableau 4.5 : Comparaison quantitative des résultats de segmentation entre un modèle à apprentissage supervisé et non supervisé

Types d'apprentissage	DSC (%)	Précision (%)	F1-Score (%)
Supervisé (FCN)	79,09	99,79	62,2
Non supervisé (auto-encodeur convolutif)	36,08	99,58	38,1

Afin de comparer les résultats des deux modèles, le *DSC*, la précision et le score F1 ont été calculés à partir des annotations des défauts. L'auto-encodeur convolutif a atteint un *DSC* maximal de 36,08 %, une précision maximale de 99,58 % et un score F1 de 38,1 %. Sachant que 99,72 % des voxels ne contiennent pas de défauts, une précision de 99,58 % est décevante. L'auto-encodeur convolutif ne réussit donc pas à mieux classer les voxels qu'une simple prédiction négative à chaque voxel. Un *DSC* de seulement 36,08 % est également décevant puisque cela montre que le modèle ne réussit pas à bien segmenter les scans. La Figure 4.23 montre la matrice de confusion de la segmentation des défauts par l'auto-encodeur convolutif. On peut y constater que la précision, le rappel et le score F1 sont respectivement de 31,3 %, 48,7 % et de 38,1 %, ce qui largement inférieur aux résultats du modèle de segmentation supervisé (63,2 %, 61,2 % et 62,2 %). On peut constater que l'auto-encodeur convolutif classe 0,28 % des voxels comme ayant un défaut alors qu'ils n'en ont

pas (faux positif) et qu'il classe 0,14 % des voxels comme n'ayant pas de défaut alors qu'ils en ont (faux négatif).

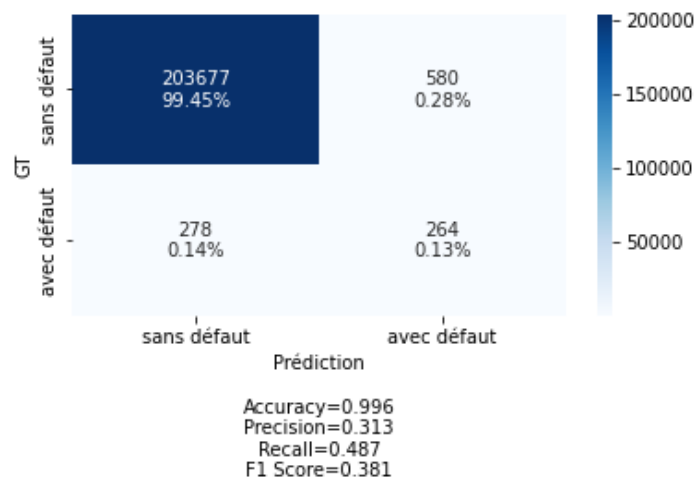
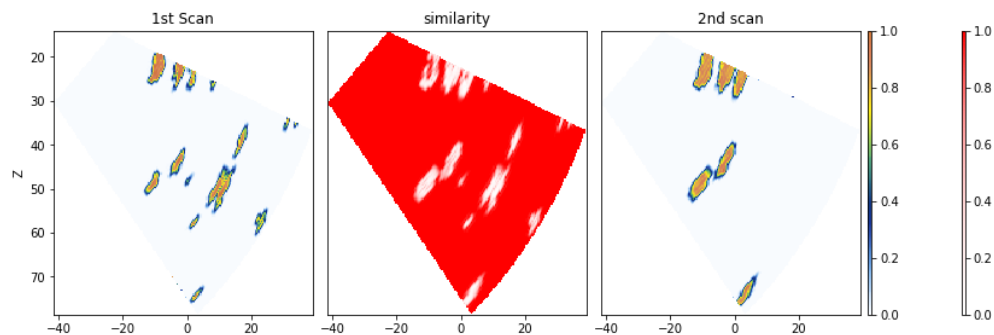


Figure 4.23 : Matrice de confusion de la segmentation par un auto-encodeur convolutif

Visuellement, il est possible de constater à la Figure 4.24 que l'auto-encodeur convolutif laisse de nombreux artefacts dans la prédiction, contrairement au modèle de segmentation supervisé. Les scans de gauche sont les prédictions à partir de l'auto-encodeur convolutif alors que les scans de droite sont les scans annotés des défauts (le GT). Au centre, il s'agit de S-scans représentant la similarité entre les deux autres scans. Le rouge correspond à une similarité élevée alors que le blanc pour représente les différences.



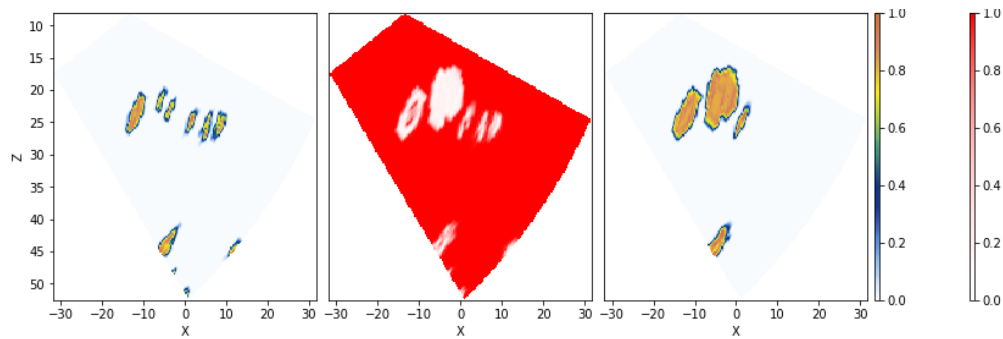


Figure 4.24 : Comparaison des résultats de segmentation d'un auto-encodeur convolutif aux annotations

Autant les résultats numériques que les résultats visuels présentés ci-haut justifient l'utilisation du modèle de segmentation supervisé. Un modèle de segmentation non supervisée possède plusieurs avantages, comme l'absence de besoin de données annotées complexes ou la capacité de segmenter des défauts rares, avec une architecture plus simple. Mais dans ce cas-ci, les performances de ce type de modèle ne sont pas suffisantes pour justifier son utilisation. Il est toutefois possible d'envisager son utilisation dans certaines situations ciblées. Par exemple, il peut être utile de combiner les deux modèles pour s'assurer de détecter les défauts rares dans le cas où les faux positifs ne sont pas problématiques. Un autre exemple est l'utilisation de l'auto-encodeur convolutif lorsque de nouvelles données qui proviennent de distribution différente sont disponibles, comme des scans au format différent (plus de *frames* ou de *samples*), des scans provenant de matériaux différents ou d'appareils différents.

4.3 Modélisation géométrique en trois dimensions

Cette section présente tout d'abord l'optimisation des hyperparamètres spécifiques à *P2S* afin de faire une comparaison juste avec *SPR*. L'entraînement à partir des hyperparamètres optimaux est présenté. Ensuite, les résultats de la reconstruction de la surface et la comparaison des deux méthodes testées sont discutés. S'en suit le choix final de la technique de reconstruction de surface.

4.3.1 Résultats de *Hyperband* appliqués à *P2S*

Le Tableau 4.6 résume les choix d'hyperparamètres possibles et indique la combinaison ayant obtenu les meilleures performances. *Hyperband* a été exécuté en utilisant $R=64$, ce qui correspond au nombre maximal de ressources autorisées, et $\eta=4$, le nombre de branches considéré. La meilleure combinaison d'hyperparamètres a obtenu une valeur de perte de 0,008. Les résultats optimaux ont été obtenus avec *GeLU* comme fonction d'activation, un LR de 0,005, un *weight decay* de 0,0035,

AdamW comme fonction d'optimisation, une taille du réseau de 1536, un nombre de patches par soudure de 1500 et de points par patch de 100.

Tableau 4.6 : Résultats de *Hyperband* pour la sélection de hyperparamètres optimaux de *P2S*

Hyperparamètres	Sélection	Valeurs possibles
<i>LR</i>	0,005	0.005 à 3
<i>Weight decay</i>	0,0035	10^{-2} à 10^{-10}
Fonction optimisation	<i>AdamW</i>	<i>Adam</i> , <i>AdamW</i> , <i>SGD</i>
Fonction d'activation	<i>GeLU</i>	<i>ReLU</i> , <i>SeLU</i> , <i>ELU</i> , <i>LeakyReLU</i> , <i>Mish</i> , <i>Swish</i> , <i>SiLU</i> , <i>GELU</i>
Taille du réseau	1536	512, 1024, 1536
Points par <i>patch</i>	100	100, 200, 300
Patches par soudure	1500	1000, 1500, 2000

Au total, 64 différentes combinaisons d'hyperparamètres ont été testées pour un temps d'entraînement combiné de plus de 22 heures sur un ordinateur équipé d'un *GPU* Nvidia GeForce RTX 2080. Il faut préciser que *early stopping* est implémenté afin d'arrêter les entraînements lorsqu'aucune amélioration à la valeur de la perte n'est constatée sur 12 *epochs*.

La Figure 4.25 résume les résultats des combinaisons testées par *Hyperband*. Les sept graphiques présentés sont tous construits suivant ce principe : pour chaque combinaison testée, la meilleure valeur de perte est identifiée par un point où *x* est la configuration (ex. *SGD* ou *Adam*) et *y* la valeur de perte. La Figure 4.25 résume visuellement les résultats de toutes les configurations testées afin d'extraire des conclusions améliorant ainsi les connaissances sur l'entraînement du modèle *P2S*. La Figure 4.25 ne contient qu'un sous-ensemble des meilleurs résultats. En effet, seulement les configurations ayant obtenu une valeur de perte minimale plus faible que 0,25 ont été affichées. Cela permet de concentrer les analyses sur les meilleures configurations seulement.

Hyperband a permis de trouver la valeur du *LR* et du *weight decay* qui maximise les performances du modèle de reconstruction de surface. La valeur finale de *LR* est de 0,005 tandis que le *weight decay* est de 0,0035. On constate sur la Figure 4.25 que les meilleurs résultats sont tous obtenus avec un *LR* plus faible, entre 0,005 et 0,14. Le même constat est émis pour la valeur du *weight decay*. Les dix meilleurs résultats sont obtenus par des valeurs de *weight decay* entre 0,001 et 0,0035.

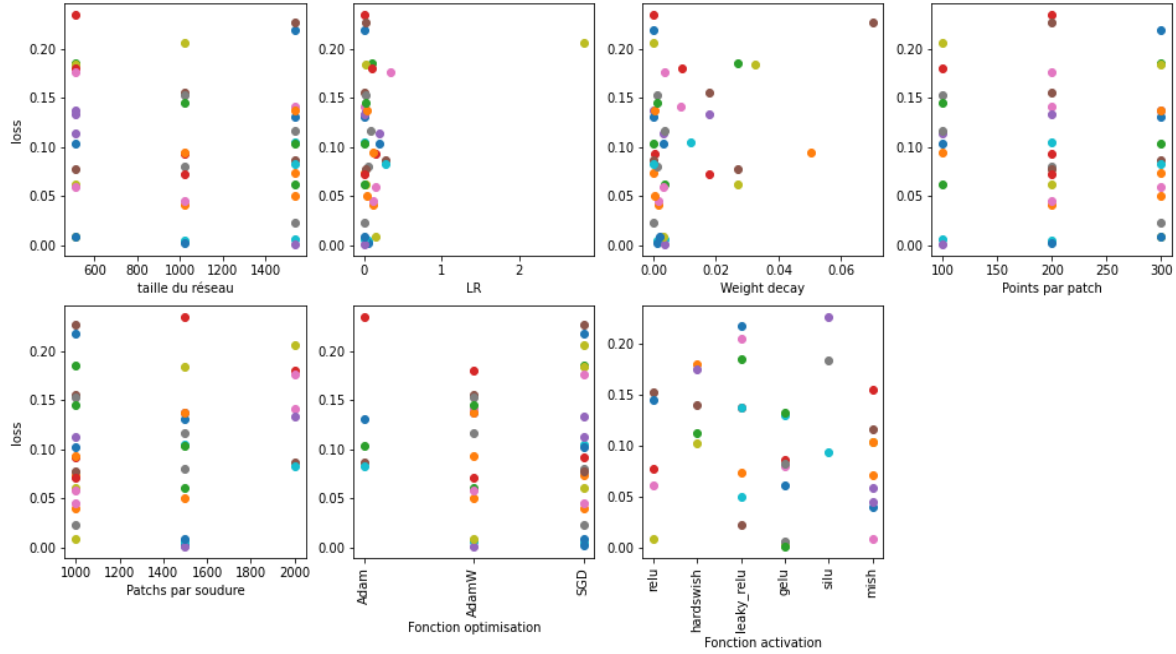


Figure 4.25 : Résultats de *Hyperband* représentés sous forme de nuage de points

La liste des fonctions d'optimisation incluses dans l'exécution de *Hyperband* a été réduite à seulement *Adam*, *AdamW* et *SGD*. Ces choix sont basés sur les résultats de l'exécution de *Hyperband* pour le modèle de segmentation multi-objectif où ces trois fonctions d'optimisation ont clairement démontré leur supériorité par rapport aux autres fonctions d'optimisation disponible. La fonction d'optimisation sélectionnée par *Hyperband* est *AdamW*. Comme discuté dans la section 0, l'implémentation de la régularisation l_2 a été améliorée par *AdamW*. Cela a pour effet d'augmenter les performances de *Adam*, une fonction d'optimisation adaptative, au même niveau que *SGD*. Les résultats de *Hyperband* montrent que *AdamW* et *SGD* surpassent largement *Adam*, comme illustré à la Figure 4.25. Bien que le meilleur résultat obtenu soit avec *AdamW*, on constate sur la Figure 4.25 que la concentration de bons résultats est plus importante pour *SGD* que pour *AdamW*. Parmi les 10 meilleurs résultats obtenus, quatre sont avec *AdamW* et six avec *SGD*.

La fonction d'activation optimale sélectionnée est *GELU*. Il est possible de remarquer à la Figure 4.25 que *GELU* surpasse largement les autres fonctions d'activation. Les quatre meilleurs résultats ont tous été obtenus avec *GELU*. En observant les 12 meilleures configurations, seulement *Mish* compétitionne avec *GELU* comme étant les meilleures fonctions d'activation pour *P2S*. En effet, parmi ces 12 configurations, *GELU* est utilisée cinq fois comme fonction d'activation et *Mish* quatre fois. Les 3 autres configurations utilisent *LeakyReLU* (2 fois) et *ReLU* (1 fois). Il semble donc y avoir un certain consensus sur l'utilisation de *GELU* pour cette étude.

La taille de réseau optimale sélectionnée par *Hyperband* est 1536, ce qui est la plus forte valeur disponible pour *Hyperband*. Comme démontré dans [211], l'effet de la taille de l'espace latent est faible à l'intérieur de la plage de valeur optimale. C'est pourquoi on peut constater à la Figure 4.25 que la distribution des résultats par taille est assez similaire. La taille de 1536 semble être celle ayant la plus forte concentration de bons résultats. En effet, dans les cinq meilleurs résultats obtenus par *Hyperband*, deux proviennent d'une taille de 1536, deux d'une taille de 1024 et un de 512. Le choix final de taille du réseau est 1536.

Le nombre de points par *patch* est un hyperparamètre important qui a un effet sur la qualité de la surface reconstruite, par la quantité d'information transmise à *P2S*, et sur le besoin en mémoire et en calcul. La meilleure configuration trouvée par *Hyperband* est obtenue à l'aide de 100 points par *patch*. Ce résultat permet de réduire le besoin en mémoire et en calcul, ce qui est bénéfique pour le modèle. Il faut toutefois préciser que le deuxième et le troisième meilleur résultat ont été obtenus avec 200 points par *patch*. On remarque à la Figure 4.25 que le nombre de points par *patch* ne semble pas avoir un effet très important sur la valeur de perte minimale. En effet, les distributions sont très semblables entre les trois valeurs disponibles. Cela justifie encore une fois de plus l'utilisation du plus faible nombre de points par *patch*. Les autres hyperparamètres semblent avoir davantage d'impact sur la valeur de la perte. Il faut rappeler que la forme des surfaces reconstruites est répétitive.

Le nombre de *patches* par soudure correspond à la quantité d'information sur la forme globale de la soudure que *P2S* obtient afin de reconstruire la surface. Tout comme le nombre de points par *patch*, cet hyperparamètre exerce une influence sur la qualité de la surface reconstruite, sur le besoin en mémoire et sur le besoin en calcul. Il est à noter que cinq des six meilleurs résultats ont été obtenus avec 1500 *patches* par soudure. En observant attentivement la Figure 4.25, on remarque que meilleurs résultats sont tous obtenus avec un nombre de *patches* par soudure de 1000 ou 1500. En somme, cette étude utilise 1500 *patches* par soudure.

Les essais de *Hyperband* ont été réalisés sans faire varier le *batch size*. Tout comme le modèle de segmentation multi-objectif, l'exécution de *P2S* nécessite une grande quantité de mémoire et un *batch size* élevé entraîne des problèmes de mémoire *RAM* au niveau du *GPU* utilisé (ayant une capacité de 11gb). Ainsi, une étude individuelle et approfondie du *batch size* a été réalisée afin de trouver la valeur optimale du *batch size* permettant d'éviter les problèmes de mémoire et de maximiser les performances de *P2S*. La recherche a permis de trouver que la valeur maximale pouvant être utilisée est de 93 *patches* par *batch*, ce qui prend en moyenne 12 minutes pour effectuer une *epoch*. Un entraînement, en utilisant ce *batch size*, a été effectué pour vérifier que les résultats du modèle l'utilisant sont de bonne qualité. Les résultats montrent que ce *batch size* n'a pas de

problème de mémoire et converge bien vers des résultats de bonne qualité (valeur de perte minimale < 0.01).

4.3.2 Entraînement du modèle *P2S*

P2S a été entraîné sur les 106 modèles *GT* pour apprendre un *prior* fort de la forme réelle des soudures et des défauts. Cet entraînement est illustré à la Figure 4.26. *Early stopping* avec une patience de 12 *epochs* a été utilisé afin de limiter le surapprentissage. Il est possible de constater que le modèle atteint une valeur de perte minimale de 0,000384 sur l'ensemble de validation à l'*epoch* 60. L'entraînement a nécessité un total de 72 *epochs* pour une durée de 835 minutes (13 heures et 55 minutes) sur un ordinateur équipé d'un *GPU* Nvidia GeForce RTX 2080.

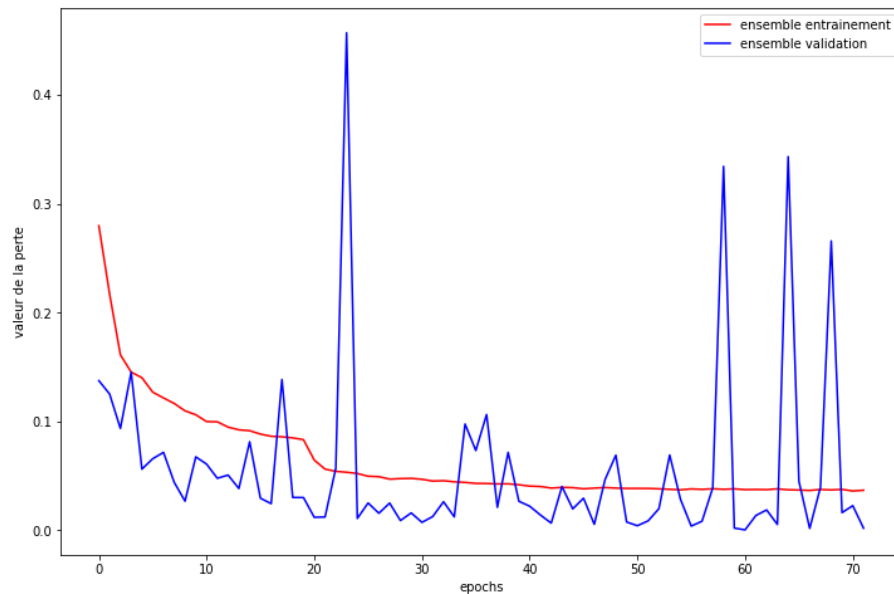


Figure 4.26 : Évolution de la valeur de la perte (ensemble d'entraînement et de validation) au fil des *epochs*

Considérant la fonction de perte utilisée par *P2S*, il est possible de calculer certains indicateurs de performance intéressants au cours de l'entraînement. Il convient de rappeler que la fonction de perte utilisée par *P2S* est une combinaison de la distance ℓ_2 —chanfrein et de l'entropie croisée. La distance ℓ_2 —chanfrein permet d'évaluer la distance absolue de la fonction de distance signée et l'entropie croisée binaire est utilisée pour évaluer la classification logit du signe. La fonction de perte finale est la somme de ces deux distances. Il est donc possible de calculer le score F1 sur la classification du signe. Le modèle obtient un score F1 de 96,5 %, grâce à une précision de 93,3 % et un rappel de 100 %. Cela correspond à une précision de :

$$Précision = \frac{VP}{VP + FP} = \frac{28}{28 + 2} = 93,3 \% \quad (4.4)$$

Le rappel équivaut à :

$$Rappel = \frac{VP}{VP + FN} = \frac{28}{28} = 100 \% \quad (4.5)$$

Finalement, le score F1 est calculé ainsi :

$$Score F1 = \frac{2 * précision * rappel}{précision + rappel} = \frac{2 * 0,933 * 1}{0,933 + 1} = 96,5 \% \quad (4.6)$$

Il est à noter que l'échantillon de validation consiste en 28 scans sur les 106 scans totaux de la base de données. Il y a eu 28 vrais positifs, 2 faux positif et aucun faux négatif.

4.3.3 Comparaison des techniques de reconstruction de surface

À l'inférence, *P2S* est appliqué au NP généré à partir des cartes de segmentation. Le Tableau 4.7 montre les résultats comparatifs de l'erreur de reconstruction pour *SPR* et *P2S* sur les données de Ondia. Il est à noter que la distance de chanfrein a été utilisée comme mesure comparative.

Tableau 4.7 : Comparaison quantitative de l'erreur de reconstruction

	<i>SPR</i>		<i>P2S</i>	
	Absolue	Relative	Absolue	Relative
<i>Prédictions — Bruit absent</i>	8,28	1,95	4,24	1,00
<i>Prédictions — Bruit faible</i>	8,52	1,48	5,76	1,00
<i>Prédictions — Bruit moyen</i>	8,93	1,55	5,78	1,00
<i>Prédictions — Bruit maximal</i>	9,57	1,55	6,18	1,00
<i>Moyenne</i>	8,81	1,63	5,49	1,00

On constate que *P2S* surpasse numériquement *SPR* en utilisant la distance de chanfrein comme indicateur de performance. *P2S* est en moyenne 1,63 fois meilleur que *SPR*. Il faut savoir que *P2S*, grâce à l'apprentissage d'un *prior* local, est très robuste au bruit et aux informations incomplètes. Les données utilisées pour créer le modèle géométrique provenant de la prédiction du *FCN* peuvent difficilement être considérées de haute qualité, car aucun contrôle n'est effectué avant l'utilisation de l'algorithme de reconstruction de surface. Étant donné la nature brute des NP générés à partir des résultats de la segmentation, la qualité de la surface reconstruite par *SPR* diminue rapidement avec une augmentation du bruit. L'utilisation d'une méthode de reconstruction de surface robuste est donc cruciale.

La Figure 4.27 montre quatre modèles géométriques créés par *P2S* et *SPR* ainsi que les modèles *GT* associés. On constate une fois de plus que *P2S* est plus robuste et plus fidèle à la représentation de la soudure. *SPR* semble produire un résultat plus fin, mais peut en revanche produire un résultat dégénéré (par exemple *SPR*, V).

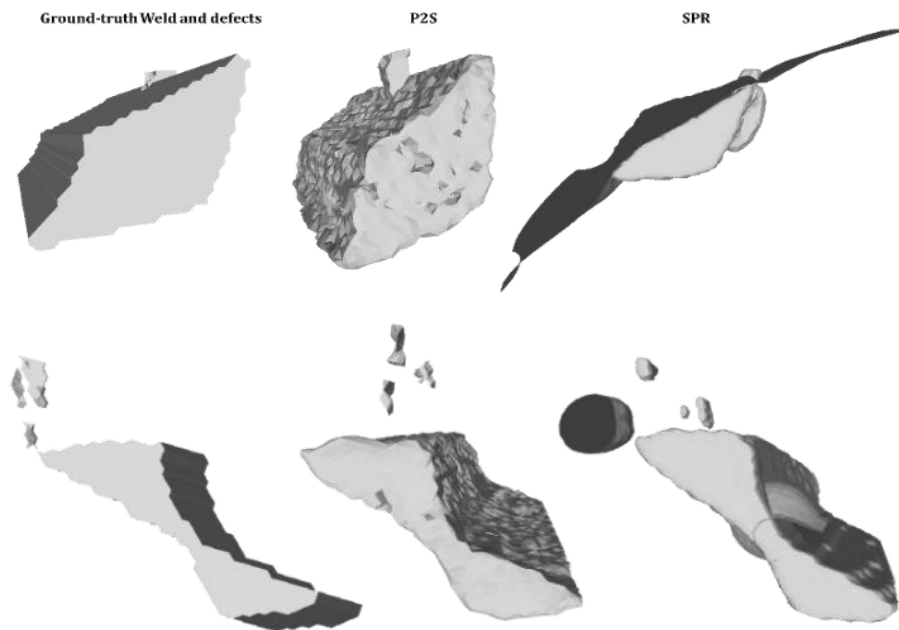


Figure 4.27 : Modèles géométriques reconstruits par *SPR* et *P2S*

Considérant que *P2S* est entraîné à reconstruire la surface de deux types de soudure, il est possible de comparer la qualité des surfaces reconstruites afin de savoir si *P2S* performe mieux avec l'un ou l'autre des types de soudure ou même les défauts. Les résultats présentés au Tableau 4.8 montrent que les surfaces des soudures reconstruites par *P2S*, autant les soudures en V que double-V, sont de bonnes qualités. Les erreurs de reconstruction étant très faible pour les NP obtenus directement du modèle de segmentation multi-objectif (prédictions sans bruit).

Tableau 4.8 : Comparaison quantitative de l'erreur de reconstruction de *P2S*

	V	Double-V	Défaut
<i>Prédictions — Bruit absent</i>	1,20	1,54	6,51
<i>Prédictions — Bruit faible</i>	3,50	2,89	6,57
<i>Prédictions — Bruit moyen</i>	3,63	3,00	6,37
<i>Moyenne</i>	2,78	2,48	6,48

Même lorsque le bruit augmente, la qualité des surfaces reconstruites demeure de bonne qualité par rapport aux défauts. La reconstruction de la surface des défauts est un défi largement plus difficile que les soudures. L'erreur moyenne de reconstruction pour les défauts (sans bruit) est de 6,51 et elle

demeure stable même si le niveau de bruit augmente. Ce phénomène est unique aux défauts, ce qui montre que *P2S* a davantage de difficulté avec les défauts. Toutefois, la qualité des surfaces reconstruites par *P2S* surpasse largement celles de *SPR*, et ce, même pour les défauts, comme le prouve la comparaison de leur moyenne respective au Tableau 4.7.

La Figure 4.28 montre la distribution de la distance de chanfrein de l'échantillon de validation pour les deux types de soudures et les défauts. On peut remarquer que *P2S* a plus de facilité à reconstruire la surface des soudures en V que des soudures en double-V. La plus grande erreur étant limitée à environ 5 alors que c'est 10 pour la plus grande erreur d'une soudure en double-V. On remarque également que la moyenne des distributions des résultats pour les deux types de soudure est faible, à 1,20 et 1,54, et la variance est assez faible également. En d'autres mots, il y a très peu de mauvaises reconstructions (distance de chanfrein > 4). *P2S* a beaucoup plus de difficulté à reconstruire la surface des défauts. En effet, la moyenne de la distribution des défauts est de 6,51, soit environ 4,5 fois supérieures à la reconstruction des deux types de soudures. La distribution des erreurs de reconstruction des défauts inclut des erreurs allant jusqu'à 35, ce qui est considérable. Il faut toutefois rappeler que les défauts, contrairement aux soudures, ont des formes moins récurrentes. Cela rend l'apprentissage du *prior* local beaucoup plus difficile. Pour améliorer la qualité des surfaces reconstruites par *P2S*, particulièrement au niveau des défauts, l'ajout de données d'entraînement est nécessaire. L'apprentissage de nouveaux défauts devrait permettre de renforcer la qualité du *prior* appris et augmenter la qualité des résultats.

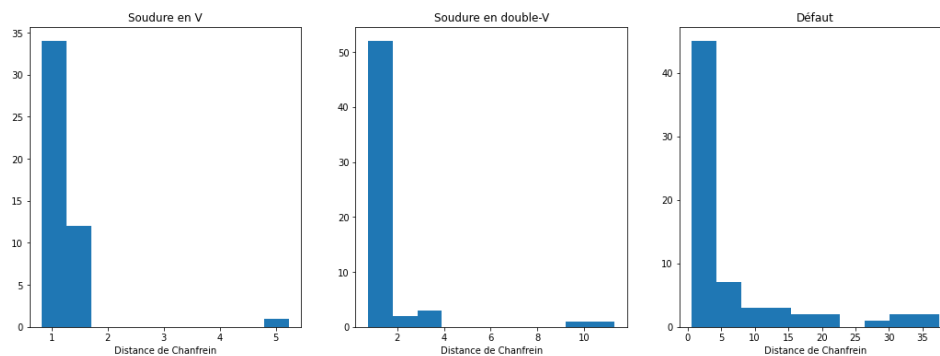


Figure 4.28 : Distribution des *DSC* de l'échantillon de validation pour les défauts

Les hyperparamètres exerçant une influence (seulement) sur la reconstruction de la surface sont optimisés pour obtenir le meilleur compromis entre la qualité de la surface et la durée de reconstruction. Il est important de noter que certains paramètres de *P2S* permettent de modifier le résultat visuel. La liste des hyperparamètres exerçant une influence sur la qualité de la reconstruction de la surface inclut la résolution de la grille, le seuil de confiance, les paramètres ϵ et σ . Ces hyperparamètres ont été optimisés à l'aide de l'algorithme *Tree of Parzen Estimators* [218]. Un total

de 10 essais a été réalisé. La solution optimale permet une reconstruction moyenne en seulement 95 secondes avec une distance de chanfrein moyenne de 4,24.

Tableau 4.9 : Optimisation des hyperparamètres de P2S

hyperparamètres	Possibilité	Résultat optimal
Seuil de confiance	4 à 26	17
ϵ	3 à 10	3
σ	3 à 9	3
Résolution de la grille	64, 80, 96, 112, 128	128
Nombre de points	5000, 10 000, 15 000, 25 000	10 000

4.4 Suivi périodique d'une soudure

Tout comme la méthode de détection des *ND*, l'évaluation de la méthode de reconnaissance des modèles géométrique a été réalisée à l'aide de plusieurs expériences. Pour chacune des expériences, la même méthodologie est utilisée. La méthode est d'abord appliquée à tous les modèles géométriques pour créer la base de données. Ensuite, la méthodologie présentée à la section 3.5 est suivie.

4.4.1 Étude de cas

La première expérience consiste en une étude de cas. Un modèle géométrique a été sélectionné, car la même soudure a été scannée à plusieurs reprises sous différentes conditions. Ce modèle géométrique possède donc plusieurs *ANN*. Entre autres, la soudure a été scannée des deux côtés (90° et 270°). Les paramètres d'acquisition de ces scans ont varié entre les acquisitions. L'objectif de cette expérience est de déterminer si la méthode de recherche des *ANN* retrouve bien les modèles géométriques des soudures correspondantes et de déterminer quelle mesure de similarité quantifie bien la ressemblance entre ces modèles géométriques.

Les modèles géométriques ont été échantillonnés afin de créer un NP de taille (4 000, 3). *LSH* et les mesures de similarité utilisent ce NP comme intrant. Aucun algorithme de réduction de dimensionnalité ni de techniques d'échantillonnage n'est utilisé. Il est important de noter que puisqu'il s'agit de la première expérience, la configuration de *LSH* n'est toujours pas optimisée.

Le résumé des résultats de cette expérience est présenté au Tableau 4.10. On y retrouve les modèles géométriques identifiés par *LSH*, la distance de chanfrein, de Hausdorff et de Wasserstein, le rang de ces mesures, lorsque comparé à l'ensemble des modèles géométriques de la base de données et finalement le rang moyen des trois indicateurs. Le modèle géométrique utilisée comme requête est le #37 270°. Comme il est possible de constater à l'aide des mesures de similarité utilisée, les modèles géométriques identifiés par *LSH* sont bien des doublons (à l'exception du #38), c'est-à-dire

que le modèle géométrique a été généré à partir de la même soudure. Il est à noter que tous les modèles géométriques trouvés par *LSH* sont des soudures en V.

Tableau 4.10 : Étude de cas de la soudure #37 270°

<i>ANN</i>	Chanfrein	Hausdorff	Wasserstein	Rang (CD, HD, WD)			Rang moyen
#38 270°	1,5251	2,8596	0,2084	2	1	1	1,33
#39 270°	1,7490	4,1589	0,4244	10	11	9	10,00
#36 90 °	1,6113	4,1901	0,3853	6	12	8	8,67
#37 90 °	2,0321	3,3670	0,3834	12	4	6	7,33
#38 90 °	1,5322	3,4748	0,2400	3	5	3	3,67
#39 90 °	1,5098	3,6540	0,2383	1	6	2	3,00
#41 90 °	1,5712	3,9605	0,2669	4	7	4	5,00
#68	1,6021	4,0163	0,3434	5	8	5	6,00

LSH a permis de trouver des modèles géométriques très similaires comme montré par le rang de chacun des *ANN* en fonction des différentes mesures de similarité. En effet, en utilisant la distance de chanfrein, *LSH* a trouvé les modèles suivants : les 6 plus proches et 9 des 12 plus proches. Ce genre de constat est le même pour les autres mesures de similarité. En somme, seulement un modèle géométrique provenant d'une soudure différente (#68) se glisse parmi les dix plus proches voisins. Ce modèle géométrique a un rang moyen de six grâce à une distance de chanfrein de 1,6021 (5^e), une distance de Hausdorff de 4,0163 (8^e) et une distance de Wasserstein de 0,3434 (5^e). Au niveau de la sélection de la mesure de similarité, on peut remarquer que les distances de chanfrein et de Wasserstein ont des résultats similaires au niveau du rang de chacun des modèles géométriques. La distance Hausdorff ne semble pas quantifier la similarité entre les modèles géométriques de façon aussi précise. En effet, parmi les plus faibles mesures, on ne retrouve pas les modèles géométriques provenant de la même soudure, ce qui représente un problème. Au regard des résultats du Tableau 4.10, la mesure de similarité choisie est chanfrein. Les raisons qui expliquent ce choix sont que la distance de chanfrein est déjà utilisée pour mesurer la qualité de la reconstruction des modèles géométriques et que les résultats montrent que la mesure quantifie bien la similarité entre les modèles géométriques.

La Figure 4.29 montre le modèle géométrique étudié et les modèles géométriques des fichiers listés dans le Tableau 4.10 qui ont la plus forte similarité avec le modèle étudié. À la première ligne, il s'agit du modèle étudié (#37 270°) et de deux modèles ayant une forte similarité : #38 270°, #39 90°. La deuxième ligne montre trois autres modèles ayant une forte similarité : #38 90°, #41 90° et #68.

On voit clairement que les fichiers identifiés par *LSH* sont effectivement des modèles géométriques similaires, à l'exception du dernier (#68) qui est le seul qui ne provient pas de la même soudure. Visuellement, il est difficile de différencier ces modèles géométriques alors que *LSH* et les mesures de similarité sont en mesure de le faire efficacement. Le fait d'inclure les défauts au processus de détection des modèles géométriques permet d'améliorer la qualité des résultats. En effet, les

soudures, que ce soit en V ou double-V, ont des formes récurrentes. Plusieurs soudures se ressemblent grandement alors que les défauts ont des formes plus particulières. Ces formes particulières permettent à *LSH* de bien distinguer les soudures entre elles.

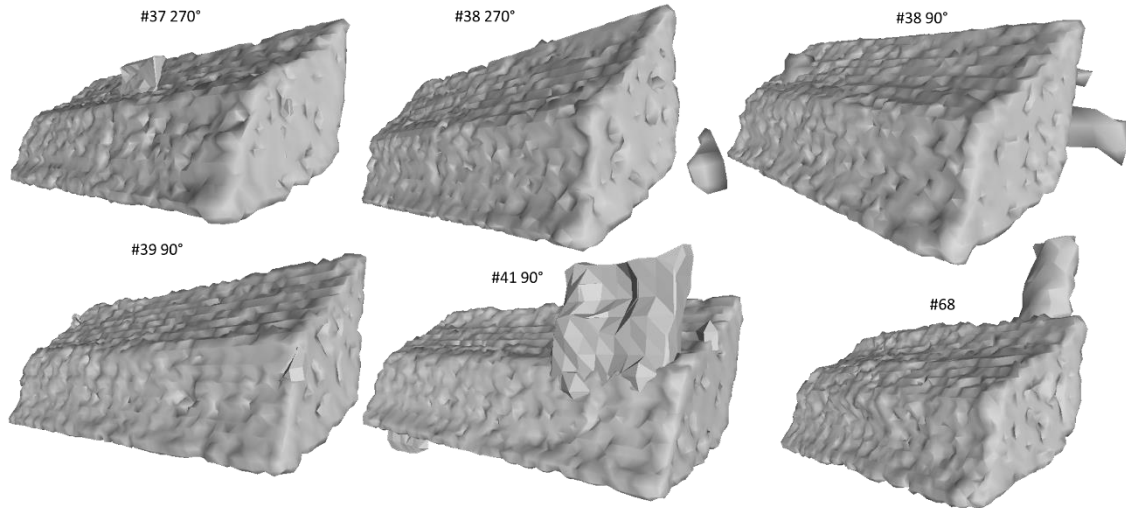


Figure 4.29 : Étude de cas de détection des modèles géométriques : le modèle géométrique étudié et les plus proches voisins

4.4.2 Valeur du seuil de similarité

La deuxième expérience consiste à déterminer la valeur de seuil qui permet de qualifier un modèle géométrique de doublon d'une même soudure. Pour déterminer ce seuil, un échantillon de 12 scans provenant tous de la même soudure a été utilisé comme mesure d'évaluation. L'objectif est de trouver la valeur pour laquelle un premier modèle géométrique qui provient d'une soudure différente est rencontré. La Figure 4.30 résume visuellement les résultats de cette expérience. Pour chaque modèle géométrique de l'échantillon, les dix mesures de similarité (distance de chanfrein) les plus faibles sont ajoutées au graphique. Les points en noir correspondent à des modèles géométriques provenant de la même soudure et les points en rouge représentent le modèle géométrique qui ne provient pas de la même soudure ayant obtenu la plus faible mesure de similarité.

Le seuil de similarité optimale consiste au seuil qui permet de distinguer les poids en noir des poids en rouge. On constate sur la Figure 4.30 que cette valeur est d'environ 1,60 considérant que tous les points en rouge sont au-dessus de ce seuil et qu'au moins un point noir est en dessous de ce seuil. Il y a seulement trois exceptions à ce constat. Il s'agit du premier modèle géométrique où la plus faible valeur recensée est de 1,77. Dans un cas comme celui-ci, il est considéré qu'il n'y a aucun modèle géométrique similaire.

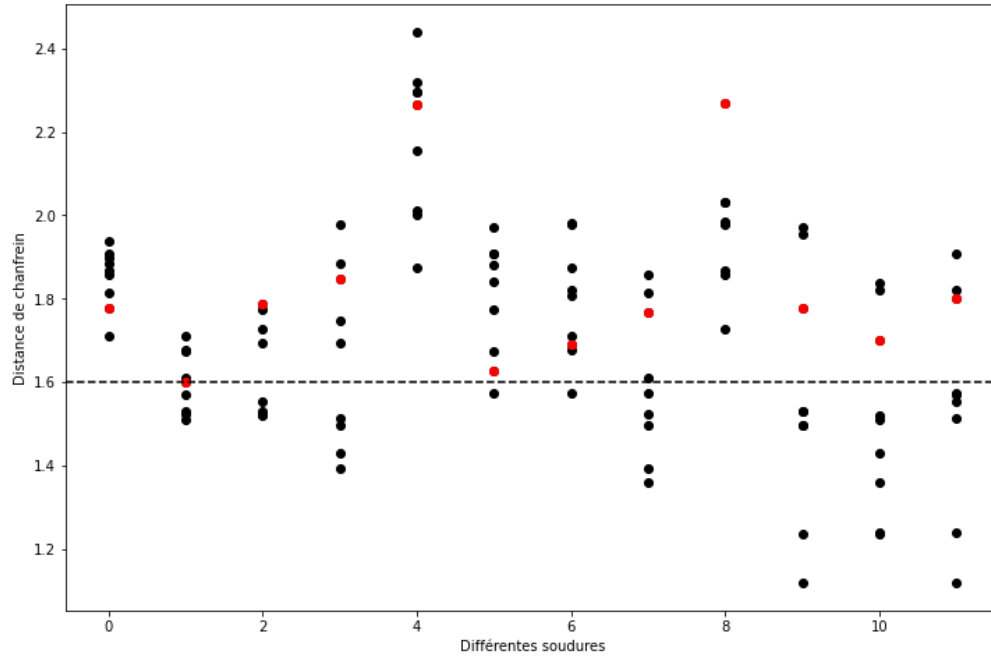


Figure 4.30 : Résumé visuel de l'expérience sur le seuil

4.4.3 Configuration de *LSH*

La troisième expérience consiste à évaluer différentes configurations de *LSH* afin de trouver la meilleure configuration. L'objectif est de trouver la configuration offrant le meilleur compromis entre le nombre de *ANN* trouvé, la meilleure précision et le plus faible délai d'exécution. Les différents paramètres modifiés consistent au nombre de projections, au nombre de *hash* utilisé, la taille des intrants et les indicateurs de performance sont le nombre de *ANN* identifié moyen, le délai moyen et la précision. Les résultats sont présentés au Tableau 4.11.

Tableau 4.11 : Liste des configurations testées et leurs résultats

Config	Nombre de projections	Nombre de <i>hash</i>	Taille des intrants	<i>ANN</i> moyen	Délai moyen (s)	Précision (%)
1	30	4	4 000	107	2,4	100
2	10	2	4 000	10	0,5	92
3	10	2	10 000	10	0,6	42
4	20	4	10 000	80	3,1	58
5	10	2	6 000	10	1,5	75

4.4.4 Maximisation des performances

Les résultats de la dernière expérience permettent de choisir la meilleure configuration de *LSH* afin de maximiser les performances de ce processus. On remarque que les meilleurs résultats sont

obtenus à l'aide de NP échantillonnés de taille (4000, 3), de 10 projections et de deux engins *LSH*. Cette combinaison permet d'obtenir une précision de 92 %, un nombre moyen de *ANN* de 10, et ce pour un délai moyen de 0,5 seconde par modèle géométrique. Il est à noter que l'augmentation de la taille des intrants ne se traduit pas en augmentation des performances, ce qui peut sembler contre-intuitif. Également, l'ajout d'engin *LSH* se traduit en une trop grande sensibilité, ce qui occasionne des délais supplémentaires et des calculs inutiles (pour l'implémentation dans le logiciel infonuagique de Ondia).

4.4.5 Résumé des quatre expériences

Un résumé des expériences est présenté au Tableau 4.12. Il résume les différentes expériences effectuées afin de montrer l'efficacité de la méthode développée et d'optimiser ses performances pour la situation à l'étude. Comme les différentes expériences l'ont montré, la méthode de détection est en mesure de bien retrouver les modèles géométriques similaires (provenant d'une même soudure). Il est à noter que l'ajout des défauts aux intrants de ce processus réduit la complexité du problème et permet une reconnaissance des soudures plus facile.

Tableau 4.12 : Résumé des expériences

Expérience	Variables testées	Conclusion
1	Efficacité générale de <i>LSH</i> à l'aide d'une étude de cas	<i>LSH</i> est en mesure de retrouver les modèles géométriques similaires
2	Valeur du seuil τ	$\tau = 1,60$
3	Configurations	La meilleure configuration est la #2

Il est à noter que le processus développé peut être automatisé, mais, dans certains cas, un analyste humain doit intervenir pour modifier une décision considérant la similarité des soudures de même type.

4.5 Implantation

4.5.1 Exemple d'utilisation

Afin d'illustrer l'utilisation du processus développé dans cette étude, la Figure 4.31 a été développée. On peut y voir le processus complet de façon synthétisé. Le processus débute par un scan *PAUT* 3D qui est analysé pour détecter des doublons (*ED* ou *ND*). Si aucun doublon n'est détecté, le processus se poursuit par la segmentation de la soudure et des défauts. Ensuite, les cartes de segmentation produites sont transformées en NP afin de reconstruire la surface de la partie principale de la soudure et des défauts. Le modèle géométrique créé est ensuite utilisé afin de retrouver s'il a déjà été scanné. Lorsqu'il a déjà été scanné, un suivi périodique est effectué grâce à une comparaison visuelle des modèles géométriques.

On peut voir à la Figure 4.31 des exemples visuels de certains extrants des étapes du processus. Le premier exemple est l'intrant au processus, c'est-à-dire, un scan *PAUT* 3D. Le second exemple visuel est une comparaison entre deux scans. Le scan de gauche (scan 1) est le scan intrant au processus alors que celui de droite (scan 2) est le plus proche voisin. On retrouve au centre un S-scan de similarité. On peut voir que bien que ce sont des *ANN*, ce ne sont pas des *ND*. Le processus est donc poursuivi. Le troisième exemple visuel est le résultat de la segmentation : les différentes parties de la soudure et les défauts. Le quatrième exemple est le modèle géométrique issu de la reconstruction de la soudure et des défauts (à partir de la segmentation). On peut voir qu'il s'agit d'une soudure en V et qu'un défaut est présent au milieu du modèle géométrique, un peu au-dessus du centre de la soudure, probablement dans la ZAT. Le dernier exemple visuel est une comparaison de deux modèles géométriques. Celui de gauche est le modèle géométrique généré par ce processus alors que celui de droite est celui du modèle géométrique le plus similaire (provenant de la même soudure). On peut voir que les deux modèles sont très similaires. Les deux modèles sont des soudures en V ayant un défaut sensiblement à la même position.

L'exécution du processus complet prend moins d'une minute par scan *PAUT* 3D (en Python sur Jupyter Notebook). La détection des doublons prend environ 3 secondes. La segmentation de la soudure et des défauts prend ensuite environ 10 secondes par scan. L'étape la plus longue est la reconstruction de la surface qui prend environ 95 secondes par scan. Finalement, le suivi périodique prend environ 5 secondes. Au total, le processus prend en moyenne 1 minute 53 secondes. Il est à noter que le délai d'exécution de ce processus pourrait être optimisé lors de son implémentation dans le logiciel infonuagique de Ondia. Par exemple, il est possible de paralléliser les calculs effectués à certaines étapes de ce processus. Une seconde façon de réduire le délai d'exécution est de faire appel à des serveurs infonuagique de très grande capacité.

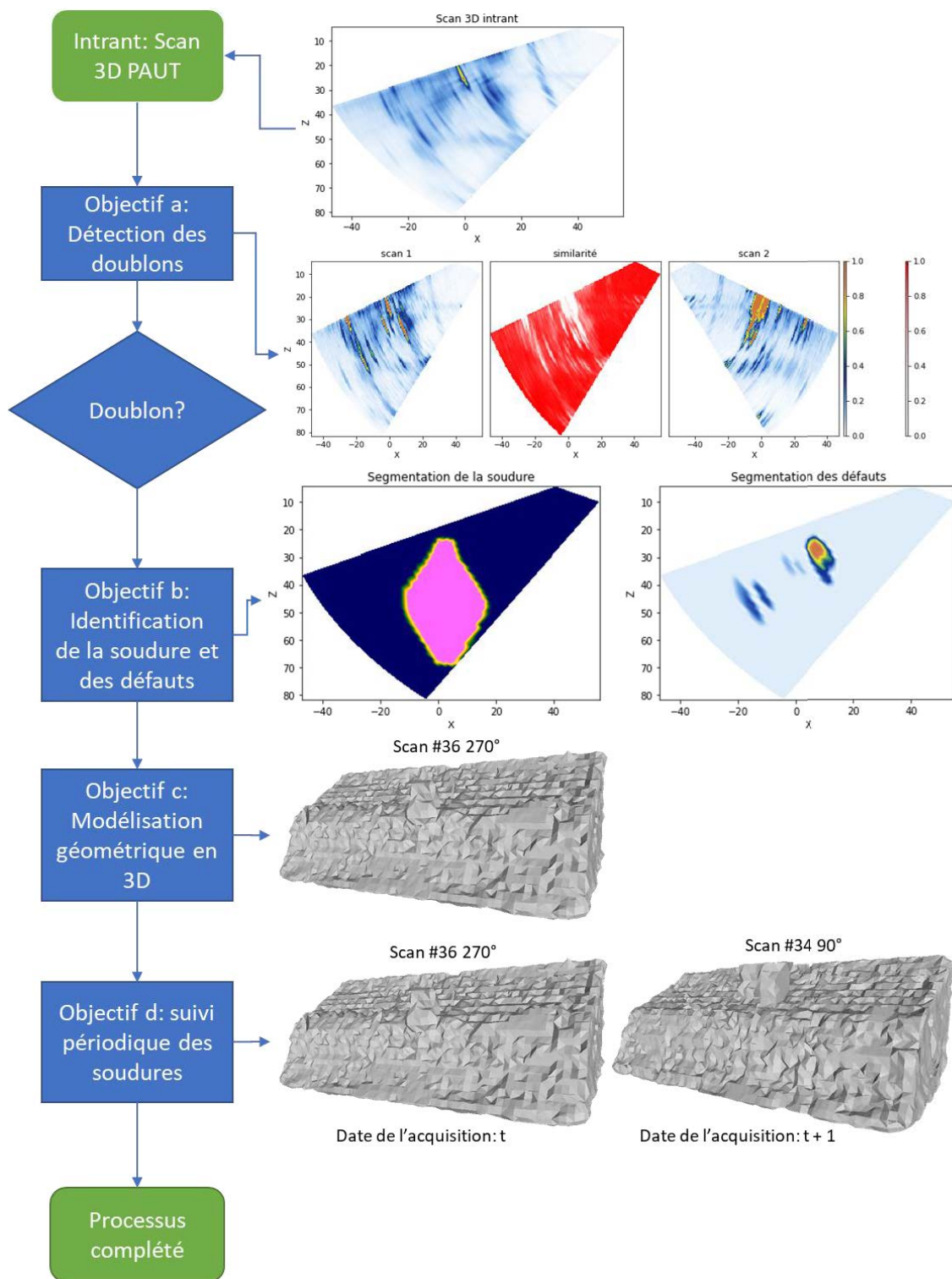


Figure 4.31 : Synthèse du processus développé dans cette étude

4.5.2 Implémentation dans un logiciel infonuagique

Dans un avenir rapproché, chaque scan *PAUT* téléchargé dans le logiciel de détection des défauts de soudure sera transmis à un processus d'analyse complet qui comprend un contrôle qualité des scans d'entrée (absence de couplant, etc.), la détection des doublons (en cas d'erreurs de manipulation ou de fraude) [219], détection des défauts de soudure, segmentation de la forme de soudure, modélisation de la soudure et des défauts et suivi périodique. La plupart des outils utilisent les dernières avancées en matière d'apprentissage profond pour fournir des informations de haute qualité à leurs utilisateurs. Le logiciel permet une analyse en temps quasi réel des scans de soudure ultrasonores de partout sur la planète. La qualité des méthodes d'analyse basées sur l'apprentissage profond permet de raccourcir le délai entre l'acquisition des données du terrain et la décision de l'analyste certifié.

Les deux méthodes de détection de doublons sont déjà implantées dans le logiciel infonuagique de Ondia pour permettre la détection des doublons rapidement dans le processus d'analyse de scans ultrasonores. Chaque scan ultrasonore téléchargé dans le logiciel d'analyse de soudure est ainsi transmis au service de duplication qui crée un rapport pour tous les fichiers. Les deux processus de détection des doublons sont activés en parallèle. Lorsqu'un *ED* est rencontré, le processus d'analyse est arrêté pour ce fichier et l'utilisateur reçoit un avertissement.

Le processus de détection des *ND* crée un rapport qui affiche les cinq voisins les plus proches ainsi que leur score de similarité. Au-dessus du seuil τ , déterminé à l'origine par un analyste certifié de Ondia (*PAUT* niveau III par le *Certification Scheme for Weldment Inspection Personnel*), le scan ultrasonore est identifié comme un *ND*. Dans ce cas, l'utilisateur doit déterminer si le scan réellement un *ND* à l'aide des informations fournies dans le rapport. Le seuil τ peut être ajusté par l'utilisateur pour augmenter la précision et le rappel du processus. La Figure 4.32 montre l'interface du logiciel infonuagique de Ondia. Le rapport disponible pour chaque projet peut être vu à gauche.

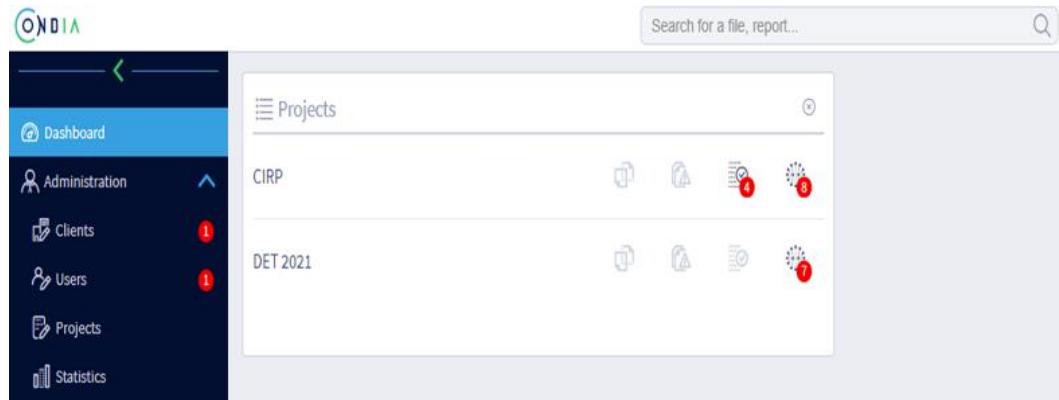


Figure 4.32 : Interface du logiciel Ondia : intégration du processus de détection des doublons

La Figure 4.33 montre le rapport de détection des doublons pour un scan ultrasonore choisi au hasard. Le fichier téléchargé (affiché comme un C-Scan) est le premier scan affiché dans l'image tandis que les C-Scan suivants sont les plus proches voisins. Le rapport indique le nom des fichiers ainsi que le score de similarité.

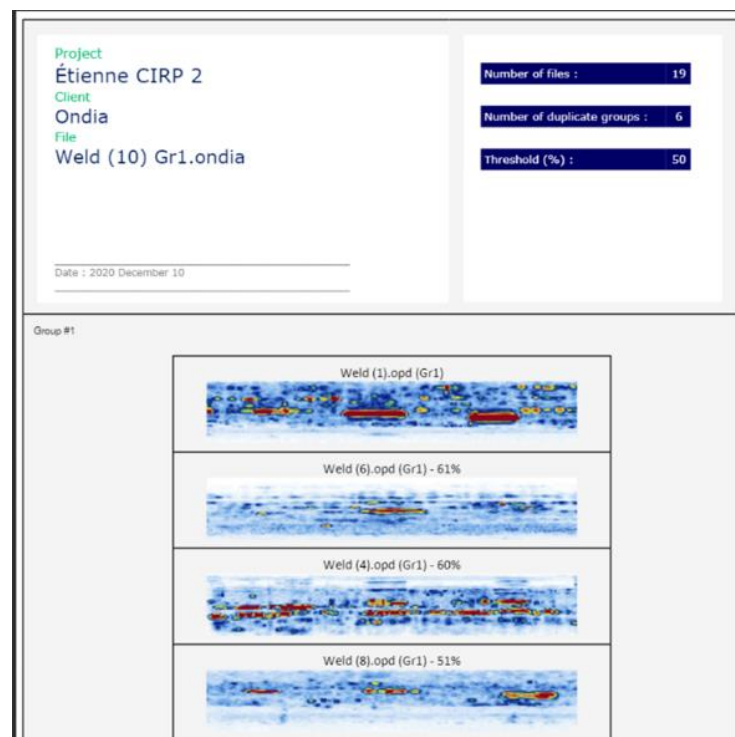


Figure 4.33 : Rapport de duplication pour un scan ultrasonore choisi aléatoirement

L'outil de segmentation de forme de soudure développé dans cette étude sera implémenté dans le logiciel infonuagique d'analyse de scans ultrasonores. L'outil de segmentation s'ajoute à une liste

d'outils d'analyse déjà disponibles tels que la détection des doublons de scans ultrasonores, la visualisation interactive, la gestion de bases de données et la détection de défauts de soudage.

Le succès de cette étape se solde par une augmentation considérable de la qualité de l'interprétation du défaut identifié : position et orientation sur un dessin en 3D et identification de la zone dans laquelle le défaut se trouve (métal fondu, métal de base, où à cheval entre les deux). L'intégration de la géométrie des défauts au dessin 3D de la soudure se fera à l'aide de libraires graphiques identifiées par Ondia à l'étape précédente. Le dessin final fera éventuellement partie de l'affichage disponible aux utilisateurs du logiciel de Ondia. Un affichage interactif permettra aux utilisateurs de voir le modèle géométrique sous tous ces angles.

L'outil de suivi périodique sera également implémenté dans le logiciel d'analyse de Ondia. Lorsqu'un modèle géométrique sera trouvé, une comparaison visuelle sera effectuée afin de visualiser les différences entre les deux modèles. De l'information sur les deux acquisitions de données sera également affichée afin de connaître la durée de vie utile entre les deux scans et connaître les autres raisons qui pourraient avoir occasionné des différences, comme certains paramètres du *PAUT*. Une échelle de couleur mettant l'accent sur la différence entre les deux modèles géométriques pourrait également être utilisée. Cela permettrait de voir rapidement les différences entre les deux modèles sans avoir à les chercher.

Il est important de souligner que l'ensemble des étapes du processus d'analyse développé dans cette recherche sont automatisés et ne requiert aucune interaction avec les utilisateurs. L'utilisateur n'a besoin que de soumettre un ou des fichiers à analyser et le processus s'exécutera de la détection des doublons jusqu'au suivi périodique.

5 CONCLUSION

5.1 Retour sur les objectifs et contributions scientifiques de la thèse

Cette recherche avait pour objectif de développer des méthodes d'IA afin de valider l'authenticité des données ultrasonores de soudure, de segmenter les données appartenant à la soudure, de visualiser la soudure en tant qu'un objet en trois dimensions et de permettre un suivi périodique d'une même soudure. Pour y arriver, une revue de littérature orientée vers les objectifs spécifiques a été présentée. Il semble qu'il n'existe aucun exemple dans la littérature ayant répondu directement à l'un ou l'autre des quatre objectifs spécifiques. Cette recherche a donc adapté des méthodes utilisées dans d'autres contextes et domaines afin de parvenir à résoudre la problématique spécifique à l'origine de cette étude. Les résultats permettent de limiter le recours aux méthodes traditionnelles de classification automatique des défauts grâce au déploiement des nouvelles méthodes développées dans cette recherche dans un logiciel commercial d'analyse des scans *PAUT* 3D. Cela devrait permettre de mettre en œuvre rapidement les avancées de cette étude pour des applications industrielles réelles.

Cette recherche explique d'abord l'importance de trouver les doublons dans le domaine des END. Considérant l'importance des soudures dans les structures, l'absence d'analyse représente un risque sur la sécurité, l'économie et l'environnement de grande ampleur pour les utilisateurs et pour la société elle-même. Il est donc nécessaire de disposer d'une méthode de détection des doublons robuste et efficace. Un processus efficace visant à détecter tous les types de doublons provenant de scans ultrasonores a été développé. Les méthodes utilisées dans la littérature pour détecter les doublons ont été examinées. Quatre expériences ont été conçues pour déterminer les configurations optimales pour la détection des *ND* et évaluer sa capacité de détection. Les résultats démontrent l'efficacité des méthodes de détection des doublons en l'appliquant à une base de données privée de scans ultrasonores 3D. Les résultats montrent que les deux méthodes sont robustes, efficaces et peuvent être appliquées à un très grand ensemble de données avec une excellente efficacité algorithmique. Le coût de calcul est minime, même avec un ensemble de données de centaines de scans ultrasonores. En effet, tous les doublons ajoutés intentionnellement à la base de données ont été détectés et identifiés correctement, qu'ils soient *ED* ou *ND*. Ces méthodes ont été ajoutées au logiciel en ligne de Ondia, permettant la détection des doublons rapidement dans le processus d'analyse. Le principal défi derrière cette contribution était de trouver le compromis entre la précision de détection (les scans *PAUT* sont très similaires) et l'efficacité de calcul, étant donné la très grande quantité de scans analysés par le service infonuagique de Ondia.

Cette recherche présente ensuite une nouvelle approche d'analyse des soudures à partir des données issues d'END. Le modèle sélectionné est un *FCN* qui effectue la segmentation unifiée des parties principales des soudures et des défauts pour ensuite en créer une représentation 3D fidèle. Le modèle prédit l'appartenance de tous les voxels aux différentes parties de la soudure et des défauts, permettant une classification et une représentation précise de la forme de la soudure et des défauts. Le modèle de segmentation atteint une précision de 97,7 % au niveau des voxels et un *DSC* de 90,47 % sur l'ensemble de données de test. Il faut préciser qu'il s'agit de l'une des premières études à effectuer la segmentation de la soudure et des défauts à partir de données ultrasonores 3D. Bien que l'ensemble de données ne comprenne que deux types de soudures, le modèle réussit à atteindre des performances intéressantes. Les résultats montrent que le modèle est capable de détecter les soudures dans les données ultrasonores tirées de données réelles. Le modèle proposé permet d'identifier les parties importantes de la soudure, comme la forme principale de la soudure (V ou double-V), le matériel de base et même la ZAT. La segmentation de la forme de la soudure est une innovation importante pour l'amélioration de la qualité des analyses des soudures. En effet, puisque l'importance d'un défaut est largement influencée par son contexte géométrique, le processus proposé ouvre de nouvelles perspectives dans l'analyse des scans ultrasonores pour l'industrie END. Il est important de rappeler que le modèle *FCN* a été entraîné à partir d'un nombre très limité de données, soit seulement 106 scans *PAUT* 3D, ce qui affecte ses performances et sa robustesse. Une augmentation du nombre d'échantillons générerait très probablement des résultats améliorés. L'apprentissage d'un modèle 3D implique beaucoup plus de paramètres qu'un modèle 2D, ce qui nécessite un plus grand ensemble de données d'échantillons d'apprentissage [48].

Les cartes de segmentation produites sont ensuite transformées en nuage de points dans le but de créer un modèle géométrique de la soudure et des défauts. Un modèle de reconstruction de surface basé sur l'apprentissage d'un *prior* a été sélectionné. Ce modèle offre, à la fois, de bonnes performances de reconstruction et une bonne performance algorithmique, permettant la reconstruction d'une soudure et d'un défaut en environ 95 secondes. L'un des grands avantages de *P2S* est qu'il est robuste au bruit. Il reconstruit des surfaces de qualité, et ce même si le nuage de points intrant est bruité. Le modèle atteint une erreur de reconstruction moyenne de seulement 4,24 en utilisant la distance de chanfrein alors que l'erreur de reconstruction est 1,95 fois plus importante lorsqu'on compare avec la reconstruction de surface par *SPR*. En plus, *P2S* souffre moins de la dégradation de la qualité des NP intrants que *SPR*. *P2S* surpasse *SPR* dans toutes les conditions, justifiant ainsi son utilisation dans cette étude. En somme, cette étude introduit une nouvelle représentation de la soudure grâce au processus de reconstruction de la surface des soudures et des défauts. Il est alors possible de visualiser la soudure et les défauts en 3D. L'ajout d'un tel outil de visualisation amène de nouvelles informations très importantes aux analyses. En particulier, il est

dorénavant possible de visualiser les défauts, de connaître leur forme précise, leur taille précise et leur localisation par rapport à la soudure.

La dernière contribution de cette recherche est l'ajout d'une méthode capable de retrouver une soudure déjà scannée afin d'en faire un suivi périodique. La méthode permet de retrouver les modèles géométriques déjà scannés à partir de la méthode *LSH* utilisée dans la détection des doublons. Le processus est cependant modifié afin de pouvoir prendre les modèles géométriques en intrant. Pour ce faire, les surfaces sont échantillonnées afin de récolter 4000 points. Le processus est en mesure de retrouver les soudures déjà scannées de façon précise et efficace, permettant ainsi son intégration dans le logiciel infonuagique de Ondia. Cela permet de faire une comparaison visuelle des soudures à différentes étapes de leur vie utile. Il est alors possible de constater la dégradation (ou la détérioration) d'une soudure ou d'un défaut entre deux scans. Un suivi périodique d'une soudure est également possible, afin de s'assurer qu'un défaut, par exemple, n'occasionne aucun risque pour la structure. Il est à noter que ce genre de suivi périodique est très difficile à faire sans modèle géométrique ou outil de comparaison. En effet, isoler les éléments importants d'une soudure devient pratiquement impossible si on considère la variation des paramètres d'acquisition et les différences entre les scans manuels.

Le processus complet développé dans cette étude peut être entièrement intégré dans le logiciel infonuagique de Ondia. Chacune des méthodes développées a été optimisée afin d'offrir les meilleures performances, la meilleure efficacité algorithmique et le plus faible délai d'exécution. L'exécution du processus complet ne prend qu'une minute, ce qui fait que son intégration dans le logiciel devrait se solder en un processus d'analyse rapide et efficace, permettant entre autres de réduire le délai entre l'acquisition des scans et le diagnostic par l'analyste certifié. La détection des doublons a déjà été intégrée au logiciel de Ondia et les autres aspects le seront prochainement.

5.2 Limitations et perspectives

La principale limite de la méthode de détection des *ND* est que l'algorithme de recherche des plus proches voisins utilisé pour identifier les *ND* implique le choix de différents paramètres (*LSH*, nombre de projections, seuil) pour optimiser la précision de la méthode de détection. Les paramètres ont été sélectionnés en fonction des données actuelles. Ces paramètres doivent être réévalués lors de l'ajout d'autres fichiers. Pour ce faire, il implique de refaire les expériences réalisées dans cette étude. Par conséquent, la méthode de détection peut être basée sur des paramètres sous-optimaux si les mises à jour des paramètres ne sont pas effectuées assez fréquemment. Puisqu'il s'agit de la même méthode utilisée (*LSH*), le même constat est valide pour la méthode permettant de retrouver les modèles géométriques pour effectuer un suivi périodique. Il est nécessaire de réviser les différents

paramètres choisis périodiquement ou lors de l'ajout de quantité importante de données afin de s'assurer de l'efficacité de la technique.

Une perspective d'amélioration du processus de détection des *ND* est l'utilisation d'une représentation encodée du scan *PAUT* pour réduire l'utilisation de la mémoire et la puissance de calcul nécessaire tout en maintenant l'efficacité de la méthode de détection des *ND* à un niveau élevé. Un auto-encodeur pourrait être utilisé pour comprimer les scans originaux en une taille compacte, mais représentative, ce qui pourrait contribuer à améliorer la robustesse de cette méthode. Une autre amélioration des méthodes de détection des doublons serait d'utiliser le modèle géométrique 3D de la soudure et des défauts comme intrant au lieu d'un scan ultrasonore 3D. Cela permettrait un processus plus robuste et plus précis puisque les informations fournies à l'algorithme de recherche des plus proches voisins représenteraient uniquement la soudure sans l'impact du processus d'acquisition des données ultrasonores qui inclut inévitablement l'impact des choix des paramètres (valeur du gain, nombre de *samples*, filtre, etc.) et le plan de numérisation.

Les principales limitations du processus de segmentation et de reconstruction de surface sont liées à la distribution des données utilisées. En effet, le modèle de segmentation ne peut segmenter que des données similaires à celles qui ont été utilisées pour l'entraînement. Ainsi, le modèle n'est en mesure de segmenter que des soudures en V et double-V et les défauts répertoriés. Il faut aussi préciser que le modèle performe bien seulement pour des scans de tailles similaires à celles utilisées. Ainsi, les scans ayant un nombre de *frames* de plus de 500, ayant plus de 32 *beams* ou plus de 640 *samples* sont susceptibles de ne pas produire de bonnes prédictions. Le même constat est aussi valide pour le modèle de reconstruction de surface, qui a appris à reconstruire la surface de soudures en V et double-V exclusivement. Le modèle bénéficierait d'un entraînement avec davantage de variabilité dans les scans. Dans un avenir proche, les processus de segmentation et de reconstruction de surface seront appliqués à davantage de types de soudure. Des scans de différentes tailles devraient également être ajoutés à la base de données. Cela devrait augmenter, à la fois, les performances du modèle et sa capacité de généralisation. Il faut aussi préciser que le processus de segmentation est limité par la mémoire disponible au niveau du *GPU*. Il est alors nécessaire de réduire la taille des scans avant de les transmettre au modèle, ce qui peut occasionner une perte de précision, et ce même si les données sont retransformées dans leur format original. Des solutions pour réduire cette limitation sont déjà envisagées. Il est également important de souligner l'importance de rester à l'affût des nouvelles percées du domaine de l'IA. En effet, les nouvelles avancées sont fréquentes et régulièrement de nouveaux modèles deviennent une nouvelle référence de performances (efficacité, vitesse d'exécution, taille du modèle) en surpassant les modèles précédents. Cette veille technologique pourrait permettre de rapidement améliorer les performances de cette étude. Il est à noter que lors de l'ajout de données massives (ou sur une base périodique), il peut être nécessaire de revoir le choix des hyperparamètres des modèles de segmentation et de

reconstruction de la surface afin de s'assurer que les hyperparamètres utilisés sont toujours les hyperparamètres optimaux.

Une des perspectives intéressantes concernant la reconstruction de surface est l'utilisation d'un modèle qui soit en mesure de produire directement un modèle géométrique 3D, un maillage de surface, à partir d'un scan *PAUT* 3D. À l'heure actuelle, il ne semble pas y avoir de modèle capable d'exécuter cette tâche efficacement.

Une perspective très intéressante afin de raffiner davantage le processus d'analyse des soudures est l'ajout d'une méthode qui utiliserait le modèle géométrique généré pour effectuer une analyse d'éléments finis. Cela permettrait de prédire le comportement de la soudure selon les charges dans son milieu environnant. L'analyse des éléments finis est une méthode qui permet de comprendre le comportement physique de structures complexes, de prédire leurs performances et leurs comportements, de prédire leurs marges de sécurité et d'identifier leurs faiblesses. En utilisant l'analyse des éléments finis, on mettrait à profit toutes les informations générées dans le processus complet afin de vraiment connaître la criticité de certains défauts. La mise en place de cette analyse demande toutefois la modification du modèle géométrique. Le choix d'une méthode de reconstruction de surface a été fait, car le modèle géométrique générée actuellement n'est utilisé que pour représenter visuellement la soudure et les défauts. Pour faire l'analyse des éléments finis, il serait toutefois nécessaire d'avoir un maillage tétraédrique. Pour passer du modèle géométrique actuel à un maillage tétraédrique, il faudrait plutôt modifier le modèle actuel à l'aide d'algorithmes d'optimisation de maillage.

Une amélioration qui pourrait être apportée aux processus d'analyse utilisant l'un des différents modèles d'IA est la caractérisation de la distribution probabiliste des données ultrasonores. À chaque nouvel ajout de données, il est important de vérifier à l'aide d'outils statistiques, comme un test-t ou une analyse de la variance (ANOVA), que les distributions des données ultrasonores demeurent les mêmes. Lorsque les distributions des données changent, il est nécessaire de procéder à un réentraînement des différents modèles d'IA afin de s'assurer qu'ils possèdent une capacité de prédiction en concordance avec les données à prédire.

Afin d'augmenter la taille de la base de données utilisée pour entraîner les différents modèles d'IA, il serait possible de faire appel aux modèles génératifs (*Generative Adversarial Networks*). Ce type de modèle est en mesure de créer de nouvelles images de haute qualité qui appartiennent à la même distribution probabiliste. Toutefois, l'utilisation avec succès des modèles génératifs pour générer des images ultrasonores est très récente, ce qui est notamment dû par la complexité des signaux ultrasonores [220].

LISTE DE RÉFÉRENCES

- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097-1105, 2012.
- [3] K. Raza and N. K. Singh, "A Tour of Unsupervised Deep Learning for Medical Image Analysis," *arXiv preprint arXiv:1812.07715*, pp. 1-29, 2018.
- [4] F. Altaf, S. Islam, N. Akhtar, and N. K. Janjua, "Going Deep in Medical Image Analysis: Concepts, Methods, Challenges and Future Directions," *arXiv preprint arXiv:1902.05655*, 2019.
- [5] American Society for Nondestructive Testing, *Nondestructive Testing Handbook, Third Edition: Volume 7, Ultrasonic Testing (UT)*. United States of America: American Society for Nondestructive Testing, Inc., 2007, p. 588.
- [6] F. Mohamed and V. S. Chan, "A Survey on 3D Ultrasound Reconstruction Techniques," in *Machine Learning in Medicine and Biology*: IntechOpen, 2019.
- [7] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, and J. Rinta-Aho, "Augmented Ultrasonic Data for Machine Learning," *arXiv preprint arXiv:1903.11399*, 2019.
- [8] W. Nash, T. Drummond, and N. Birbilis, "A review of deep learning in the study of materials degradation," *npj Materials Degradation*, vol. 2, pp. 1-12, 2018.
- [9] M. P. Groover, *Fundamentals of modern manufacturing: materials processes, and systems*. John Wiley & Sons, 2007.
- [10] M. Meng, Y. J. Chua, E. Wouterson, and C. P. K. Ong, "Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks," *Neurocomputing*, vol. 257, pp. 128-135, 2017.
- [11] A. Annoni *et al.*, "Artificial Intelligence: How Knowledge Is Created, Transferred, and Used," ed: Elsevier AI Resource Center, 2018.
- [12] E. Ahmed *et al.*, "Deep learning advances on different 3D data representations: A survey," *arXiv preprint arXiv:1808.01462*, pp. 1-35, 2018.
- [13] N. Munir, H.-J. Kim, J. Park, S.-J. Song, and S.-S. Kang, "Convolutional neural network for ultrasonic weldment flaw classification in noisy conditions," *Ultrasonics*, vol. 94, pp. 74-81, 2019.
- [14] R. Draï, M. Khelil, and A. Benchaala, "Time frequency and wavelet transform applied to selected problems in ultrasonics NDE," *NDT & e International*, vol. 35, no. 8, pp. 567-572, 2002.
- [15] F. Cruz, E. Simas Filho, M. Albuquerque, I. Silva, C. Farias, and L. Gouvêa, "Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing," *Ultrasonics*, vol. 73, pp. 1-8, 2017.
- [16] J. Zapata, R. Vilar, and R. Ruiz, "An adaptive-network-based fuzzy inference system for classification of welding defects," *NDT & e International*, vol. 43, no. 3, pp. 191-199, 2010.
- [17] Y. Chen, H.-W. Ma, and G.-M. Zhang, "A support vector machine approach for classification of welding defects from ultrasonic signals," *Nondestructive Testing and Evaluation*, vol. 29, no. 3, pp. 243-254, 2014.
- [18] W. Hou, Y. Wei, Y. Jin, and C. Zhu, "Deep features based on a DCNN model for classifying imbalanced weld flaw types," *Measurement*, vol. 131, pp. 482-489, 2019.
- [19] W. Hou, Y. Wei, J. Guo, Y. Jin, and C. a. Zhu, "Automatic detection of welding defects using deep neural network," in *Journal of Physics: Conference Series*, 2018, vol. 933, no. 1: IOP Publishing, p. 012006.
- [20] A. Kaur and P. Garg, "Feature Extraction and Classification of Welding Defects using Artificial Neural Network," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 5, pp. 886-889, 2017.
- [21] J. Liu, G. Xu, L. Ren, Z. Qian, and L. Ren, "Defect intelligent identification in resistance spot welding ultrasonic detection based on wavelet packet and neural network," *The International Journal of Advanced Manufacturing Technology*, vol. 90, no. 9-12, pp. 2581-2588, 2017.

- [22] N. Munir, H.-J. Kim, S.-J. Song, and S.-S. Kang, "Investigation of deep neural network with drop out for ultrasonic flaw classification in weldments," *Journal of Mechanical Science and Technology*, vol. 32, no. 7, pp. 3073-3080, 2018.
- [23] American Institute of Nondestructive Testing. "Addressing The Nde Technician Shortage." <https://www.trainingndt.com/ndt-news/entry/addressing-the-nde-technician-shortage> (accessed 2018-08-13, 2018).
- [24] Quality Magazine. "Gap between Supply, Demand of Well-Trained Technicians Drives NDT Training Services." <https://www.qualitymag.com/articles/91994-gap-between-supply-demand-of-well-trained-technicians-drives-ndt-training-services-market> (accessed 2014-07-08, 2014).
- [25] C. Metalworking. "Canada faces shortage of NDT inspectors." <https://www.canadianmetalworking.com/article/measurement/canada-faces-shortage-of-ndt-inspectors> (accessed 2013-12-09, 2013).
- [26] R. Young, K. Newton, T. Dunhill, and C. Huggins, "A landscape for the future of NDT in the UK economy," *Brochure, Materials KTN, Knowledge Transfer Networks, Materials KTN, NPL Product Verification Programme, Materials Knowledge Transfer Network*, 2014.
- [27] Z. Iscan, M. N. Kurnaz, Z. Dokur, and T. Olmez, "Ultrasound image segmentation by using wavelet transform and self-organizing neural network," *Neural Information Processing-Letters and Reviews*, vol. 10, no. 8-9, pp. 183-191, 2006.
- [28] Y. Guo *et al.*, "Fully Convolutional Neural Network With GRU for 3D Braided Composite Material Flaw Detection," *IEEE Access*, vol. 7, pp. 151180-151188, 2019.
- [29] F. Margrave, K. Rigas, D. A. Bradley, and P. Barrowcliffe, "The use of neural networks in ultrasonic flaw detection," *Measurement*, vol. 25, no. 2, pp. 143-154, 1999.
- [30] S. Song and L. Schmerr, "Ultrasonic flaw classification in weldments using probabilistic neural networks," *Journal of Nondestructive Evaluation*, vol. 11, no. 2, pp. 69-77, 1992.
- [31] N. Nacereddine, A. B. Goumeidane, and D. Ziou, "Unsupervised weld defect classification in radiographic images using multivariate generalized Gaussian mixture model with exact computation of mean and shape parameters," *Computers in Industry*, vol. 108, pp. 132-149, 2019.
- [32] H. Hu, G. Peng, X. Wang, and Z. Zhou, "Weld defect classification using 1-D LBP feature extraction of ultrasonic signals," *Nondestructive Testing and Evaluation*, vol. 33, no. 1, pp. 92-108, 2018.
- [33] M. S. H. Nizam, S. Marizan, and S. A. Zaki, "Vision based identification and classification of weld defects in welding environments: A review," *Indian Journal of Science and Technology*, vol. 9, no. 20, pp. 1-5, 2016.
- [34] V. R. Rathod and R. Anand, "A comparative study of different segmentation techniques for detection of flaws in NDE weld images," *Journal of Nondestructive Evaluation*, vol. 31, no. 1, pp. 1-16, 2012.
- [35] S. Seyedtabaai, "Performance evaluation of neural network based pulse-echo weld defect classifiers," *Measurement Science Review*, vol. 12, no. 5, pp. 168-174, 2012.
- [36] G. Mathers, *The welding of aluminium and its alloys*. Woodhead publishing, 2002.
- [37] M. Deriche, "Defect classification in NDT applications using time frequency features LDA and a KNN classifier," *Australian Journal of Basic and Applied Sciences*, vol. 8, no. 7, pp. 485-498, 2014.
- [38] L. Liu, J. Zhang, X. Fu, L. Liu, and Q. Huang, "Unsupervised segmentation and elm for fabric defect image classification," *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 12421-12449, 2019.
- [39] R. Ye, C.-S. Pan, M. Chang, and Q. Yu, "Intelligent defect classification system based on deep learning," *Advances in Mechanical Engineering*, vol. 10, no. 3, pp. 1-7, 2018.
- [40] C. T. Cenate, B. S. Rani, R. Ramadevi, D. Sangeetha, and B. Venkatraman, "Optimization of the Cascade Feed Forward Back Propagation network for defect classification in ultrasonic images," *Russian Journal of Nondestructive Testing*, vol. 52, no. 10, pp. 557-568, 2016.
- [41] S. Sambath, P. Nagaraj, and N. Selvakumar, "Automatic defect classification in ultrasonic NDT using artificial intelligence," *Journal of nondestructive evaluation*, vol. 30, no. 1, pp. 20-28, 2011.

- [42] I. Virkkunen and T. Koskinen, "Flaw Detection in Ultrasonic Data Using Deep Learning," in *Baltica XI 2019: International Conference on Life Management and Maintenance for Power Plants*, 2019: VTT Technical Research Centre of Finland.
- [43] P. Napoletano, F. Piccoli, and R. Schettini, "Anomaly detection in nanofibrous materials by cnn-based self-similarity," *Sensors*, vol. 18, no. 1, p. 209, 2018.
- [44] B. Chen, Z. Fang, Y. Xia, L. Zhang, Y. Huang, and L. Wang, "Accurate defect detection via sparsity reconstruction for weld radiographs," *NDT & E International*, vol. 94, pp. 62-69, 2018.
- [45] B. Karthikeyan, V. Vaithyanathan, B. Venkatraman, and M. Menaka, "Analysis of image segmentation for radiographic images," *Indian Journal of Science and Technology*, vol. 5, no. 11, pp. 3660-3664, 2012.
- [46] M. Souza, R. Almeida, J. Rebello, and S. Soares, "Detection of lack of fusion weld defects by radiography," in *AIP Conference Proceedings*, 2009, vol. 1096, no. 1: AIP, pp. 1174-1181.
- [47] J. Park *et al.*, "MRPC eddy current flaw classification in tubes using deep neural networks," *Nuclear Engineering and Technology*, 2019.
- [48] S. Liu *et al.*, "Deep Learning in Medical Ultrasound Analysis: A Review," *Engineering*, 2019.
- [49] H. Li, T. Zhao, N. Li, Q. Cai, and J. Du, "Feature Matching of Multi-View 3D Models Based on Hash Binary Encoding," *Neural Network World*, vol. 27, no. 1, pp. 95-105, 2017.
- [50] W. T. Koh, "Leeden National Oxygen and CEO fined over fatal blast at Tanjong Kling Road," ed. Singapore: Yahoo News Singapore, 2021.
- [51] F. Naumann and M. Herschel, "An introduction to duplicate detection," *Synthesis Lectures on Data Management*, vol. 2, pp. 1-87, 2010.
- [52] A. Larbanet, "Application de l'algorithme de Max-hashing pour le référencement de fichiers vidéo et la détection de contenus et de flux connus à haute vitesse sur GPU," École Polytechnique de Montréal, 2014.
- [53] D. Rachmawati, J. Tarigan, and A. Ginting, "A comparative study of Message Digest 5 (MD5) and SHA256 algorithm," in *Journal of Physics: Conference Series*, 2018, vol. 978, no. 1: IOP Publishing, p. 012116.
- [54] A. Kunhu, H. Al-Ahmad, and F. Taher, "Medical images protection and authentication using hybrid DWT-DCT and SHA256-MD5 hash functions," in *2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2017: IEEE, pp. 397-400.
- [55] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
- [56] S. Simpson, "Fault identification in gas metal arc welding with signature images," *Science and Technology of welding and Joining*, vol. 13, no. 1, pp. 87-96, 2008.
- [57] S. Simpson, "Signature images for arc welding fault detection," *Science and Technology of Welding and Joining*, vol. 12, no. 6, pp. 481-486, 2007.
- [58] A. Sumesh, K. Rameshkumar, K. Mohandas, and R. S. Babu, "Use of machine learning algorithms for weld quality monitoring using acoustic signature," *Procedia Computer Science*, vol. 50, pp. 316-322, 2015.
- [59] L. Morra and F. Lamberti, "Benchmarking unsupervised near-duplicate image detection," *Expert Systems with Applications*, vol. 135, pp. 313-326, 2019.
- [60] X. Liu and L. Xu, "Detecting Approximately Duplicate Records in Database," in *Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012*, 2013: Springer, pp. 325-332.
- [61] M. Aumüller, E. Bernhardsson, and A. Faithfull, "ANN-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms," in *International Conference on Similarity Search and Applications*, 2017: Springer, pp. 34-49.
- [62] H. Xu, J. Wang, Z. Li, G. Zeng, S. Li, and N. Yu, "Complementary hashing for approximate nearest neighbor search," in *2011 International Conference on Computer Vision*, 2011: IEEE, pp. 1631-1638.
- [63] Y. Ke, R. Sukthankar, L. Huston, Y. Ke, and R. Sukthankar, "Efficient near-duplicate detection and sub-image retrieval," in *ACM International Conference on Multimedia*, 2004, vol. 4, no. 1, p. 5.
- [64] L. Paulevé, H. Jégou, and L. Amsaleg, "Locality sensitive hashing: A comparison of hash function types and querying mechanisms," *Pattern Recognition Letters*, vol. 31, pp. 1348-1358, 2010.

- [65] L. Pang, Y. Lan, J. Guo, J. Xu, J. Xu, and X. Cheng, "Deeprank: A new deep architecture for relevance ranking in information retrieval," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017: ACM, pp. 257-266.
- [66] J. Wang *et al.*, "Learning fine-grained image similarity with deep ranking," in *Conference on Computer Vision and Pattern Recognition*, 2014: IEEE, pp. 1386-1393.
- [67] R. Cohen, "Signal denoising using wavelets," *Project Report, Department of Electrical Engineering Technion, Israel Institute of Technology, Haifa*, 2012.
- [68] C. Liner, "An overview of wavelet transform concepts and applications," *University of Houston*, pp. 1-17, 2010.
- [69] G. A. Guarneri, F. N. Junior, and L. de Arruda, "Weld discontinuities classification using principal component analysis and support vector machine," *XI Simpósio Brasileiro de Automação Inteligente*, pp. 2358-4483, 2013.
- [70] M. R. Saybani, T. Y. Wah, A. Amini, S. R. A. S. Yazdi, and A. Lahsasna, "Applications of support vector machines in oil refineries: A survey," *International Journal of Physical Sciences*, vol. 6, no. 27, pp. 6295-6302, 2011.
- [71] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417-420, 2016.
- [72] B. Li, K. Xu, D. Feng, H. Mi, H. Wang, and J. Zhu, "Denoising convolutional autoencoder based B-mode ultrasound tongue image feature extraction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019: IEEE, pp. 7130-7134.
- [73] G. A. Guarneri, F. N. Junior, and L. V. R. de Arruda, "Comparative evaluation of artificial neural networks models to classify weld flaws using pulse-echo ultrasonic signals," in *22nd International Congress of Mechanical Engineering (COBEM 2013) November*, 2013, pp. 3-7.
- [74] J. Ye, S. Ito, and N. Toyama, "Computerized Ultrasonic Imaging Inspection: From Shallow to Deep Learning," *Sensors*, vol. 18, no. 11, p. 3820, 2018.
- [75] K. Virupakshappa and E. Oruklu, "Multi-Class Classification of Defect Types in Ultrasonic NDT Signals with Convolutional Neural Networks," in *2019 IEEE International Ultrasonics Symposium (IUS)*, 2019: IEEE, pp. 1647-1650.
- [76] W. Hou, D. Zhang, Y. Wei, J. Guo, and X. Zhang, "Review on Computer Aided Weld Defect Detection from Radiography Images," *Applied Sciences*, vol. 10, pp. 1878-1894, 2020.
- [77] H. Zhu, W. Ge, and Z. Liu, "Deep Learning-Based Classification of Weld Surface Defects," *Applied Sciences*, vol. 9, no. 16, p. 3312, 2019.
- [78] H. Jiang *et al.*, "Convolution neural network model with improved pooling strategy and feature selection for weld defect recognition," *Welding in the World*, vol. 64, no. 4, pp. 731-744, 2021.
- [79] M. M. Naqiuddin, M. S. Leong, L. Hee, and M. Azrieasrie, "Ultrasonic signal processing techniques for Pipeline: A review," in *MATEC Web of Conferences*, 2019, vol. 255: EDP Sciences, p. 06006.
- [80] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 2016: IEEE, pp. 241-246.
- [81] I. S. Souza, M. C. Albuquerque, E. F. de SIMAS FILHO, and C. T. FARIAS, "Signal processing techniques for ultrasound automatic identification of flaws in steel welded joints—a comparative analysis," in *18th World Conference on Nondestructive Testing*, 2012, pp. 16-20.
- [82] L. Bai, A. Velichko, and B. W. Drinkwater, "Characterization of defects using ultrasonic arrays: a dynamic classifier approach," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 62, no. 12, pp. 2146-2160, 2015.
- [83] S. Gai, B. Zhang, C. Yang, and L. Yu, "Speckle noise reduction in medical ultrasound image using monogenic wavelet and Laplace mixture distribution," *Digital Signal Processing*, vol. 72, pp. 192-207, 2018.
- [84] R. Naga, S. Chandralingam, T. Anjaneyulu, and K. Satyanarayana, "Denoising EOG signal using stationary wavelet transform," *Measurement Science Review*, vol. 12, no. 2, pp. 46-51, 2012.

- [85] N. Munir, J. Park, H.-J. Kim, S.-J. Song, and S.-S. Kang, "Performance enhancement of convolutional neural network for ultrasonic flaw classification by adopting autoencoder," *NDT & E International*, p. 102218, 2020.
- [86] L. Posilović, D. Medak, M. Subašić, T. Petković, M. Budimir, and S. Lončarić, "Flaw Detection from Ultrasonic Images using YOLO and SSD," in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2019: IEEE, pp. 163-168.
- [87] S. Shi, X. Wang, and H. Li, "Pointnet: 3d object proposal generation and detection from point cloud," *arXiv preprint arXiv:1812.04244*, 2018.
- [88] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.
- [89] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016: Springer, pp. 21-37.
- [90] R. Blondeau, *Metallurgy and mechanics of welding: processes and industrial applications*. John Wiley & Sons, 2013.
- [91] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2088-2096.
- [92] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 72, 2017.
- [93] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652-660.
- [94] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders," *arXiv preprint arXiv:1807.02011*, pp. 1-8, 2018.
- [95] S. Youkachen, M. Ruchanurucks, T. Phatrapomnant, and H. Kaneko, "Defect Segmentation of Hot-rolled Steel Strip Surface by using Convolutional Auto-Encoder and Conventional Image processing," in *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, 2019: IEEE, pp. 1-5.
- [96] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015: Springer, pp. 234-241.
- [97] J. Yang, M. Faraji, and A. Basu, "Robust segmentation of arterial walls in intravascular ultrasound images using Dual Path U-Net," *Ultrasonics*, vol. 96, pp. 24-33, 2019.
- [98] F. Isensee *et al.*, "nnu-net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, pp. 1-11, 2018.
- [99] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*, 2016: Springer, pp. 424-432.
- [100] A. Kaku *et al.*, "DARTS: DenseUnet-based Automatic Rapid Tool for brain Segmentation," *arXiv preprint arXiv:1911.05567*, 2019.
- [101] P. Coupé *et al.*, "AssemblyNet: A Novel Deep Decision-Making Process for Whole Brain MRI Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019: Springer, pp. 466-474.
- [102] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D vision (3DV)*, 2016: IEEE, pp. 565-571.
- [103] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026-1034.
- [104] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected

- CRFS," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 2018.
- [105] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.
 - [106] D. Mishra, S. Chaudhury, M. Sarkar, S. Manohar, and A. S. Soin, "Segmentation of vascular regions in ultrasound images: A deep learning approach," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018: IEEE, pp. 1-5.
 - [107] O. Oktay *et al.*, "Anatomically constrained neural networks (ACNNs): application to cardiac image enhancement and segmentation," *IEEE transactions on medical imaging*, vol. 37, no. 2, pp. 384-395, 2017.
 - [108] M. Vakalopoulou *et al.*, "AtlasNet: multi-atlas non-linear deep networks for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018: Springer, pp. 658-666.
 - [109] Z. Mirikharaji and G. Hamarneh, "Star shape prior in fully convolutional networks for skin lesion segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018: Springer, pp. 737-745.
 - [110] C. Liu, F. Liu, L. Wang, L. Ma, and Z.-M. Lu, "SEGMENTATION OF NERVE ON ULTRASOUND IMAGES USING DEEP ADVERSARIAL NETWORK," *INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING INFORMATION AND CONTROL*, vol. 14, no. 1, pp. 53-64, 2018.
 - [111] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672-2680.
 - [112] S. Kazeminia *et al.*, "GANs for medical image analysis," *arXiv preprint arXiv:1809.06222*, 2018.
 - [113] P. Moeskops, M. Veta, M. W. Lafarge, K. A. Eppenhof, and J. P. Pluim, "Adversarial training and dilated convolutions for brain MRI segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*: Springer, 2017, pp. 56-64.
 - [114] J. Tan, L. Jing, Y. Huo, Y. Tian, and O. Akin, "LGAN: Lung Segmentation in CT Scans Using Generative Adversarial Network," *arXiv preprint arXiv:1901.03473*, 2019.
 - [115] V. K. Singh *et al.*, "An Efficient Solution for Breast Tumor Segmentation and Classification in Ultrasound Images Using Deep Adversarial Learning," *arXiv preprint arXiv:1907.00887*, 2019.
 - [116] D. Yang *et al.*, "Automatic liver segmentation using an adversarial image-to-image network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2017: Springer, pp. 507-515.
 - [117] S. Kohl *et al.*, "Adversarial networks for the detection of aggressive prostate cancer," *arXiv preprint arXiv:1702.08014*, 2017.
 - [118] E. Kozegar, M. Soryani, H. Behnam, M. Salamati, and T. Tan, "Computer aided detection in automated 3-D breast ultrasound images: a survey," *Artificial Intelligence Review*, pp. 1-23, 2019.
 - [119] H. R. Torres, S. Queiros, P. Morais, B. Oliveira, J. C. Fonseca, and J. L. Vilaca, "Kidney segmentation in ultrasound, magnetic resonance and computed tomography images: A systematic review," *Computer methods and programs in biomedicine*, vol. 157, pp. 49-67, 2018.
 - [120] M. Xian *et al.*, *A benchmark for breast ultrasound image segmentation (BUSIS)*. Infinite Study, 2018.
 - [121] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI," *Journal of Magnetic Resonance Imaging*, vol. 49, no. 4, pp. 939-954, 2019.
 - [122] F. Canè *et al.*, "From 4D Medical Images (CT, MRI, and Ultrasound) to 4D Structured Mesh Models of the Left Ventricular Endocardium for Patient-Specific Simulations," *BioMed research international*, vol. 2018, 2018.

- [123] C. Gnonnou and N. Smaoui, "Segmentation and 3D reconstruction of MRI images for breast cancer detection," in *International Image Processing, Applications and Systems Conference*, 2014: IEEE, pp. 1-6.
- [124] M. Stevanella *et al.*, "Mitral valve patient-specific finite element modeling from cardiac MRI: application to an annuloplasty procedure," *Cardiovascular Engineering and Technology*, vol. 2, no. 2, pp. 66-76, 2011.
- [125] F.-C. Ghesu *et al.*, "Multi-scale deep reinforcement learning for real-time 3D-landmark detection in CT scans," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 1, pp. 176-189, 2017.
- [126] J.-G. Lee *et al.*, "Deep learning in medical imaging: general overview," *Korean journal of radiology*, vol. 18, no. 4, pp. 570-584, 2017.
- [127] B. Sahiner *et al.*, "Deep learning in medical imaging and radiation therapy," *Medical physics*, vol. 46, no. 1, pp. e1-e36, 2019.
- [128] A. Maier, C. Syben, T. Lasser, and C. Riess, "A gentle introduction to deep learning in medical image processing," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 86-101, 2019.
- [129] Y. Xu, Y. Wang, J. Yuan, Q. Cheng, X. Wang, and P. L. Carson, "Medical breast ultrasound image segmentation by machine learning," *Ultrasonics*, vol. 91, pp. 1-9, 2019.
- [130] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.
- [131] Y. Jaafra, J. L. Laurent, A. Deruyver, and M. S. Naceur, "A review of meta-reinforcement learning for deep neural networks architecture search," *arXiv preprint arXiv:1812.07995*, 2018.
- [132] A. Alansary *et al.*, "Evaluating reinforcement learning agents for anatomical landmark detection," *Medical image analysis*, vol. 53, pp. 156-164, 2019.
- [133] S. Saikia, "Building a Convolutional Autoencoder with Keras using Conv2DTranspose," in *PNG* vol. 700x326, C. Auto-Encoder, Ed., ed. medium.com, 2019.
- [134] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, vol. 2: IEEE, pp. 1398-1402.
- [135] N. Tuluptceva, B. Bakker, I. Fedulova, H. Schulz, and D. V. Dylov, "Anomaly detection with deep perceptual autoencoders," *arXiv preprint arXiv:2006.13265*, 2020.
- [136] J. J. Cerrolaza *et al.*, "Fetal Skull Reconstruction via Deep Convolutional Autoencoders," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018: IEEE, pp. 887-890.
- [137] T. Joyce, A. Chartsias, and S. A. Tsiftaris, "Deep multi-class segmentation without ground-truth labels," 2018.
- [138] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks," *Medical image analysis*, vol. 54, pp. 30-44, 2019.
- [139] G. Pang, C. Shen, L. Cao, and A. v. d. Hengel, "Deep Learning for Anomaly Detection: A Review," *arXiv preprint arXiv:2007.02500*, 2020.
- [140] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *International conference on machine learning*, 2017: PMLR, pp. 214-223.
- [141] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep autoencoding models for unsupervised anomaly segmentation in brain mr images," in *International MICCAI Brainlesion Workshop*, 2018: Springer, pp. 161-169.
- [142] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019: IEEE, pp. 1-8.
- [143] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient gan-based anomaly detection," *arXiv preprint arXiv:1802.06222*, 2018.
- [144] M. A. Degel, N. Navab, and S. Albarqouni, "Domain and geometry agnostic CNNs for left atrium segmentation in 3D ultrasound," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018: Springer, pp. 630-637.

- [145] H. E. Atlason, A. Love, S. Sigurdsson, V. Gudnason, and L. M. Ellingsen, "Unsupervised brain lesion segmentation from MRI using a convolutional autoencoder," in *Medical Imaging 2019: Image Processing*, 2019, vol. 10949: International Society for Optics and Photonics, p. 109491H.
- [146] A. Tuysuzoglu, J. Tan, K. Eissa, A. P. Kiraly, M. Diallo, and A. Kamen, "Deep adversarial context-aware landmark detection for ultrasound imaging," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018: Springer, pp. 151-158.
- [147] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Asian Conference on Computer Vision*, 2018: Springer, pp. 622-637.
- [148] H. S. Vu, D. Ueta, K. Hashimoto, K. Maeno, S. Pranata, and S. M. Shen, "Anomaly detection with adversarial dual autoencoders," *arXiv preprint arXiv:1902.06924*, 2019.
- [149] A. K. Mondal, J. Dolz, and C. Desrosiers, "Few-shot 3d multi-modal medical image segmentation using generative adversarial learning," *arXiv preprint arXiv:1810.12241*, 2018.
- [150] Y. Huo *et al.*, "3D whole brain segmentation using spatially localized atlas network tiles," *NeuroImage*, vol. 194, pp. 105-119, 2019.
- [151] M. Singh Minhas and J. Zelek, "AnoNet: Weakly Supervised Anomaly Detection in Textured Surfaces," *arXiv*, p. arXiv: 1911.10608, 2019.
- [152] D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. Song, "A Benchmark for Anomaly Segmentation," *arXiv preprint arXiv:1911.11132*, 2019.
- [153] S. K. Lim, Y. Loo, N.-T. Tran, N.-M. Cheung, G. Roig, and Y. Elovici, "DOPING: Generative data augmentation for unsupervised anomaly detection with GAN," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018: IEEE, pp. 1122-1127.
- [154] C. P. Ngo, A. A. Winarto, C. K. K. Li, S. Park, F. Akram, and H. K. Lee, "Fence GAN: towards better anomaly detection," *arXiv preprint arXiv:1904.01209*, 2019.
- [155] S. Li, C. Zhang, and X. He, "Shape-aware semi-supervised 3d semantic segmentation for medical images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2020: Springer, pp. 552-561.
- [156] A. Alhazmi and S. Semwal, "3D Volume Visualization in Medical Application," in *Proceedings of the International Conference on Modeling, Simulation and Visualization Methods (MSV)*, 2018: The Steering Committee of The World Congress in Computer Science, Computer ..., pp. 58-64.
- [157] T. Kim and J. Park, "Analyzing 3D cell data of optical diffraction tomography through volume rendering," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, 2018: IEEE, pp. 1-4.
- [158] I. Mehmood *et al.*, "An efficient computerized decision support system for the analysis and 3D visualization of brain tumor," *Multimedia Tools and Applications*, pp. 1-26, 2018.
- [159] P. Bizopoulos and D. Koutsouris, "Deep Learning in Cardiology," *IEEE reviews in biomedical engineering*, vol. 12, pp. 168-193, 2018.
- [160] R. P. Singh, S. Gupta, and U. R. Acharya, "Segmentation of prostate contours for automated diagnosis using ultrasound images: A survey," *Journal of computational science*, vol. 21, pp. 223-231, 2017.
- [161] B. R. De Araújo, D. S. Lopes, P. Jepp, J. A. Jorge, and B. Wyvill, "A survey on implicit surface polygonization," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, p. 60, 2015.
- [162] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, pp. 1-13, 2013.
- [163] Z. Wu *et al.*, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912-1920.
- [164] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, "3D Point-Capsule Networks," *arXiv preprint arXiv:1812.10775*, 2018.
- [165] Y. Liao, S. Donne, and A. Geiger, "Deep marching cubes: Learning explicit surface representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2916-2925.
- [166] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "Meshcnn: a network with an edge," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1-12, 2019.

- [167] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "MeshNet: Mesh Neural Network for 3D Shape Representation," *arXiv preprint arXiv:1811.11424*, 2018.
- [168] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson, and C. Fookes, "Image2mesh: A learning framework for single image 3d reconstruction," in *Asian Conference on Computer Vision*, 2018: Springer, pp. 365-381.
- [169] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3907-3916.
- [170] G. Gkioxari, J. Malik, and J. Johnson, "Mesh r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9785-9795.
- [171] A. Dai and M. Nießner, "Scan2Mesh: From Unstructured Range Scans to 3D Meshes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5574-5583.
- [172] R. Hanocka, G. Metzger, R. Giryas, and D. Cohen-Or, "Point2Mesh: A Self-Prior for Deformable Meshes," *arXiv preprint arXiv:2005.11084*, 2020.
- [173] P. Erler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Wimmer, "Points2surf learning implicit surfaces from point clouds," in *European Conference on Computer Vision*, 2020: Springer, pp. 108-124.
- [174] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3d surface generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 216-224.
- [175] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165-174.
- [176] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo, "Deep geometric prior for surface reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10130-10139.
- [177] B. Ummenhofer and V. Koltun, "Adaptive Surface Reconstruction With Multiscale Convolutional Kernels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5651-5660.
- [178] X. Wei, Z. Chen, Y. Fu, Z. Cui, and Y. Zhang, "Deep Hybrid Self-Prior for Full 3D Mesh Generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5805-5814.
- [179] W. Yifan, S. Wu, C. Oztireli, and O. Sorkine-Hornung, "Iso-Points: Optimizing Neural Implicit Surfaces with Hybrid Representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 374-383.
- [180] Z. Wang *et al.*, "Neural-IMLS: Learning Implicit Moving Least-Squares for Surface Reconstruction from Unoriented Point clouds," *arXiv preprint arXiv:2109.04398*, 2021.
- [181] B. Ma, Z. Han, Y.-S. Liu, and M. Zwicker, "Neural-Pull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces," *arXiv preprint arXiv:2011.13495*, 2020.
- [182] S. Song, Z. Cui, and R. Qin, "Vis2Mesh: Efficient Mesh Reconstruction from Unstructured Point Clouds of Large Scenes with Learned Virtual View Visibility," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6514-6524.
- [183] J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, "SA-ConvONet: Sign-Agnostic Optimization of Convolutional Occupancy Networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6504-6513.
- [184] W. Zhao, J. Lei, Y. Wen, J. Zhang, and K. Jia, "Sign-Agnostic Implicit Learning of Surface Self-Similarities for Shape Modeling and Reconstruction from Raw Point Clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10256-10265.
- [185] T. Pahlberg, E. Johansson, O. Hagman, and M. Thurley, "Wood fingerprint recognition using knot neighborhood K-plet descriptors," *Wood science and technology*, vol. 49, no. 1, pp. 7-20, 2015.

- [186] A. Andoni, P. Indyk, and I. Razenshteyn, "Approximate nearest neighbor search in high dimensions," in *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, 2018: World Scientific, pp. 3287-3318.
- [187] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509-517, 1975.
- [188] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *Vldb*, 1997, vol. 97: Citeseer, pp. 426-435.
- [189] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 4, pp. 824-836, 2018.
- [190] J. Yang, X. Lu, and W. Chen, "A Robust Scheme for 3D Point Cloud Copy Detection," *arXiv preprint arXiv:2110.00972*, 2021.
- [191] D. Shukla and M. Sharma, "Watermarking schemes for copy protection: A survey," *International Journal of Computer Science and Engineering Survey*, vol. 3, no. 1, p. 65, 2012.
- [192] N. Medimegh, S. Belaid, and N. Werghi, "A survey of the 3D triangular mesh watermarking techniques," *International Journal of Multimedia*, vol. 1, no. 1, 2015.
- [193] J. Wang, T. Zhang, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, pp. 769-790, 2017.
- [194] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600-612, 2004.
- [195] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE signal processing letters*, vol. 9, pp. 81-84, 2002.
- [196] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019: IEEE, pp. 1480-1484.
- [197] N. Nuechterlein and S. Mehta, "3D-ESPNet with pyramidal refinement for volumetric brain tumor image segmentation," in *International MICCAI Brainlesion Workshop*, 2018: Springer, pp. 245-253.
- [198] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.
- [199] K. Jamieson and A. Talwalkar, "Non-stochastic best arm identification and hyperparameter optimization," in *Artificial Intelligence and Statistics*, 2016: PMLR, pp. 240-248.
- [200] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2682-2690.
- [201] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, pp. 1-13, 2017.
- [202] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.
- [203] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimedia Tools and Applications*, vol. 79, no. 19, pp. 12777-12815, 2020.
- [204] S. Santurkar, D. Tsipras, A. Ilyas, and A. Mądry, "How does batch normalization help optimization?," in *Proceedings of the 32nd international conference on neural information processing systems*, 2018, pp. 2488-2498.
- [205] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
- [206] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3-19.
- [207] S. Audet, "Unsupervised Learning for Object Recognition," 2006.
- [208] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1-48, 2019.

- [209] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018: IEEE, pp. 1542-1547.
- [210] R. Bènière, G. Subsol, G. Gesquière, F. Le Breton, and W. Puech, "A comprehensive process of reverse engineering from 3D meshes to CAD models," *Computer-Aided Design*, vol. 45, no. 11, pp. 1382-1393, 2013.
- [211] I. Marin, S. Gotovac, M. Russo, and D. Božić-Štulić, "The Effect of Latent Space Dimension on the Quality of Synthesized Human Face Images," *Journal of Communications Software and Systems*, vol. 17, no. 2, pp. 124-133, 2021.
- [212] S.-H. Lee and K.-R. Kwon, "Robust 3D mesh model hashing based on feature object," *Digital Signal Processing*, vol. 22, no. 5, pp. 744-759, 2012.
- [213] I. Haloui, J. S. Gupta, and V. Feuillard, "Anomaly detection with Wasserstein GAN," *arXiv preprint arXiv:1812.02463*, 2018.
- [214] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh: Measuring errors between surfaces using the hausdorff distance," in *Proceedings. IEEE international conference on multimedia and expo*, 2002, vol. 1: IEEE, pp. 705-708.
- [215] M. Guthe, P. Borodin, and R. Klein, "Fast and accurate Hausdorff distance calculation between meshes," 2005.
- [216] J. Ma *et al.*, "Loss odyssey in medical image segmentation," *Medical Image Analysis*, p. 102035, 2021.
- [217] J. Behley *et al.*, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9297-9307.
- [218] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.
- [219] E. Provencal and L. Laperrière, "Detection of exact and near duplicates in phased-array ultrasound weld scan," in *10th International Conference on Digital Enterprise Technology (DET 2021)*, Budapest, Hungary, Elsevier, Ed., 2021, pp. 1-6.
- [220] L. Posilović, D. Medak, M. Subašić, M. Budimir, and S. Lončarić, "Generating ultrasonic images indistinguishable from real images using Generative Adversarial Networks," *Ultrasonics*, vol. 119, p. 106610, 2022.
- [221] G. Zhang and H. Li, "Effectiveness of scaled exponentially-regularized linear units (SERLUs)," *arXiv preprint arXiv:1807.10117*, 2018.
- [222] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.
- [223] G. Bingham and R. Miikkulainen, "Discovering parametric activation functions," *Neural Networks*, 2022.
- [224] S. Cai, Y. Shu, G. Chen, B. C. Ooi, W. Wang, and M. Zhang, "Effective and efficient dropout for deep convolutional neural networks," *arXiv preprint arXiv:1904.03392*, 2019.
- [225] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.
- [226] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," *arXiv preprint arXiv:1702.03275*, 2017.
- [227] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [228] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.
- [229] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in neural information processing systems*, 1992, pp. 950-957.
- [230] L. Prechelt, "Early stopping-but when?," in *Neural Networks: Tricks of the trade*: Springer, 1998, pp. 55-69.
- [231] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017: IEEE, pp. 464-472.

- [232] Y. Wu *et al.*, "Demystifying learning rate policies for high accuracy training of deep neural networks," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019: IEEE, pp. 1971-1980.
- [233] F. Schneider, L. Balles, and P. Hennig, "DeepOBS: A deep learning optimizer benchmark suite," *arXiv preprint arXiv:1903.05499*, 2019.
- [234] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, "On empirical comparisons of optimizers for deep learning," *arXiv preprint arXiv:1910.05446*, 2019.
- [235] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400-407, 1951.
- [236] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *Ussr computational mathematics and mathematical physics*, vol. 4, no. 5, pp. 1-17, 1964.
- [237] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of machine learning research*, vol. 12, no. 7, 2011.
- [238] G. Hinton, N. Srivastava, and K. Swersky, "Rmsprop: Divide the gradient by a running average of its recent magnitude," *Neural networks for machine learning, Coursera lecture 6e*, p. 13, 2012.
- [239] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [240] T. Dozat, "Incorporating nesterov momentum into adam," 2016.
- [241] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
- [242] S. Jadon, "A survey of loss functions for semantic segmentation," *arXiv preprint arXiv:2006.14822*, 2020.
- [243] D. Zhou *et al.*, "Iou loss for 2D/3D object detection," in *2019 International Conference on 3D Vision (3DV)*, 2019: IEEE, pp. 85-94.
- [244] M. Yi-de, L. Qing, and Q. Zhi-Bai, "Automated image segmentation using improved PCNN model based on cross-entropy," in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, 2004: IEEE, pp. 743-746.
- [245] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.
- [246] J. Bertels, D. Robben, D. Vandermeulen, and P. Suetens, "Optimization with soft dice can lead to a volumetric bias," in *International MICCAI Brainlesion Workshop*, 2019: Springer, pp. 89-97.
- [247] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*: Springer, 2017, pp. 240-248.
- [248] S. R. Hashemi, S. S. M. Salehi, D. Erdogmus, S. P. Prabhu, S. K. Warfield, and A. Gholipour, "Asymmetric loss functions and deep densely-connected networks for highly-imbalanced medical image segmentation: Application to multiple sclerosis lesion detection," *IEEE Access*, vol. 7, pp. 1721-1735, 2018.
- [249] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4413-4421.

6 ANNEXE 1 : PARAMÈTRES IMPORTANTS DES RÉSEAUX DE NEURONES

6.1 Fonction d'activation

Les fonctions d'activation ont un grand impact sur les performances d'un RDN. En effet, certaines fonctions d'activation ont tendance à accentuer la présence de problèmes tels que le *vanishing gradient* alors que d'autres fonctions permettent d'éviter ce problème. C'est le cas notamment de *ReLU* qui transforme les valeurs négatives en valeur nulle [201]. *ReLU* est d'ailleurs l'une des fonctions d'activation les plus utilisées. Or, il existe plusieurs fonctions d'activations développées plus récemment qui sont potentiellement plus performantes. La Figure 6.1 illustre quelques fonctions d'activation : *sigmoïde*, *TanH*, *ReLU*, *LeakyReLU*, *Maxout* et *ELU*.

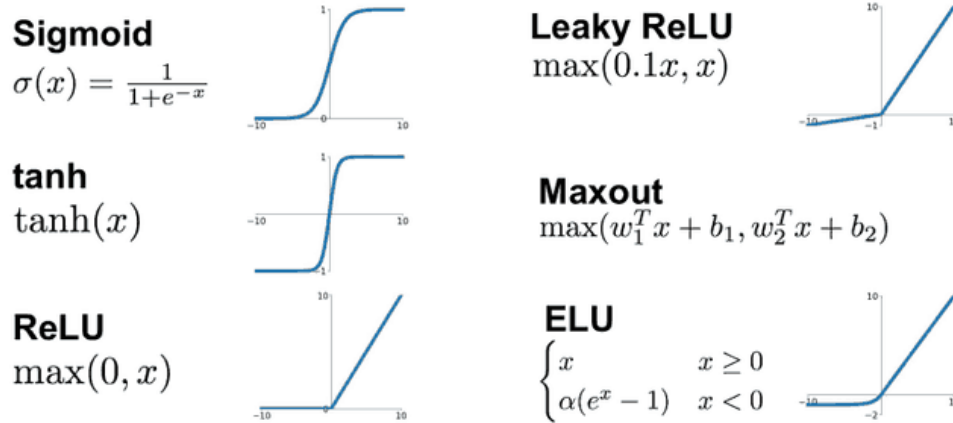


Figure 6.1 : Exemples visuels de quelques fonctions d'activation

Il est possible de constater que *sigmoïde* et *TanH* saturent à 1 alors que les fonctions d'activations de la famille de *ReLU* traite les valeurs négatives différemment des valeurs positives. Il existe une multitude de fonctions d'activations de la famille de *ReLU* : *ELU*, *LeakyReLU*, *PReLU*, *SeLU* et bien d'autres. Malgré leurs ressemblances, ces fonctions peuvent faire varier grandement les performances d'un modèle. Par exemple, *LeakyReLU* permet à une petite quantité d'informations du gradient de circuler lorsque la valeur de l'intrant est négative. *LeakyReLU* a été introduit pour empêcher *ReLU* de créer des neurones morts, c'est-à-dire qu'elles sont bloquées à toujours produire zéro. Au contraire, *ELU* contient un régime de saturation négatif pour contrôler la variance propagée. Ces deux fonctions d'activation très différentes ont des propriétés apparemment contradictoires. Pourtant, chacune s'est avérée plus efficace que *ReLU* dans diverses tâches.

L'importance du choix de fonction d'activation a été démontrée dans [201, 221]. Il a été démontré que le choix de la fonction d'activation a un effet significatif sur la performance du RDN. En effet,

simplement remplacer *ReLU* par *Swish* a permis d'améliorer les performances (classification top-1) par 0,90 % sur ImageNet. La Figure 6.2 illustre bien la différence de performance qu'occasionne le choix des fonctions d'activation.

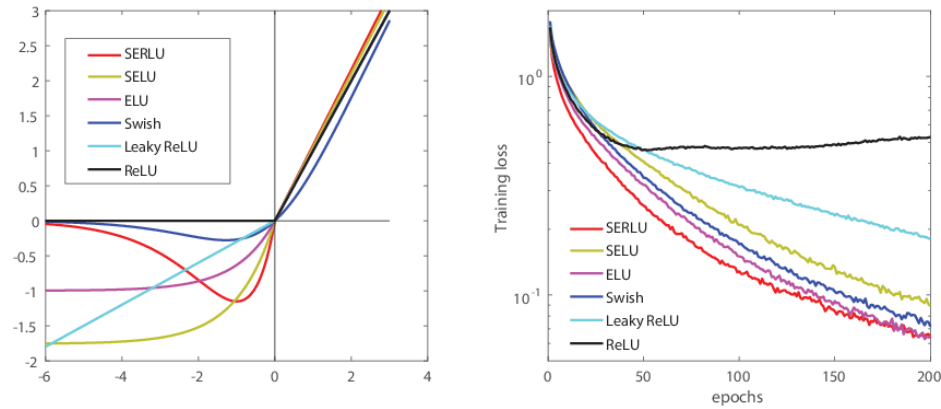


Figure 6.2 : Différentes fonctions d'activation et leur performance sur CIFAR10, tiré de [221]

GELU (*Gaussian Error Linear Units*) correspond à $x\Phi(x)$ où $\Phi(x)$ est la fonction standard gaussienne cumulative [222]. *GELU* pondère les entrées par leur percentile et non par leur signe, comme dans *ReLU*. Ceci engendre une non-linéarité bénéfique. Son comportement est similaire à *ReLU*, comme on peut constater à la Figure 6.3, en étant plus lisse.

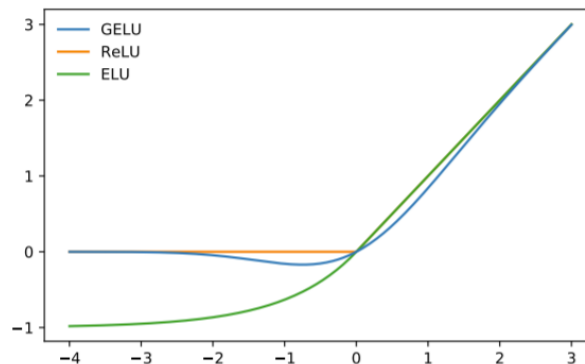


Figure 6.3 : *GELU*, *ReLU* et *ELU*, tiré de [222]

Les différences notables entre *GELU* et les autres sont que *GELU* est non-convexe, non monotone et non linéaire dans la partie positive. *GELU* présente des courbes en tout point. Cet avantage permet à *GELU* de mieux approximer des fonctions difficiles que *ReLU* ou *ELU*. En se basant sur diverses bases de données (*MNIST*, *cifar-10*), *GELU* performe mieux que *ReLU* et *ELU* dans les tâches de vision par ordinateur [222].

Néanmoins, il ne semble pas avoir de consensus dans la littérature à savoir quelle fonction d'activation est la meilleure, bien que *ReLU* soit actuellement considérée comme la fonction d'activation la plus performante et donc, la plus utilisée [201, 223]. Le choix optimal de la fonction d'activation semble grandement dépendre de la situation (modèle, objectif, données). Les performances des nouvelles fonctions d'activation étant inconsistantes et dépendantes de la tâche. Il existe des interactions complexes entre une fonction d'activation et d'autres choix de conception de RDN, ce qui ajoute à la difficulté de sélectionner une fonction d'activation appropriée pour une tâche donnée.

6.2 Méthode de régularisation

La régularisation est essentielle au bon fonctionnement des RDN profonds [224]. Plusieurs méthodes ont été proposées : *dropout*, normalisation (*batch*, couche, groupe, instance), *early stopping* et *weight decay*. Ces techniques ont pour principal objectif de limiter le surapprentissage. Parmi la liste, *dropout* est l'une des techniques les plus utilisées, simples et efficaces. Pendant l'entraînement, les couches *dropout* sont désactivées de manière aléatoire, selon une probabilité choisie, ce qui évite au réseau de concentrer les poids sur un nombre de neurones limité. En général, l'ajout de couche *dropout* a un effet bénéfique sur les performances des RDN [22]. L'efficacité a été démontrée à plusieurs occasions [225]. Or, il ne semble pas y avoir de consensus sur la valeur à choisir. Il est généralement convenu qu'une valeur de 0,5 offre les meilleurs résultats. Il est donc nécessaire de déterminer quelle valeur permet d'optimiser les performances pour la situation à l'étude.

Il existe plusieurs types de normalisation à l'intérieur de RDN : *batch*, instance, couche, groupe pour ne nommer que les principales. La normalisation a pour objectif d'aider l'entraînement de modèle en résolvant le problème de décalage interne de la covariance [205]. Ce problème survient quand la distribution des entrées de chaque couche change pendant l'entraînement, au fur et à mesure que les paramètres des couches précédentes changent. L'une des solutions à ce problème est la normalisation par *batch* (BN) de la préactivation, ce qui va comme suit :

$$BN(x) = \gamma \frac{x - \mu B}{\sigma B} + \beta \quad (6.1)$$

μB et σB sont la moyenne et l'écart-type de la *batch* alors que γ et β sont des paramètres appris. Les statistiques sont calculées pour chaque *batch*, canaux par canaux. Les études ont démontré que l'utilisation de la normalisation par *batch* permet d'accélérer l'entraînement des RDN. En revanche, lorsque le *batch size* est faible, l'utilisation de la normalisation par *batch* ne semble pas améliorer significativement les performances. Également, il existe une différence dans la façon de calculer la normalisation entre l'entraînement et l'inférence, ce qui peut occasionner des différences dans les

performances. En effet, à l'inférence, la normalisation par *batch* est calculée sur l'ensemble des données grâce à une moyenne et un écart-type mobile.

Une amélioration à la normalisation par *batch*, appelée *Batch ReNormalization*, est proposée [226]. La *Batch ReNormalization* introduit deux nouveaux paramètres r et d permettant de contrôler la moyenne et l'écart-type, ce qui a pour effet de réduire la différence importante quand le *batch size* est faible.

$$\begin{aligned} r &= \frac{\sigma_B}{\sigma} & d &= \frac{\mu_B - \mu}{\sigma} \\ BR(x) &= \frac{x - \mu_B}{\sigma_B} r + d \end{aligned} \quad (6.2)$$

Bien que la normalisation par *batch* soit couramment utilisée en apprentissage profond, il ne semble pas y avoir de consensus sur l'utilité de celle-ci. Il semble y avoir une certaine controverse sur ce que celle-ci (ou la *Batch ReNormalization*) résout réellement [204]. En comparant trois modèles (un sans la normalisation par *batch*, un avec normalisation et un avec normalisation bruitée), l'étude montre qu'il n'y a pas vraiment différence de performance. L'étude démontre toutefois que la normalisation par *batch* permet de raffiner la valeur de la perte et par conséquent le gradient. Finalement, les auteurs de cette étude démontrent que l'utilisation de régularisation l_1 , l_2 ou l_∞ permet d'obtenir des résultats similaires. La Figure 6.4 illustre bien ce phénomène.

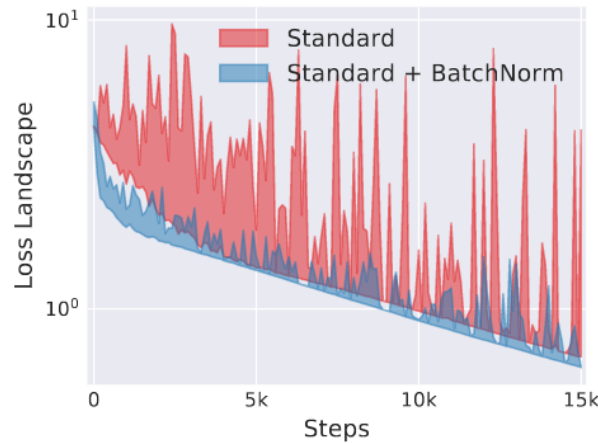


Figure 6.4 : Valeurs de perte avec ou sans la normalisation par *batch*, tiré de [204]

La normalisation par couche (*layer normalization*) normalise chaque intrant d'une *batch* indépendamment en utilisant tous les canaux [227]. Le but est d'avoir les mêmes performances, que le *batch size* soit faible ou important. Chaque canaux et chaque *batch* sont normalisés indépendamment [228]. Le but est de normaliser le contraste de l'intrant. Les auteurs considèrent

que le contraste des intrants est d'une importance capitale pour les performances du RDN. La normalisation par groupe normalise chaque intrant d'une *batch* indépendamment [206]. Ainsi, le modèle est indépendant du *batch size*. Selon les auteurs, les filtres de convolutions ont tendance à se regrouper par tâche similaire. Cette technique normalise les canaux par groupe défini arbitrairement. Chaque filtre d'un même groupe devrait se spécialiser dans la même tâche. La Figure 6.5 illustre plusieurs types de normalisation : *batch*, couche, instance et groupe. Chaque sous-graphique montre un tenseur de *feature map*, avec N comme axe du *batch size*, C comme axe des canaux et (H, W) comme axes spatiaux. Les pixels en bleu sont normalisés par la même moyenne et variance, calculée en agrégeant les valeurs de ces pixels.

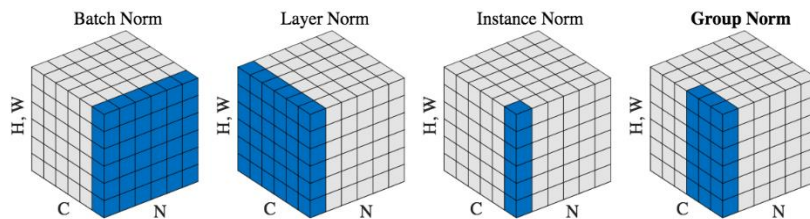


Figure 6.5 : Illustration des différents types de normalisation pour un exemple en 2D, tiré de [206]

La normalisation par *batch* est la technique de normalisation la plus utilisée et celle considérée comme étant la plus performante. Or, considérant le faible *batch size* utilisé dans cette étude, ce type de normalisation risque de ne pas être efficace. Il est donc nécessaire de tester d'autres techniques de normalisation.

Une étude empirique compare l'efficacité de deux techniques de régularisation : *dropout* et la normalisation par *batch* [203]. L'étude utilise deux bases de données connues, MNIST et Cifar10, pour comparer l'efficacité de différents modèles utilisant différentes combinaisons de paramètres des deux techniques. Les critères de performance considérés sont la durée d'entraînement, la précision de classification et le nombre de paramètres. Les conclusions de cette étude sont nombreuses et partagées. L'utilisation de *dropout* et de la normalisation par *batch* augmentent la durée d'entraînement, en particulier la normalisation par *batch*. Par exemple, la prédiction avec la normalisation par *batch* est 30 % plus longue et demande plus d'énergie. Son utilisation permet toutefois d'augmenter la précision sans avoir d'effets secondaires. L'ajout de *dropout* a l'effet inverse. Dans cette étude, il a été observé que *dropout* réduit la précision des modèles l'utilisant. Cette conclusion entre en conflit avec le résultat de plusieurs études. Il est donc suggéré de tester l'effet de *dropout* sur les performances des *CNNs* avant de l'adopter. En somme, l'utilisation de ces techniques de régularisation est recommandée pour les *CNNs*, mais demande de faire des

expériences pour fixer les meilleures conditions d'utilisation. La normalisation par batch est préférée à *dropout*.

Weight decay ou la régularisation l_2 est une technique de régularisation qui pénalise le modèle lorsqu'il gagne en complexité [1, 229]. En pratique, *weight decay* est un terme, d'une valeur faible, qui est multiplié aux poids des paramètres puis additionner à la valeur de la fonction de perte comme ceci :

$$\lambda \sum_i w_i^2 \quad (6.3)$$

Où w_i est le i^{e} poids dans le réseau et λ est un coefficient positif qui donne plus ou moins d'importance à la pénalité. La valeur de ce terme varie et il ne semble pas y avoir de valeur optimale. *Weight decay* est donc ajouté à *Hyperband* afin de trouver la valeur optimale de λ .

L'utilisation de *early stopping* permet d'éviter le surapprentissage en terminant l'entraînement lorsque le modèle cesse d'apprendre, pour un nombre d'*epochs* définies [230]. Cela permet également de réduire la durée de l'entraînement.

6.3 Taux d'apprentissage

Le *learning rate* (*LR*), taux d'apprentissage, indique à l'optimiseur quelle distance déplacer les poids dans la direction opposée au gradient pour une *mini-batch*. Si le *LR* est faible, alors l'apprentissage est plus fiable, mais l'optimisation prendra beaucoup de temps, car les étapes vers l'optimum local de la fonction de perte sont plus faibles [1]. Si le *LR* est élevé, alors l'entraînement peut ne pas converger. Les changements de poids peuvent être si importants que l'optimiseur ne réussit pas à trouver l'optimum local.

En règle générale, l'entraînement débute avec un *LR* plus important, considérant que les poids initiaux sont loin de l'optimalité, et la valeur du *LR* diminue au cours de l'entraînement afin de permettre un ajustement plus précis des poids. Il existe des méthodes qui ajustent la valeur du *LR* au cours de l'entraînement, ce qui permet d'améliorer les performances des modèles. Ces techniques incluent notamment *CosineAnnealingLR*, *CyclicLR*, *ReduceLROnPlateau* et *StepDecay*. Elles tentent d'optimiser le problème de sélection du *LR*. *CyclicLR* propose une toute nouvelle façon d'utiliser le *LR* au cours de l'entraînement [231]. *CyclicLR* est illustré à la Figure 6.6. On y voit bien que la valeur du *LR* est cyclique. Le cycle de 100 *epochs* comprend deux zones en forme de V où le *LR* croît jusqu'à atteindre la valeur maximale et ensuite descend pour atteindre la valeur minimale. Bien que le fonctionnement de *CyclicLR* semble être contre-intuitif, son efficacité a été prouvée en améliorant

les performances des RDN l'utilisant. Il est également possible de créer des cycles décroissants. *CosineAnnealingLR* est similaire à *CyclicLR*.

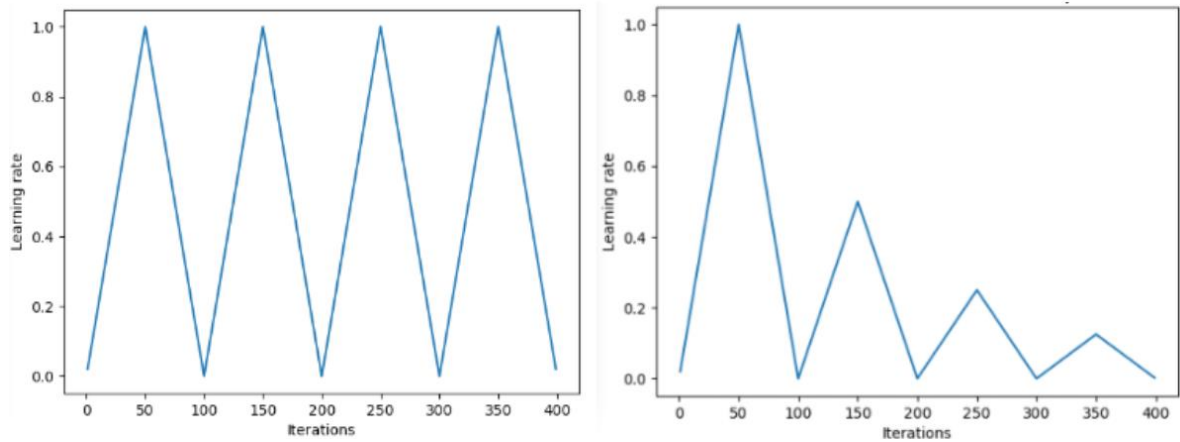


Figure 6.6 : Taux d'apprentissage de *CyclicLR* : *Triangular* et *Triangular2*, tiré de [231]

ReduceLROnPlateau réduit la valeur du *LR* lorsque le modèle cesse d'apprendre pour une période prédéfinie. Il s'agit d'une technique utile qui permet de réduire la difficulté de la recherche du bon *LR*. *StepDecay* réduit le *LR* graduellement en suivant une planification prédéterminée.

Il est à noter que chaque méthode possède à son tour une sous-catégorie d'hyperparamètres à optimiser pour obtenir des résultats optimaux tels que la durée des cycles, la patience (nombre d'itérations sans optimisation), la valeur maximale et minimale du *LR*, et bien d'autres.

Une étude ayant pour objectif de démystifier les bonnes pratiques du *LR* pour obtenir des résultats de haute précision a évalué plusieurs scénarios afin de trouver la procédure à suivre pour choisir le *LR* [232]. Les résultats montrent que le choix du *LR* dépend de l'ensemble de données et du modèle à entraîner. Il n'existe pas de valeur universellement meilleure.

6.4 Fonction d'optimisation

La fonction d'optimisation a un grand impact sur les résultats. Plusieurs études ont tenté de développer la meilleure fonction d'optimisation permettant d'atteindre les résultats optimaux [233, 234]. Les fonctions d'optimisation ont pour objectif de trouver un minimum local dans une fonction différentiable. Elles sont utilisées pour réduire la valeur de la perte et mettre à jour les poids dans la rétropropagation. La mise à jour des poids W_t par les fonctions d'optimisation correspond à ceci :

$$W_{t+1} = W_t - \alpha \left(\frac{\partial \text{Erreur}}{\partial W_t} \right) \quad (6.4)$$

Où a est le LR et W_{t+1} est les poids après l'ajustement. Il s'avère que les fonctions d'optimisation ne sont pas toutes équivalentes. *Stochastic Gradient Descent* (SGD) est l'une des fonctions d'optimisation la plus simple utilisée en IA [235]. SGD ne nécessite pas beaucoup de mémoire et est rapide. Toutefois, SGD met les poids à jour très régulièrement, ce qui peut entraîner des oscillations dans les résultats. La règle de mise à jour des paramètres θ à l'aide de $SGD(H_t, \eta_t)$ est la suivante :

$$\theta_{t+1} = \theta_t - \eta_t \nabla \ell(\theta_t) \quad (6.5)$$

Où H_t correspond à l'historique au temps t , ℓ représente la fonction de perte calculée sur l'ensemble des données à l'aide d'un RDN et η_t est le LR au moment t . SGD avec Momentum a pour objectif d'accélérer la convergence vers le minimum local [236]. Ceci est réalisé en combinant linéairement la direction du gradient avec une constante multiplication de la mise à jour précédente des paramètres. La règle de mise à jour de SGD avec $Momentum(H_t, \eta_t, \gamma)$ est :

$$\begin{aligned} v_0 &= 0 \\ v_{t+1} &= \gamma v_t + \nabla \ell(\theta_t) \\ \theta_{t+1} &= \theta_t - \eta_t v_{t+1} \end{aligned} \quad (6.6)$$

Où γ est un hyperparamètre du *momentum*. *Adaptive Gradient* (*Adagrad*) est une modification de SGD permettant de réduire le LR au fil des itérations [237]. Cela a pour effet de réduire l'impact des changements de poids au fil du temps, ce qui peut aider à converger vers un minimum local.

$$\begin{aligned} g_t &= \frac{1}{n} \sum_{i=1}^n \nabla_{\theta} \mathcal{L}(x^{(i)}, y^{(i)}, \theta_t) \\ G_t &= \sum_{\tau=1}^t g_{\tau} g_{\tau}^T \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\epsilon I + \text{diag}(G_t)}} \cdot g_t \end{aligned} \quad (6.7)$$

Où η est le LR initial et ϵ est une petite valeur permettant d'éviter la division par zéro. Le succès de cet algorithme repose sur la matrice G_t qui représente la somme du produit extérieur des gradients jusqu'au temps t . *Root Mean Square Propagation* est une amélioration de *Adaptive Gradient* [238]. La règle de mise à jour de $RMSProp(H_t, \eta_t, \gamma, \rho, \epsilon)$ est :

$$\begin{aligned} v_0 &= 1, \quad m_0 = 0 \\ v_{t+1} &= \rho v_t + (1 - \rho) \nabla \ell(\theta_t)^2 \\ m_{t+1} &= \gamma m_t + \frac{\eta_t}{\sqrt{v_{t+1} + \epsilon}} \nabla \ell(\theta_t) \\ \theta_{t+1} &= \theta_t - m_{t+1} \end{aligned} \quad (6.8)$$

Ces deux méthodes utilisent une moyenne mobile pondérée pour déterminer le LR au temps t pour chaque itération. Ils incluent un hyperparamètre γ , généralement établis à 0,95. *Adaptive Gradient Methods (Adam)* est la fonction la plus utilisée [239]. Elle combine *Root Mean Square Propagation* et *SGD* avec *Momentum* ensemble. *Adam* possède plusieurs avantages comme un besoin faible en mémoire, une forte efficacité algorithmique et offre de bonnes performances lorsqu'il est utilisé avec de larges ensembles de données. La mise à jour de $Adam(H_t, \alpha_t, \beta_1, \beta_2, \epsilon)$ est la suivante :

$$\begin{aligned}
m_0 &= 0, & v_0 &= 0 \\
m_{t+1} &= \beta_1 m_t + (1 - \beta_1) \nabla \ell(\theta_t) \\
v_{t+1} &= \beta_2 v_t + (1 - \beta_2) \nabla \ell(\theta_t)^2 \\
b_{t+1} &= \frac{\sqrt{1 - \beta_2^{t+1}}}{1 - \beta_1^{t+1}} \\
\theta_{t+1} &= \theta_t - \alpha_t \frac{m_{t+1}}{\sqrt{v_{t+1} + \epsilon}} b_{t+1}
\end{aligned} \tag{6.9}$$

Les valeurs par défaut de β_1 , β_2 et ϵ sont respectivement de 0,9, 0,999 et $10E-8$. *NAdam* correspond à la combinaison de *Adam* et *Nesterov* [240]. La règle de mise à jour des paramètres θ de $NAdam(H_t, \alpha_t, \beta_1, \beta_2, \epsilon)$ se définit comme suit :

$$\begin{aligned}
m_0 &= 0, & v_0 &= 0 \\
m_{t+1} &= \beta_1 m_t + (1 - \beta_1) \nabla \ell(\theta_t) \\
v_{t+1} &= \beta_2 v_t + (1 - \beta_2) \nabla \ell(\theta_t)^2 \\
b_{t+1} &= \frac{\sqrt{1 - \beta_2^{t+1}}}{1 - \beta_1^{t+1}} \\
\theta_{t+1} &= \theta_t - \alpha_t \frac{\beta_1 m_{t+1} + (1 - \beta_1) \nabla \ell(\theta_t)}{\sqrt{v_{t+1} + \epsilon}} b_{t+1}
\end{aligned} \tag{6.10}$$

AdamW est une variation de *Adam* qui a une implémentation améliorée de *weight decay* [241]. *AdamW* possède une meilleure capacité de généralisation que *Adam*.

Des études empiriques ont tenté de comprendre les relations complexes qui existent entre les fonctions d'optimisations et les résultats subséquents [234]. Cette recherche empirique a comparé les principales fonctions d'optimisation (*Adam*, *AdamW*, *NAdam*, *Momentum*, *Nesterov*, *Root Mean Square Propagation* et *SGD*) sur des tâches de classification d'image. Les résultats indiquent que *Root Mean Square Propagation*, *Adam* et *NAdam* surpassent *SGD*, *Nesterov* et *Momentum* dans la plupart des situations testées. Il s'avère toutefois qu'il n'existe pas de fonction d'optimisation optimale pour chacune des catégories de test [233]. En somme, il est nécessaire de bien choisir la fonction d'optimisation et d'optimiser ses hyperparamètres (LR , γ , β_1 , β_2 et autres) afin de se rapprocher du minimum local.

6.5 Fonction de perte

La façon d'évaluer l'apprentissage du modèle a un impact considérable sur la qualité d'un entraînement [242, 243]. Il existe des fonctions de perte spécifique pour chacune des tâches reliées à la vision par ordinateur : régression, classification, détection et segmentation. Au niveau de la segmentation sémantique, il y a différents types de fonctions de pertes : basée sur la distribution, sur la région, sur les frontières et composée. La Figure 6.7 montre les différents types de fonctions de perte et les fonctions associées.

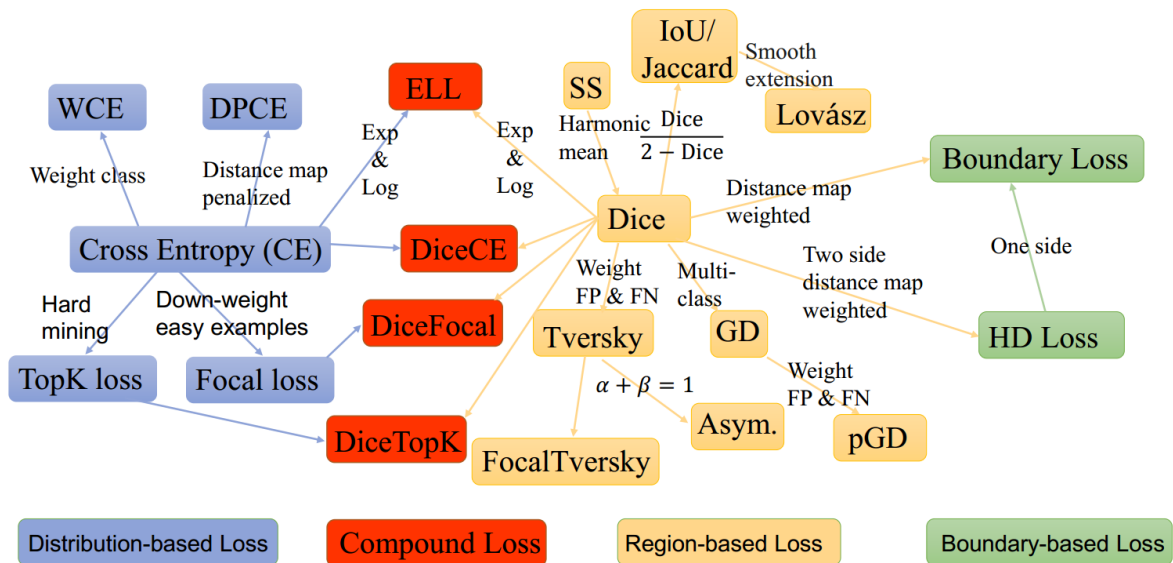


Figure 6.7 : Type de fonctions de perte pour la segmentation, tiré de [216]

Chaque type quantifie la différence entre le *GT* et les prédictions de façon différente, mais partage tout le même objectif qui est d'atteindre les meilleurs résultats de segmentation possible. Les fonctions de pertes les plus couramment utilisées en segmentation sont l'entropie croisée, *dice*, *focal* et leur dérivée (qui sont améliorations apportées aux versions originales).

L'entropie croisée est définie comme une mesure de la différence entre deux distributions de probabilité pour une variable aléatoire ou un ensemble d'événements donné [244]. En segmentation, elle est utilisée pour mesurer la classification au niveau des pixels. Il existe des variants à l'entropie croisée comme l'entropie croisée pondérée où des poids sont donnés aux exemples positifs. Ce variant est particulièrement utile dans le cas où l'une des catégories importantes est sous-représentée. La perte *focale*, *Focal Loss (FL)*, peut être vue comme un autre variant de l'entropie croisée. Elle réduit l'importance des exemples faciles permettant ainsi au modèle de se concentrer

sur les exemples plus difficiles [245]. Les exemples faciles sont généralement associés aux classes les plus représentées alors que les exemples difficiles correspondent aux classes rares (par exemple certains types de défauts). La *FL* fonctionne particulièrement bien dans les cas où les données sont déséquilibrées. La *FL* est formulée ainsi :

$$FL = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (6.11)$$

Où $-\log(p_t)$ correspond à l'entropie croisée et $\gamma > 0$. Quand $\gamma = 1$, la *FL* devient simplement l'entropie croisée.

Les fonctions de pertes basées sur la distribution évaluent les pixels individuellement sans considérer les objets dans leur ensemble, ce qui occasionne des artéfacts. Les fonctions de perte basées sur la région évaluent les objets dans leur ensemble, ce qui résout le problème. Le *DSC* est une mesure largement utilisée dans la communauté de vision par ordinateur pour calculer la similitude entre deux images. La fonction de perte *dice* est :

$$Dice(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1} \quad (6.12)$$

Où $y, \hat{p}$ est la paire *GT*, prédiction. Une multitude d'améliorations ont été apportées à cette fonction de perte. Les performances des modèles utilisant *Soft Dice Loss* (SDL) outrepassent régulièrement ceux utilisant l'entropie croisée. SDL équivaut à :

$$SDL(y, \hat{p}) = 1 - \frac{2y\hat{p}}{y^2 + \hat{p}^2} \quad (6.13)$$

Soft dice loss peut en revanche mener à un biais volumétrique indésirable, en sous-estimant ou surestimant la taille des objets segmentés dans le cas où il y a une forte incertitude [246]. *Generalised Dice Loss* (GDL) est une autre variation de *dice loss* permettant de réduire les problèmes de classes déséquilibrées, fréquentes en END [247]. Elle correspond à :

$$GDL(y, \hat{p}) = 1 - 2 \frac{\sum_{l=1}^L w_l \sum_n y_{ln} \hat{p}_{ln}}{\sum_{l=1}^L w_l \sum_n y_{ln} + \hat{p}_{ln}} \quad (6.14)$$

Où w_l est pour fournir une invariance aux différentes propriétés de l'ensemble annoté.

La fonction de perte asymétrique a été développée afin de faire la segmentation d'image médicale très déséquilibrée [248]. Les modèles de segmentation peuvent souffrir du déséquilibre des données en ayant des prédictions fortes en précision, mais faibles en rappel. Ces modèles entraînés sont généralement biaisés vers les classes les plus représentées, ce qui peut être très problématique

dans certaines applications, comme celle-ci, où les faux négatifs sont plus importants que les faux positifs. Les fonctions de pertes basées sur le *DSC* calculent les faux négatifs comme ayant la même importance que les faux positifs, ce qui en fait une fonction de perte symétrique. Afin de faire un meilleur ajustement de l'importance des faux négatifs et des faux positifs, la fonction de perte asymétrique est proposée. Cette fonction de perte est basée sur le score F_β qui est défini comme suit :

$$F_\beta = (1 + \beta^2) \frac{\text{précision} \times \text{rappel}}{\beta^2 \times \text{précision} + \text{rappel}} \quad (6.15)$$

Le score F_β est réécrit en fonction des ensembles P et G :

$$F_\beta = \frac{(1 + \beta^2) \sum_{i=1}^N p_i g_i}{(1 + \beta^2) \sum_{i=1}^N p_i g_i + \beta^2 \sum_{i=1}^N (1 - p_i) g_i + \sum_{i=1}^N p_i (1 - g_i)} \quad (6.16)$$

Où p_i est la probabilité du voxel i d'être positif (défaut par exemple) et $(1 - p_i)$ est la probabilité que le voxel i soit négatif (par exemple sans défaut ou le matériel de base). Les données du *GT* g_i équivalent à 1 pour un résultat positif (défaut, soudure en V, soudure en double-V et réflexion incomplète) et 0 pour un résultat négatif (sans défaut ou matériel de base). Il est important de souligner que la fonction de perte asymétrique est binaire. Il faut donc faire le calcul pour chaque classe séparément. L'*IoU* est largement utilisée en IA. L'*IoU* est illustrée à la Figure 6.8.

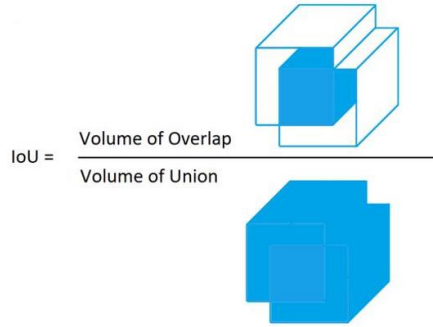


Figure 6.8 : Visualisation de *IoU*

Il s'agit de la somme de l'intersection des deux ensembles sur l'union des deux ensembles :

$$IoU(y, \hat{p}) = 1 - \frac{\sum_{c=1}^C \sum_{n=1}^N y_c \hat{p}_c}{\sum_{c=1}^C \sum_{n=1}^N y_c + \hat{p}_c - y_c \hat{p}_c} \quad (6.17)$$

Sensitivity-Specificity (SS) est une fonction de perte ayant pour objectif de maximiser ces deux caractéristiques importantes. SS correspond à $\lambda \cdot \text{Sensitivity} + (1 - \lambda) \cdot \text{Specificity}$ Où $\text{Sensitivity} = \frac{VP}{VP+FN}$ et $\text{Specificity} = \frac{VN}{VN+FP}$. L'équation complète de SS se veut comme-ci :

$$SS(y, \hat{p}) = \lambda \frac{\sum_{n=1}^N (y_n - \hat{p}_n)^2 r_n}{\sum_{n=1}^N y_n + \epsilon} + (1 - \lambda) \frac{\sum_{n=1}^N (y - \hat{p}_n)^2 (1 - y_n)}{\sum_{n=1}^N (1 - y_n) + \epsilon} \quad (6.18)$$

Où λ est le poids permettant l'ajustement de chaque caractéristique et ϵ est nécessaire pour éviter les divisions par zéro. La fonction de perte Tversky peut être vue comme une généralisation de la *dice Loss*. Elle ajoute un poids aux faux positifs et aux faux négatifs à l'aide de β .

$$TL(y, \hat{p}) = 1 - \frac{1 + y\hat{p}}{1 + y\hat{p} + \beta(1 - y)\hat{p} + (1 - \beta)y(1 - \hat{p})} \quad (6.19)$$

Quand $\beta = \frac{1}{2}$, Tversky devient la *dice loss*. La fonction de perte Focal-Tversky (FTL) combine les avantages des deux fonctions de perte focale et Tversky. FTL permet d'apprendre des exemples difficiles à l'aide du coefficient γ qui varie entre 1 et 3.

$$FTL = (L_{Tversky})^{\frac{1}{\gamma}} \quad (6.20)$$

Les fonctions de pertes basées sur le *DSC* sont grandement utilisées en segmentation. Les performances des modèles l'utilisant sont de haute qualité. En revanche, en raison de sa nature non convexe, *dice loss* pourrait échouer à atteindre les résultats optimaux. La fonction de perte Lovász a été développée pour résoudre le problème des fonctions de perte non convexe en ajoutant le lissage à l'aide de l'extension Lovász [249]. Lovász est similaire à l'*IoU*, mais elle utilise une stratégie différente de substitution. En détail, *IoU* remplace simplement la segmentation dans l'indice Jaccard par des probabilités softmax, alors que Lovász utilise un substitut convexe linéaire par morceaux à la fonction de perte *IoU* basée sur l'extension Lovász de fonctions d'ensemble sous-modulaire.

L'approche *LogCosh* a été largement utilisée dans les problèmes de régression pour lisser la courbe. *LogCoshDice* (LC-D) est proposé pour permettre de résoudre le problème de fonction de perte non convexe [242]. *LogCoshDice* est formulé ainsi :

$$LC - D = \log(\cosh(\text{DiceLoss})) \quad (6.21)$$

Où $\cosh = \frac{e^x + e^{-x}}{2}$. LC-D se situe entre [-1, 1]. Il s'agit d'une variante de *dice* particulièrement intéressante pour les ensembles de données asymétriques. La fonction de perte basée sur la distance de Hausdorff est basée sur les frontières. Elle est définie comme le maximum de toutes les

distances minimales entre deux courbes. La distance de Hausdorff est de nature non convexe ce qui limite son utilisation, bien que l'objectif de chaque modèle de segmentation soit intrinsèquement d'optimiser la distance de Hausdorff. La distance de Hausdorff (DH) se définit mathématiquement comme ceci :

$$DH(X, Y) = \max_{x \in X} \min_{y \in Y} \|x - y\|_2 \quad (6.22)$$

Comme illustré dans cette section, il existe de multiples fonctions de perte ayant toute pour objectif de permettre une segmentation de qualité.