



Septembre 2024

Antoine PERRIN-DELORT

Consultant Data Scientist - MP DATA

Rapport de Stage de fin d'étude

Étudiant FISE - DaSci

Sous la supervision école de Romain BILLOT

Sous la supervision entreprise de Quentin ANDRÉ



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom



© 2024 - ANTOINE PERRIN-DELORT - ANTOINE.PERRINDELORT@GMAIL.COM
ALL RIGHTS RESERVED.

INSTITUT MINES TÉLÉCOM ATLANTIQUE
BREST, BRETAGNE, FRANCE

MP DATA
BOULOGNE-BILLANCOURT, 92100, FRANCE

GAZTRANSPORT ET TECHNIGAZ
SAINT-RÉMY-LÈS-CHEVREUSES, 78470, FRANCE

Confidentialité

Ce rapport a été anonymisé et les données confidentielles ont été floutées par rapport à sa version scolaire, en accord avec GTT afin de permettre sa diffusion.



Consultant Data Scientist - MP DATA

ABSTRACT

Le présent rapport a été réalisé dans le contexte de mon stage de fin d'études, effectué à l'issue de la [spécialisation en Data Sciences](#) d'IMT Atlantique. Dépêché par MP DATA chez Gaztransport et Technigaz (GTT), j'ai pendant 6 mois travaillé à la mise en place d'une Intelligence Artificielle afin de pouvoir vérifier automatiquement les plans réalisés par le client. Ces plans contiennent notamment des dessins techniques, qui sont invariants par un certain nombre de transformations, rendant inutilisables les techniques traditionnelles de détection d'erreurs. C'est une problématique encore en cours d'exploration en Computer Vision. La méthodologie utilisée pour résoudre ce problème était expérimentale, confortée par un papier de recherche et impliquant un réseau siamois, dont le réseau dupliqué a été entraîné sur des données historiques. Avant d'aboutir à cette IA, j'ai implémenté différents tests afin de valider l'outil mis en place par mon tuteur et dont je me suis servi pour la labélisation, puis extraire les données historiques de GTT afin de constituer différents datasets. Un code d'entraînement structuré et adaptable a été élaboré afin de pouvoir être le plus libre possible pour le choix des paramètres d'entraînement. De plus, l'explicabilité étant au cœur de la démarche, de nombreuses visualisations, dont des Grad-CAMs, ont été instaurées. Enfin, les entraînements ont été effectués sur un cluster interne à l'entreprise, qu'il a fallu prendre en main pour ensuite choisir la stratégie de distribution adaptée.

Les différentes connaissances et compétences mobilisées étaient en adéquation avec les enseignements vus en cours dans la TAF DaSci, et plus globalement dans mon cursus au sein d'IMTA.

Table des matières

1	AVANT-PROPOS	1
2	CONTEXTUALISATION	2
2.1	MP DATA	2
2.1.1	Présentation de l'entreprise d'accueil	2
2.1.2	Les valeurs de MP DATA	4
2.1.3	Responsabilité Sociétale des Entreprises	4
2.2	Gaztransport et Technigaz	5
2.2.1	Présentation du client	5
2.2.2	Une entreprise aux nombreuses filiales	9
2.2.3	Raison d'être & valeurs	10
2.2.4	Responsabilité Sociétale des Entreprises	11
2.2.5	Organisation Interne	13
2.2.6	Mon positionnement au sein de GTT	15
3	PRÉSENTATION DE LA MISSION	19
3.1	Le problème à résoudre	19
3.2	Les acteurs	24
3.3	Les enjeux	25
3.4	Résultats attendus	26
4	RÉALISATION DE LA MISSION	29
4.1	Familiarisation avec les plans	29

4.1.1	Explication détaillée	29
4.1.2	Champ d'application actuel	30
4.2	Etat de l'art	30
4.3	Intelligence Artificielle retenue	32
4.4	Création du dataset de classification	34
4.4.1	Vérité des plans	35
4.4.2	Récupération des plans	37
4.4.3	Créations des images & labels	38
4.5	Création du dataset pour le réseau siamois	40
4.5.1	Création de paires d'images différentes	41
4.5.2	Supervision d'un labelisateur	42
4.5.3	Récupération des plans & labels	42
4.5.4	Création des images & zones d'intérêt	43
4.6	Entrainement de l'IA de classification	43
4.6.1	Choix du modèle	43
4.6.2	Utilisation d'un cluster de calculs	44
4.6.3	Stratégies de parallélisation	46
4.6.4	Choix des paramètres d'entraînement	48
4.7	Explicabilité de l'IA & métriques utilisées	50
4.7.1	Matrice de confusion	50
4.7.2	Courbe de Précision-Rappel	51
4.7.3	Saliency Map	52
4.7.4	Embeddings du modèles	54
4.7.5	Histogramme des poids & biais	55
5	DIFFICULTÉS RENCONTRÉES	57
5.1	Difficultés liées aux tests logiciels	57
5.2	Difficultés liées à la création des datasets	58
5.2.1	Extraction des plans	58
5.3	Difficultés liées au cluster	58
5.3.1	Prise en main	58
5.3.2	Firewall	59
5.3.3	Ordonnancement	59

5.3.4 Stratégies de parallélisation	60
5.4 Entrainement	61
5.4.1 Nombre d'entrainements	61
5.4.2 Temps d'entraînement	61
5.4.3 Entrainements non-probants	61
5.5 Difficultés liées aux métriques	63
5.5.1 Embeddings	63
5.5.2 Histogramme des poids & biais	64
6 RÉSULTATS	65
6.1 Réseau de classification	65
6.2 Réseau Siamois	68
7 PRISE DE RECUL	69
7.1 Bilan du travail effectué	69
7.2 Critique des méthodes utilisées	70
7.2.1 Travaux préalables	70
7.2.2 Stratégies de parallélisation	71
7.3 Enseignements et projet professionnel	71
7.4 Liens avec cursus suivi au sein d'IMT Atlantique	72
7.5 Impact du projet & individuel	72
7.5.1 Impact Economique & Organisationnel	73
7.5.2 Impact environnemental	73
7.6 Conclusion	75
A DÉTAIL DES DONNÉES	76
A.1 Les technologies de GTT	76
A.2 Les cuves	78
A.2.1 La géométrie	78
A.2.2 Les différentes zones	78
A.3 Les différents composants	80
A.4 Coding	82
A.5 Les différents types de plan	83
A.6 La nomenclature des livrables	83

A.7	Exemples de plan	84
B	CALCUL DE LA TAILLE MÉMOIRE D'UN RÉSEAU	86
C	ALGORITHME D'ORDONNANCEMENT DE JARVIS	88
D	CALCULS DE L'IMPACT ENVIRONNEMENTAL DE L'IA	90
D.1	Méthodologie	90
D.2	Données à disposition	91
D.3	Entrainements	92
D.3.1	Calcul d'un entraînement	93
D.3.2	Estimation des résultats préalables / tests / débbugages	94
D.4	Inférences	94
E	EXEMPLE D'ENTRAÎNEMENT EN ATTENTE	96
	GLOSSAIRE	102
	REFERENCES	114

Table des figures

2.2.1 Prévision de la demande en GNL en 2018	6
2.2.2 Les hexapodes de GTT	8
2.2.3 GTT et ses filiales	9
2.2.4 Les 10 règles d'or de sécurité de GTT	12
2.2.5 Émissions de gaz à effet de serre pour différents carburants fossiles	13
2.2.6 Organigramme Général GTT	15
2.2.7 Organigramme Sous-Direction des Plans	18
3.1.1 Organisation interne d'un projet	20
3.1.2 Les différents éléments d'un plan	21
3.1.3 Exemple d'invariance d'un dessin technique	22
3.1.4 Positionnement d'AutoCheck au sein de la Sous-Direction des Plans	23
3.2.1 Les acteurs liées à la mission	24
3.4.1 Exemple comparaison systématique	27
4.2.1 Un même plan selon différent niveaux de résolution	31
4.3.1 Fonctionnement d'un réseau siamois	33
4.4.1 Création du dataset de classification	35
4.4.2 Application du processus de fabrication du dataset à une image	39
4.5.1 Illustration d'un cas limites de plans similaires pour le réseau siamois	41
4.6.1 Illustration d'un modèle de classification multi-label	44
4.6.2 Illustration d'un modèle de classification multi-output	44
4.6.3 Topologie d'un noeud de Jarvis	45

4.6.4 Topologie de Jarvis	45
4.6.5 Fonctionnement de la Distributed Data Parallel	47
4.6.6 Fonctionnement de la Fully Sharded Data Parallel	47
4.7.1 Explication de la précision et du rappel	51
4.7.2 Illustration de l'obtention des Class Activation Map	53
4.7.3 Exemple représentation embeddings	54
4.7.4 Fonctionnement d'un neurone en Machine Learning	55
4.7.5 Exemple de distribution de poids & biais	56
6.1.1 Évolution de la précision	66
6.1.2 Évolution du rappel	66
6.1.3 Évolution du f_2 _score	66
6.1.4 Évolution du learning rate	66
6.1.5 Résultats	66
6.1.6 Matrice de confusion binaire	67
6.1.7 Gradcams d'un plan correctement prédit	68
7.1.1 Résumé du travail effectué	70
7.5.1 Exemple base de données de plans requêtables	73
A.1.1 Les technologies de GTT	77
A.2.1 Formats de cuves	78
A.2.2 Localisation et numérotation des cuves	78
A.2.3 Localisation des Flatwall, Dièdres et Trièdres au sein d'une cuve	79
A.2.4 Localisation des différentes zones spéciales	80
A.3.1 Localisation des différentes éléments de la technologie MARK III	81
A.3.2 Apparence d'une Membrane Sheet (MS)	81
A.4.1 Nomenclature des Flat Panels (FP)	83
A.6.1 Nomenclature des livrables	84
A.7.1 Échantillon de plans de GTT	85
E.0.1 Etat du cluster à t+44	97
E.0.2 Priorité des jobs	97
E.0.3 Disponibilité d'un noeud	98

E.o.4Etat du cluster à t+72	98
E.o.5Etat du cluster à t+82	99
E.o.6Etat du cluster à t+125	100
E.o.7Etat du cluster à t+129	100

Liste des tableaux

4.4.1 Encodage catégorielle	40
4.4.2 Encode one-hot	40
4.4.3 Exemple d'encodage one-hot à partir d'un encodage catégoriel	40
C.o.1 Description des différents types de calculs	89

Remerciements

Tout d'abord, je souhaiterai remercier Quentin André, mon maître de stage, pour son encadrement à la fois bienveillant et pertinent ainsi que pour les nombreux conseils prodigués durant mon stage qui me seront utiles dans le futur. Je souhaite notamment insister sur la facilité du dialogue qui était des plus appréciable.

De manière plus large, je tiens à exprimer ma reconnaissance à toute l'équipe d'AutoCheck, et donc notamment à Juliette Darnal et Guillaume Morin avec lesquels il fut agréable de travailler, de part leur bonne humeur et leur professionnalisme.

Dans la même lancée, je tiens à remercier Nicolas Marois, autre collègue de MP DATA en mission chez GTT sans qui certaines pauses et afterworks auraient été bien moroses !

Plus globalement, mes remerciements sont également destinés à l'ensemble de l'équipe de la Sous-Direction des Plans pour leur accueil ainsi que pour les bons moments passés ensemble.

Je ne manquerai certainement pas d'exprimer ma gratitude envers Romain Billot, mon professeur encadrant au sein d'IMT Atlantique, qui m'a suivi avec pédagogie sur différents projets tout au long de cette année et avec qui j'ai toujours eu plaisir à échanger.

Mes pénultièmes remerciements sont destinées à MP DATA pour avoir eu confiance en mes capacités et m'avoir accordé ce stage.

Finalement, je tenais à remercier mon entourage dans son ensemble, que ce soit ma famille ou bien mes amis, pour toute l'aide et le soutien qu'ils ont su et pu m'apporter au fil des années.

1

Avant-propos

Le rapport suivant, relatant mon expérience de 6 mois en tant que Consultant en Data Science, vient mettre un terme à mes études au sein d'IMT Atlantique commencées il y a 3 années, en 2021. En effet, c'est à la fois la dernière étape et une condition *sine qua non* à l'obtention de mon diplôme d'Ingénieur Généraliste.

En remontant plus loin, c'est un cycle de 5 années d'études qui s'achève, étant donné que pour rentrer au sein d'IMT Atlantique, j'ai dû passer des concours au terme de 2 années de Classes Préparatoires aux Grandes Écoles.

Enfin, plus globalement, ne pensant pas à ce jour continuer en thèse contrairement à certains de mes camarades habilement convertis par certains professeurs, ce rapport et ses quelques dizaines de pages viennent parachever l'ensemble de ma scolarité.

Ainsi se clôt le chapitre « Études» de ma vie et commence celui intitulé « Vie active»...

Ready to work with passionate people ?

- MP DATA

2

Contextualisation

2.1 MP DATA

2.1.1 PRÉSENTATION DE L'ENTREPRISE D'ACCUEIL

MP DATA est une entreprise de conseil spécialisée en Data et Intelligence Artificielle (IA), fondée en 2015 par [Grégoire GAILLAC](#), l'actuel PDG. Le cœur de métier de la société est l'acquisition, le traitement et la valorisation des données, ce qui se reflète dans les services et l'accompagnement proposés à ses clients :

- Intelligence Artificielle & Optimisation : Expertise en Intelligence Artificielle (IA), Machine Learning (ML) et Recherche Opérationnelle (RO). Sur ces projets, l'entreprise présente un ROI client moyen de 300 000€ et pouvant aller jusqu'à plusieurs millions d'euros.
- Data Engineering & MLops : Mise en place de Pipelines CI/CD sécurisées et de solutions Cloud
- Conseil et Stratégie Data : Élaboration de roadmap pour transitionner vers l'IA et la Data, chiffrage de chaque cas d'usage et étude de faisabilité systématique

La société compte une centaine de collaborateurs et est en rapide expansion.

Le siège de l'entreprise est situé à Boulogne-Billancourt mais l'entreprise possède également des bureaux à Toulouse et Clermont-Ferrand. De plus, l'entreprise a comme objectif **Vision 2026** de s'implémenter dans 9 nouvelles villes afin d'étendre sa zone d'activité.

Même si historiquement, la concentration stratégique de MP DATA est l'industrie, l'entreprise a su se diversifier au fil des années et effectuer des missions au sein de secteurs plus variés, dont voici une liste non-exhaustive agrémentée d'exemples de réalisation :

- Transport (Aéronautique, ferroviaire, automobile...) : [Optimisation de flottes d'avions](#) ; [Prévision d'affluence](#)
- Energie (Industrie pétrolière, renouvelable...) : [Maintenance prédictive d'éoliennes et panneaux photovoltaïques](#) ; [Deep Learning pour les réseaux d'acheminement de gaz](#)
- Industrie (Sidérurgie, chimie...) : [Machine Learning pour optimisation de synthèse chimique](#) ; [Ordonnancement Adaptatif](#)
- Sciences naturelles (Pharmacie, cosmétique) : Détection automatique des défauts d'étiquetage des flacons cosmétiques, [Optimisation de procédés de synthèse de médicaments](#)
- Défense & Spatial : Analyse automatique d'images satellites ; Maintenance prédictive des avions de chasse
- Média (Télévision, régies publicitaires...) : Segmentation client et Analytics

- sur la e-chaine de l'un des plus grands médias français
- Banques & Assurances : [OCR pour le reconnaissance de documents d'identités](#); Détection de fraude à l'assurance vie
 - E-commerce : Analyse NLP sur les commentaires clients sur le site e-commerce d'un leader du retail

2.1.2 LES VALEURS DE MP DATA

MP DATA possède 3 valeurs fondamentales qui se retrouvent dans les actions et objectifs de l'entreprise :

1. **L'excellence** : Les ingénieurs de MP DATA étant formés dans les Grandes Ecoles d'ingénieurs françaises (X, Mines, Centrale ...). Des formations et certifications sont également proposées aux employés afin de se tenir à jour de l'état de l'art et de se parfaire aux dernières avancées technologiques.
2. **L'engagement** : L'entreprise possède un fort engagement Santé, Sécurité et Environnement (SSE), se conformant aux exigences de la MASE, se fixant des objectifs annuels en plus de tenir des réunions mensuelles et d'avoir mis en place un processus d'amélioration continue à ce sujet.
3. **Le partage** : L'entreprise ayant récemment mis en place un "Data Lab" ainsi qu'un processus de veille technologique collaborative et de référents techniques sur les différents sujets de l'IA et la Data. Ce système par les consultants, pour les consultants, vise à ce que chacun puisse profiter des compétences des autres afin de développer les siennes.

2.1.3 RESPONSABILITÉ SOCIÉTALE DES ENTREPRISES

Concernant la Responsabilité Sociétale des Entreprises (RSE), l'entreprise fait part d'un fort engagement SSE comme évoqué sous-section 2.1.2. Pour ajouter à cela, on peut également noter la présence d'équipements visant à améliorer la santé des employés comme des bureaux motorisés, permettant de travailler debout ou assis, et permettant donc de changer de position de travail ce qui est encouragé.

Du point de vue environnemental, l'entreprise mène des projets dans des domaines responsables comme les énergies renouvelables mais également dans des domaines

polluants comme la pétro-chimie ou le transport du GNL (sous-section 2.1.1). L'optimisation des procédés dans ces domaines pouvant avoir à la fois des effets bénéfiques ou néfastes sur l'environnement.

L'entreprise possède une marge d'amélioration dans le domaine, comme la mise en place du remboursement total des transports en commun afin que ses collaborateurs privilégient ces derniers, voire dans la même lancée en fournissant des aides à l'achat ou l'utilisation du vélo comme moyen de transport.

Alors que l'IA est globalement une industrie polluante MP DATA ne semble pas particulièrement engagée pour l'écologie et l'environnement en menant une réflexion quant à l'impact énergétique de ses projets.

2.2 GAZTRANSPORT ET TECHNIGAZ

2.2.1 PRÉSENTATION DU CLIENT

Gaztransport et Technigaz (GTT) est un bureau d'étude français dont la principale activité consiste en la conception de plans de fabrication de cuves destinées au transport du Gaz Naturel Liquéfié (GNL). Le siège de l'entreprise est situé à Saint-Rémy-Lès-Chevreuses dans les Yvelines et ses principaux chantiers partenaires en Chine et en Corée du sud.

L'entreprise a été fondée en 1994 de la fusion des deux entités éponymes de l'industrie navale, Gaztransport et Technigaz, elles même respectivement créés en 1965 et 1963. Cette fusion avait pour but de mettre en commun les technologies développées par chacune des deux sociétés afin de pouvoir s'imposer sur le marché des systèmes de confinement du GNL.

Avec le recul, cette stratégie s'est avérée payante, étant donné que l'entreprise détient un monopole mondial et quasi-total dans son secteur d'activité, à raison de 95 à 100% des parts de marché des méthaniers et des terminaux flottants en fonction des années. La société a explosé en capitalisation depuis la début de la guerre en Ukraine, du fait de l'utilisation croissante des méthanier pour acheminer du gaz vers l'Europe. Ce gain en popularité des méthaniers s'expliquant notamment par la volonté de l'Union Européenne de ne plus dépendre du gaz russe, et donc de moins en importer via [les deux gazoducs Nord](#)

Stream mais également du **sabotage de ces derniers** aujourd'hui attribué à l'Ukraine [6].

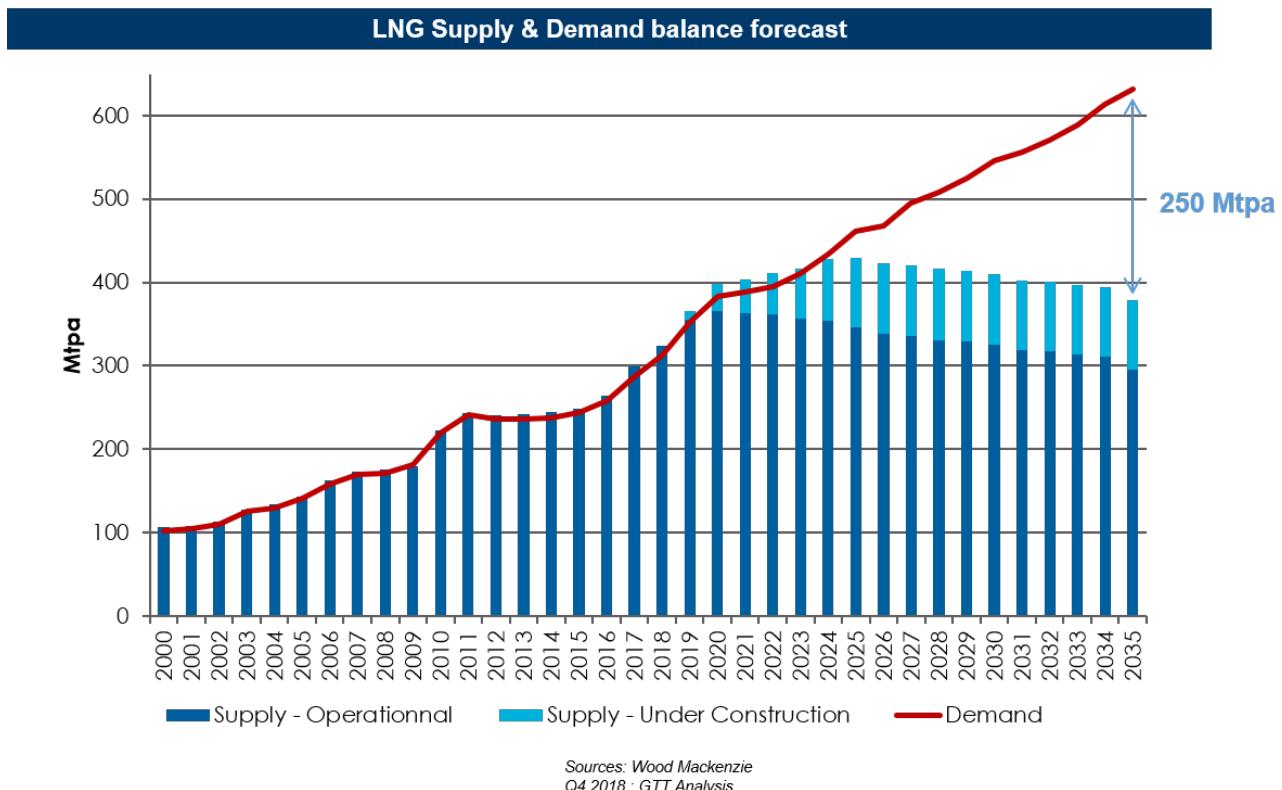


Figure 2.2.1 – Prévision de la demande mondiale de GNL en Millions de tonnes par an (Mtpa) réalisée en 2018. Même sans la croissance de la demande liée à la guerre en Ukraine, un manque d'offre et donc un essor économique pour GTT, avait été prévu

Cependant, bien que l'importation de GNL ait augmenté de 60% entre 2021 et 2022 [2], il semblerait que l'Europe ait déjà connu son pic de consommation de GNL lié à la guerre en Ukraine [3], avec une consommation en baisse d'environ 20% sur le premier semestre 2024.

Cette position de leader technologique est dû à de forts investissements en Recherche & Développement (R&D), en adéquation avec le fait que le cœur de métier de GTT est son expertise technologique. Ces investissements ont permis le développement et perfectionnement de plusieurs technologies permettant de maintenir le gaz à basse température, de sorte qu'il reste à l'état liquide et *de facto* occupe moins de place, mais également d'utiliser le gaz se vaporisant pour propulser les navires !

Ces investissements se reflètent à différents niveaux :

— C'est une des ETI déposant le plus de brevets en France, à raison d'une soixantaine de brevet par an. Toutes catégories confondues, c'est la 23ème entreprise en France.

— L'entreprise possède ses propres hexapodes, permettant de simuler le *sloshing*. Le sloshing est le mouvement d'un liquide contenu dans un récipient en accélération, entraînant la formation de vagues au sein du récipient. Dans le cadre du transport du GNL, c'est une source de considérations techniques importantes, les vagues créées impliquant de nombreuses contraintes de résistances au niveau des cuves. Cela induit également un réchauffement du liquide et donc une vaporisation qui augmente la pression au sein des cuves.

— GTT a créé et acquis de nombreuses filiales comme évoqué dans la partie suivante.



Figure 2.2.2 – GTT dispose de quatre robots hexapodes capables de simuler les mouvements de houle. Les données enregistrées par leurs 100 capteurs permettent d'analyser les effets du ballottement du GNL dans sa cuve.

2.2.2 UNE ENTREPRISE AUX NOMBREUSES FILIALES



Figure 2.2.3 – Localisation de GTT et ses différentes filiales

Au fil des années, GTT a fondé ou acquis de nombreuses filiales, afin de répondre à plusieurs besoins. On peut notamment distinguer :

Les filiales permettant à GTT de se développer à l'international et dont les activités englobent la formation de nouveaux employés ainsi qu'un service d'assistance 24/24. Ces filiales comportent :

- GTT Training
- GTT North America
- GTT South EastAsia

Les filiales permettant à GTT de développer de nouvelles technologies, afin de proposer de

nouveaux services, d'améliorer ceux existant voire de préparer une future transition de l'entreprise, étant donné son domaine d'activité. Ces différentes filiales comptent :

- Cryovision (2012) qui propose des services de contrôle de l'usure et de l'état des cuves à partir de caméras thermiques et de capteurs acoustiques ;
- Ose Engineering (2014) qui rassemble des experts en IA, analyse de données et modélisation & simulation de systèmes complexes ;
- Cryometrics (2015) qui commercialise des logiciels permettant la transmission en temps réel de données et la surveillance du comportement des cuves et du liquide qu'elles contiennent ;
- AscenzMarorka (2018-2019) dont la principale activité est le Smart Shipping ;
- Elogen (2020) spécialisée dans la conception et l'assemblage d'électrolyseurs et permettant la production d'hydrogène « vert ». L'acquisition de cette filiale devrait permettre à GTT de ne plus dépendre des énergies fossiles et permet donc à l'entreprise de prévoir une transition de son secteur d'activité qui, à terme, ne seront plus exploitables ;

2.2.3 RAISON D'ÊTRE & VALEURS

SÉCURITÉ, EXCELLENCE, INNOVATION, TRAVAIL EN ÉQUIPE, TRANSPARENCE, telles sont les valeurs sur lesquelles s'appuie l'activité de GTT. C'est la mise en œuvre de ces exigences, de cette culture, qui a permis à la société d'atteindre la position qui est la sienne au niveau mondial. A travers elles, GTT a su gagner des relations de confiance solides et durables avec ses clients.

- **Sécurité** : GTT opère dans le secteur des technologies du transport et du stockage du gaz liquéfié, ce qui conduit le groupe à attacher une très grande importance à la sécurité. GTT se doit d'assurer la sécurité de ses collaborateurs, de ses technologies, de ses services et de ses clients. En effet, étant donné le volume de gaz transporté, si un accident venait à se produire, celui-ci rivaliserait en terme de conséquences désastreuses avec [l'accident de Beyrouth en 2020](#) et ne manquerait pas de mener la société à sa faillite ;

— **Excellence** : GTT doit tendre vers l'excellence dans tous ses processus, condition qui lui permet de rester présent sur ses marchés, présents et à venir.

— **Innovation** : Au fil du temps, les technologies de GTT se sont imposées sur le marché mondial en tant que références dans le transport du GNL. Cette démarche d'innovation se poursuit à tous les niveaux (technologies, organisation) et est la condition *sine qua non* pour conforter son avance technologique mais également assurer la survie de l'entreprise étant donné son modèle économique : l'entreprise produit des brevets et plans qu'elle vend à ses clients ;

— **Travail en équipe** : Les activités de GTT nécessitent un travail en équipe permanent : en interne, avec ses clients, les clients de ses clients et ses fournisseurs ;

— **Transparence** : Renforcer la transparence dans ses relations permet à GTT d'établir des relations de confiance à long terme avec ses clients directs, ses clients finaux et entre ses collaborateurs ;

2.2.4 RESPONSABILITÉ SOCIÉTALE DES ENTREPRISES

GTT oeuvre à différents niveaux concernant les enjeux RSE.

Au niveau sociétal, l'entreprise lutte contre le harcèlement avec des campagnes de sensibilisation par le biais d'affiches vectrices de phrases stéréotypée et/ou injurieuses.

Du côté de la sécurité du travail, GTT fait tout son possible pour empêcher le moindre accident, y compris hors de ses enceintes :

- 10 « golden rules » de la sécurité sont affichés dans toutes les salles ;
- il existe des KPI liés au nombre d'incidents afin de mieux pouvoir comprendre les causes et les éviter ;
- l'entreprise dépêche des inspecteurs sur les chantiers (Corée, Chine...) afin de s'assurer que ces derniers respectent également les normes de sécurité ;



Figure 2.2.4 – Les 10 règles d'or de sécurité de GTT

Sur le point de vue écologique, GTT est à la fois un bon et mauvais élève. Car même si la firme permet la généralisation du GNL comme type de carburant pour les navires, qui possède les effets les moins néfastes sur l'environnement comparé au gazole et au « bunker » [13] [11], ainsi que plus globalement l'utilisation par les entreprises et ménages du GNL en lieu et place du charbon, pétrole ou fioul, l'entreprise ne montre pas pour autant patte blanche.

En effet, le GNL reste un carburant fossile et ne constitue pas une transition valable comme le montre le non-soutien de la [Banque Mondiale](#) et du [Programme des Nations Unies pour l'Environnement](#) [11].

Comparison of emissions by fuel type

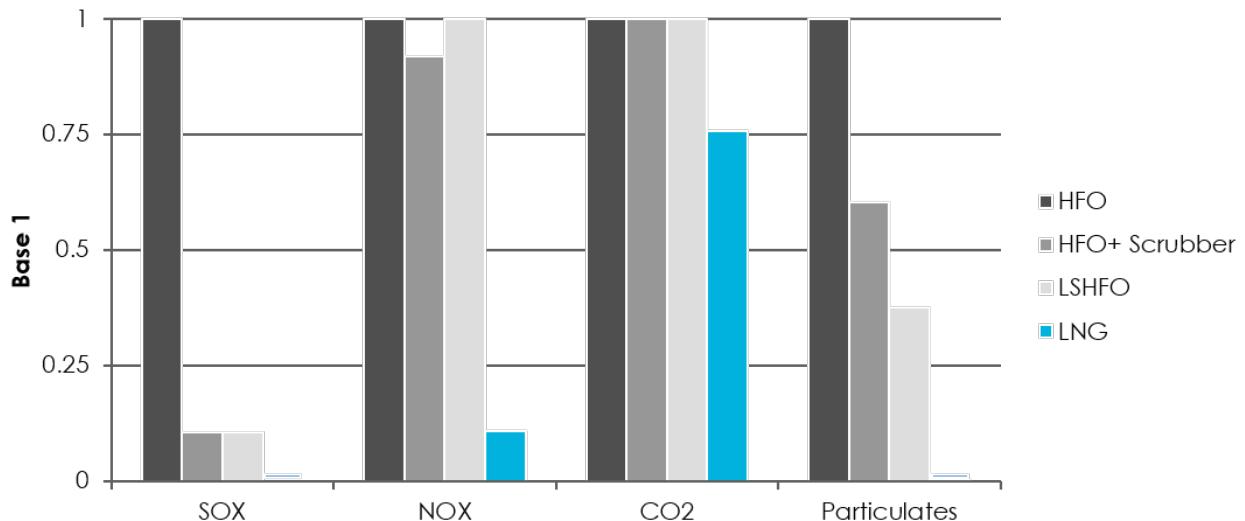


Figure 2.2.5 – Bien que le GNL produise moins de gaz à effet de serre que les autres carburants fossiles, il ne constitue pas pour autant énergie de transition

Le groupe effectue également des investissements dans les énergies propres, comme le montre son rachat d'Elogen, et investit dans des sociétés « vertes » pour compenser une partie de son bilan carbone. Il faut néanmoins garder à l'esprit que ces considérations écologiques sont également motivées par des considérations économiques (déclin futur assuré du marché des énergies fossiles et pénalités financières liées au bilan carbone). Mais cela reste des points positifs du point de vue écologique.

Enfin, même si jusqu'à présent aucun des principaux clients de GTT ne semble avoir été épingle par l'ONG [Shipbreaking Platform \[1\]](#) pour abandon de navires au lieu de son désarmement, on peut tout de même noter qu'il n'existe pas de [programme à responsabilité élargie du producteur](#) qui permettrait de s'assurer du recyclage, ou à défaut du bon démantèlement des cuves construites grâce à l'expertise de GTT.

2.2.5 ORGANISATION INTERNE

Le groupe est organisée en structure hiérarchique, avec à sa tête la Direction Générale. Cette dernière élabore la stratégie du groupe et conduit la revue de direction annuelle. Les

différents aspects de l'activité et du fonctionnement de l'entreprise sont dirigés par des départements :

- Le Secrétariat Général : veille au respect des obligations légales et réglementaires. Ce département est composé de différents services : juridique, conformité, qualité et Hygiène Sécurité Environnement (HSE) ;
- La Direction de l'Innovation : conduit le développement de nouvelles technologies et gère la propriété intellectuelle ;
- La Direction Commerciale : gère les relations commerciales avec les chantiers et établit les différents documents contractuels ;
- La Direction Technique : fournit les notes d'étude, les plans propres à un projet aux chantiers et assure un support technique ;
- La Direction du Digital : s'assure du bon fonctionnement des services numériques : de leur mise en place et de leur bon fonctionnement ;
- La Direction Administrative & Financière : met à disposition, contrôle les ressources financières et intègre les Systèmes d'Informations ;
- La Direction des Ressources Humaines : gère le personnel et la communication interne ;

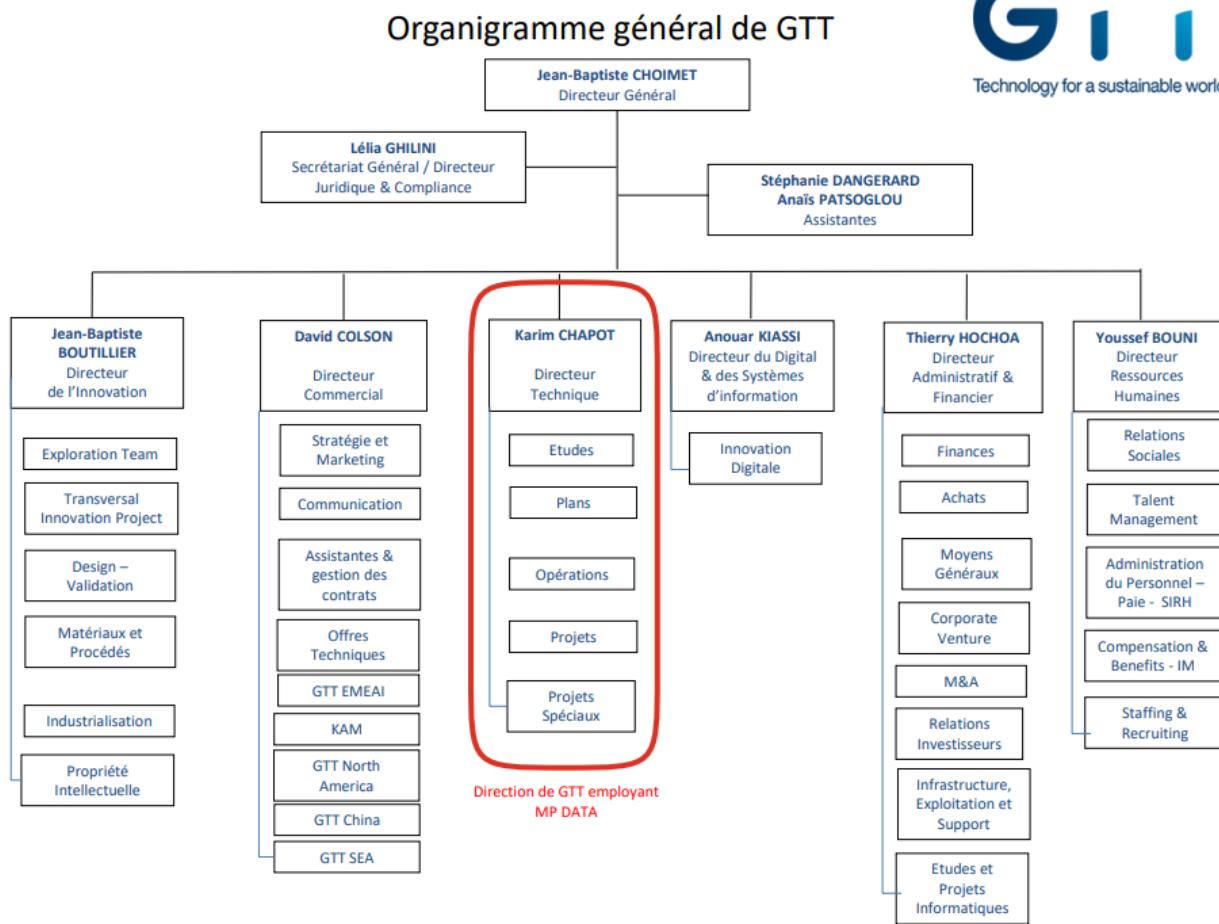


Figure 2.2.6 – Organigramme général de l'organisation de GTT

2.2.6 MON POSITIONNEMENT AU SEIN DE GTT

Mon stage s'effectue au sein de la Direction Technique, et plus précisément de la Sous-Direction des Plans (SDP). La Direction Technique est organisée en plusieurs sous-direction, à savoir :

- La Sous-Direction des Projets (SDPr) : s'occupe de piloter les projets en terme de coûts, qualités et délais
- La Sous-Direction des Opérations (SDO) : fournit une assistance technique aux clients de GTT lors de la construction sur le chantier ainsi que la maintenance
- La Sous-Direction des Études (SDE) : fournit les notes d'études (calculs de structure et dynamique des fluides)
- La SDP : fournit les plans de confinement et de manutention de gaz lors des phases avant-projet

La SDP, est dirigée par Julien DELAHAYE et est elle-même divisée en 3 services :

- Containment System A (CSA) : travaille sur le côté gestion de projet du revêtement des cuves ;
- Containment System B (CSB) : travaille sur le côté design du revêtement des cuves ;
- Handling System (HS) : travaille sur les systèmes de manutention des gaz ;

A la tête de chacun de ces services se trouve un responsable qui planifie et coordonne les ressources du service, en plus de présenter les avancements et problèmes éventuels lors de réunions avec les autres managers. Les 3 services sont organisés de manière similaire, avec comme différents rôles en leur sein :

- Référent Métiers Outils CAO : prend en charge les projets de développement d'outils Conception Assistée par Ordinateur (CAO). Ces différents outils sont transverses aux CSA, CSB et HS. Il assure également la relation avec la Direction Informatique (DSI)
- Référent Métier : définit les règles de design et en assure le respect sur les plans ;
- Référent pôle plans : améliore les techniques de réalisation des plans et gère les dessinateurs ;
- Lead Design : prend en charge des projets et est responsable des livrables des projets en coût, qualité et délais ;
- Lead Design Confirmé : spécialisé dans les relations clients du point de vue technique ;
- Lead Outils CAO : qui prend en charge le développement d'outils CAO, de modèles 3D et de méthodologies de travail ;
- Dessinateur : prend en charge la réalisation des plans au service des projets ;



Figure 2.2.7 – Organigramme de la sous-direction des plans

*Votre mission, à supposer que vous l'acceptiez,
consiste à...*

- Mission Impossible

3

Présentation de la mission

3.1 LE PROBLÈME À RÉSOUDRE

Le client livre des plans de construction de cuves permettant le transport de GNL à différents chantiers. En une année, c'est quelques centaines de navires qui peuvent ainsi être construits sur la base de 25 à 50 projets différents.

Un projet de conception de cuves est composé de nombreux plans réalisés par les dessinateurs au sein de la Sous-Direction des plans. Pour donner un ordre d'idée, un unique projet comporte environ :

- 1 100 références (sur les 13 500 élaborées par GTT)
- 2 300 dessins techniques
- 180 plans, répartis sur une centaine de livrables pour un total de 3 000 à 10 000 pages selon la technologie

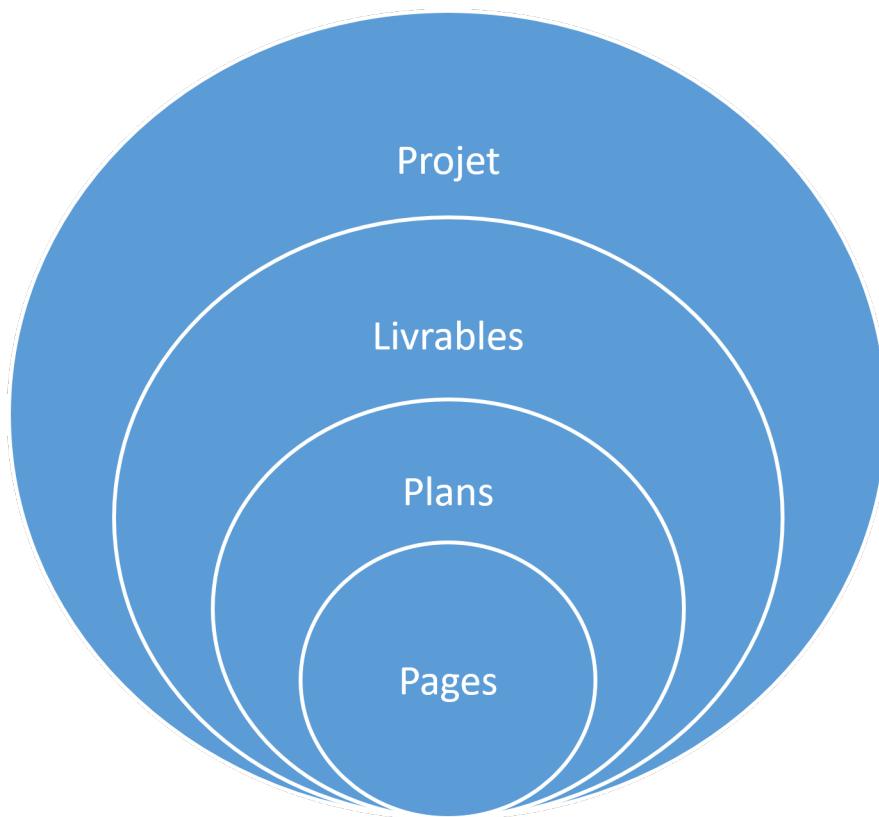


Figure 3.1.1 – Organisation interne d'un projet

Ainsi, un unique projet comporte des milliers voire des dizaines de milliers de pages de documents PDFs.

Les plans de GTT comportent de nombreux dessins techniques, qui comprennent un certains nombre de contraintes géométriques (concernant les angles, distances...) que les différents éléments du plans se doivent de respecter. La plus simple étant que la longueur des différents éléments doit respecter l'échelle du plan.

Au sein de ces plans, on distingue 3 types d'éléments :

- Les éléments invariants : dont on peut inférer le positionnement par des règles : cartouche, titre, échelle...
- Les tables : comportant des données tabulaires difficilement récupérables
- Les éléments variants : dont on ne peut inférer le positionnement, en grande partie ce sont les dessins techniques. Il est important de noter que la véracité

d'un plan est invariant par rotation, translation ou d'autres modifications comme des changements sur la localisation des flèches de cotation : cela représentera toujours les mêmes éléments et c'est ce qui rend la tâche de détection d'erreurs si compliquée car il existe une quasi-infinité de représentations pour un même dessin technique.

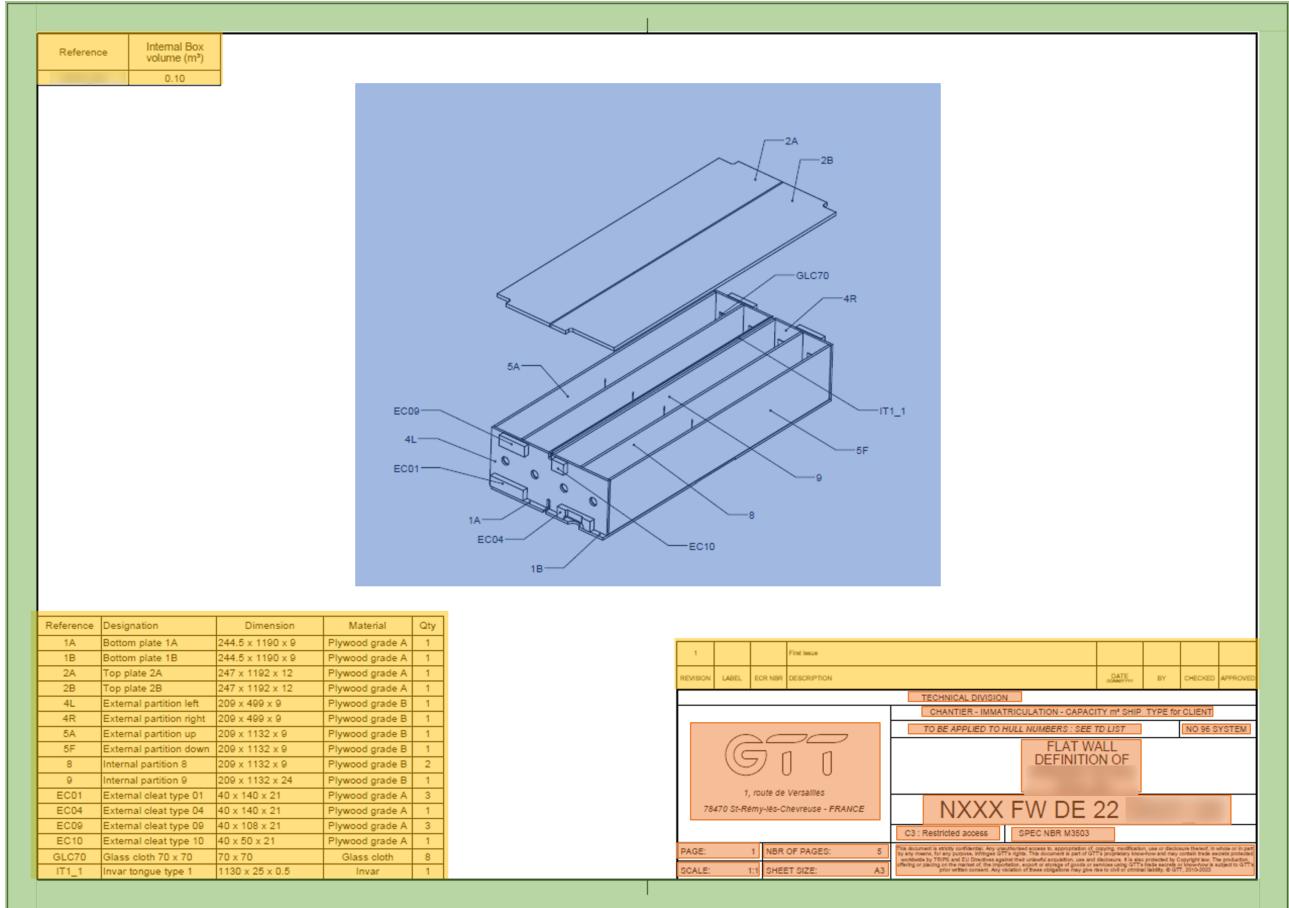


Figure 3.1.2 – Les différents éléments d'un plan : en orange les invariants, en jaune les tables, en bleu les variants. On a ici mis en évidence l'extérieur du cadre en vert.

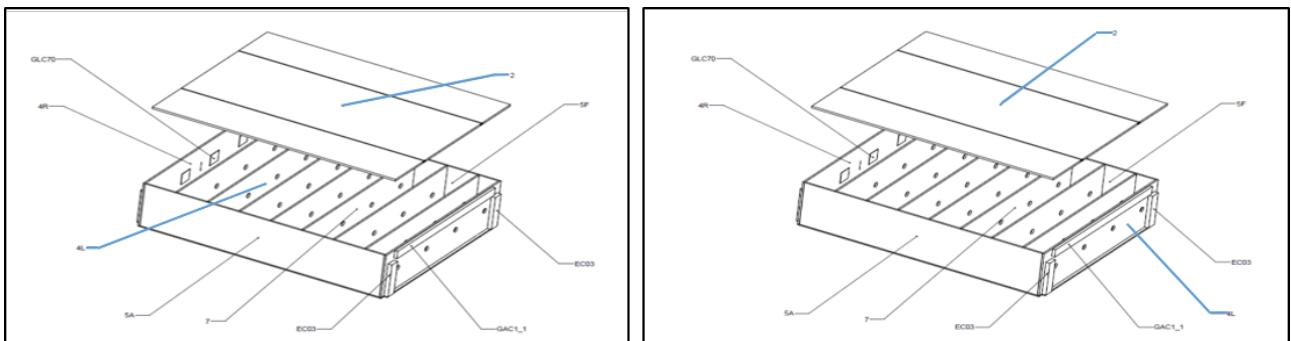


Figure 3.1.3 – Exemple d'invariance d'un dessin technique : deux flèches de légendes (mises en évidence en bleu) ont vu leur localisation modifiée et pourtant les deux images représentent le même dessin technique

Une fois élaborés, les plans sont communiqués aux différents chantiers et doivent donc ne comporter aucune erreur, sinon cela entraîne au mieux des retards, si l'erreur est détectée par GTT, ou au pire des pénalités si c'est le chantier qui les détecte. On les nomme respectivement Anomalie DéTECTée GTT (ADGTT) et Anomalie DéTECTée Client (ADC).

Ces anomalies font partie des KPI de GTT et il est donc important de pouvoir les détecter au plus tôt, i.e. par GTT. Pour cela tous les livrables passent par une étape de vérification.

La tâche de vérification des plans est laborieuse, répétitive, chronophage et surtout coûteuse ! En effet cette dernière peut prendre jusqu'à 10% du temps et des moyens investis dans chaque projet. L'action de vérifier les plans est nommée stroboscheck et consiste, par exemple dans le cadre d'un projet dit « copie » (basé sur un projet déjà réalisé), à ouvrir le livrable original et celui modifié et d'alterner rapidement entre les deux pour chaque page afin d'essayer de distinguer une différence par superposition rapide sur l'écran. A cette définition, on comprend la monotonie de la tâche et le faible gain de valeur ajoutée. Dans la suite, on nommera le fait de détecter les erreurs entre deux projets une détection d'erreurs « en comparaison ».

De plus, le stroboscheck permet certes de détecter des certaines erreurs, mais pas toutes, l'erreur étant évidemment humaine et pour certains projets, le nombre de pages à vérifier beaucoup trop important.

GTT souhaitant pouvoir détecter automatiquement des erreurs sur ses plans, MP DATA a été missionnée pour développer une application remédiant à ce problème. Cette application a été nommée **AutoCheck**. Ceci est la mission de MP DATA, mais me concernant **le problème a résoudre a consisté à effectuer une preuve de concept de l'utilisation de IA dans la détection d'erreur en comparaison.**

Quentin André travaille sur la mission depuis Février 2023 et s'est occupé de traiter les erreurs solvables systématiquement ainsi que de mettre en place une interface utilisateur et un serveur dédié à l'application. Durant mon stage, c'est sur ces même tâches qu'il a travaillé.

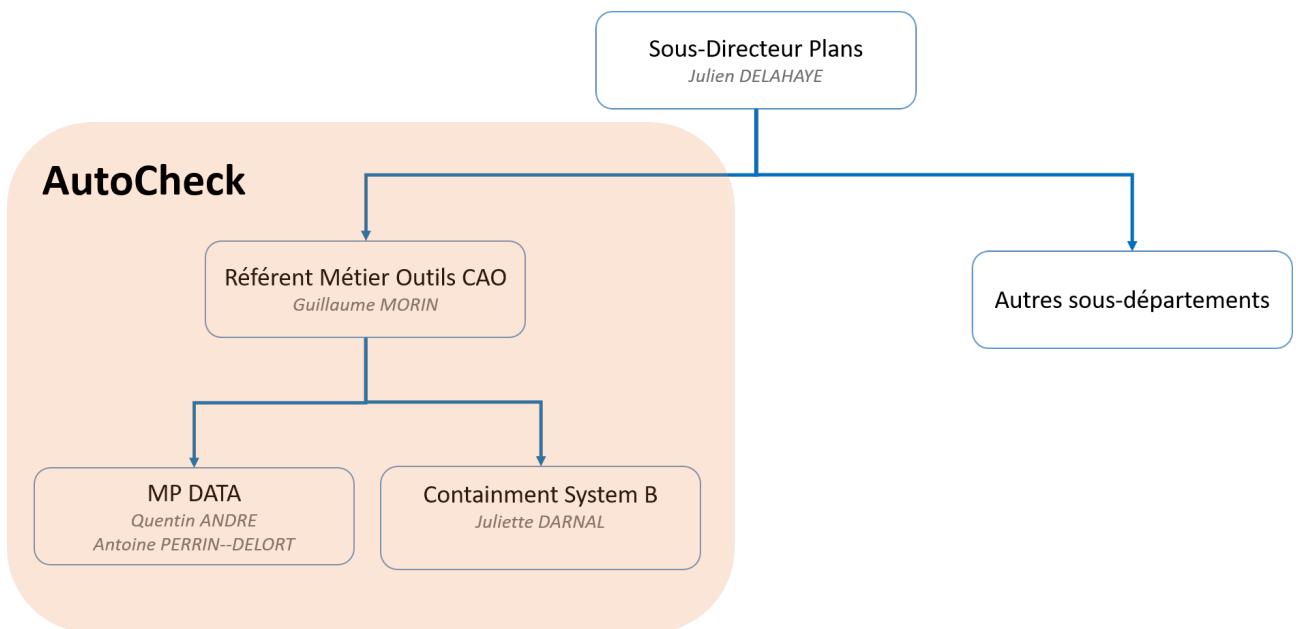


Figure 3.1.4 – Positionnement d'AutoCheck au sein de la Sous-Direction des Plans

Au sein de la sous-direction des plans, l'équipe d'AutoCheck que j'ai intégré courant avril est constituée de 4 personnes :

- Guillaume MORIN : Référent Métier Outil CAO dont le rôle a été défini ci-dessus ;
- Juliette DARNAL : membre du CSB, elle gère le projet dans sa globalité et assure le lien avec le côté métier en recueillant les retours d'expérience des dessinateurs afin de permettre une amélioration continue d'AutoCheck ;

— Quentin André : Mon tuteur et également consultant en Data Science. Il est en charge du développement technique du projet ;

— Antoine PERRIN-DELORT : Stagiaire de MP DATA en charge de la mise en place de l'IA pour le check des éléments variants ;

Cet effectif réduit couplé à des réunions hebdomadaires facilite la transmission des informations utiles au projet. De plus, le fait de travailler auprès des équipes ainsi que le travail de Juliette a grandement simplifié l'étoffement de l'application en terme de fonctionnalités et confort utilisateur.

3.2 LES ACTEURS

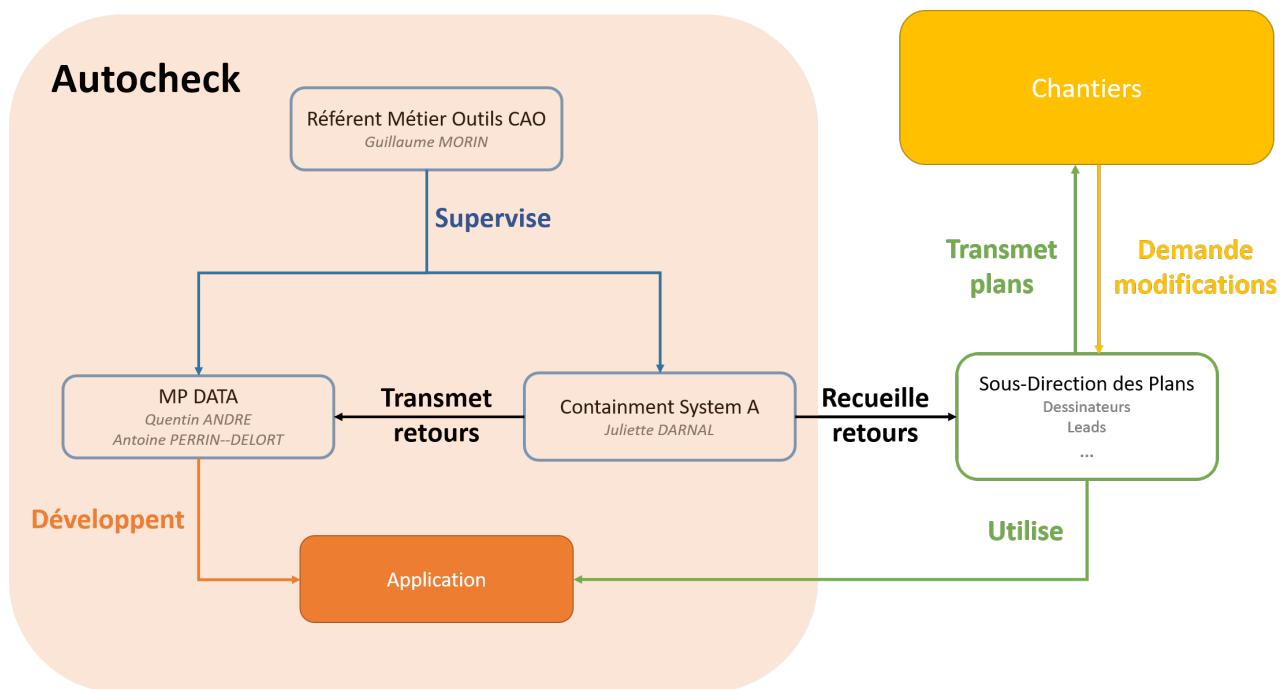


Figure 3.2.1 – Identification des différents acteurs liés à la mission

Le projet AutoCheck fait intervenir de multiples acteurs. Certains rôles, comme ceux tenus par Juliette DARNAL, Guillaume MORIN, Quentin ANDRÉ et moi-même ont déjà été définis ci-dessus (Figure 3.1). Cependant, les acteurs intervenant dans le cadre de ce projet ne se limitent pas à ces derniers.

Effectivement, le développement d'une application répond à un besoin, qui est exprimé par un acteur. C'est ici la Sous-Direction des Plans appartenant à GTT qui souhaite utiliser AutoCheck afin de faciliter son processus de stroboscheck des plans qu'elle doit transmettre aux différents chantiers. Le tout pour permettre de fluidifier les échanges GTT/chantiers en réduisant au maximum les échanges impliquant des erreurs sur les plans.

3.3 LES ENJEUX

Les enjeux de ce projet sont multiples pour les divers acteurs.

LES ENJEUX DE MP DATA

Du point de vu de MP DATA, ce projet, en plus du côté lucratif, doit permettre de nouer des relations avec son client qui possède de nombreux sujets pour lesquels l'entreprise pourrait proposer ses services . En effet, le succès de ce projet pourrait lui permettre de se démarquer par rapport à Ose Engineering , société interne à GTT dont le domaine d'activité est similaire à celui de MP DATA et à qui GTT confie la majorité des sujets liés à l'IA (cf sous-section 2.2.2).

De plus, GTT s'intéresse de plus en plus à l'IA et semble vouloir identifier et développer les différents cas d'usage que la société pourrait avoir. MP DATA pourrait donc profiter de cet engouement pour améliorer ses relations avec son client et ainsi obtenir de nouvelles missions.

Il est important de souligner, car il est rare que cela se fasse, que je ne suis pas facturé au client : MP DATA investit donc également sur moi.

Enfin, MP DATA souhaite capitaliser sur cette mission pour en décrocher de nouvelles chez d'autres clients comme Stellantis, SNCF ou Tractobell, qui ont des problématiques similaires et pour lesquels l'expérience engrangée par l'équipe d'AutoCheck pourrait les aider. En effet, le fait d'avoir déjà effectué cette tâche prouverait que MP DATA est capable de répondre à cette problématique, et que de plus elle possède des équipes expertes qui sauront implémenter une solution efficace et rapidement.

LES ENJEUX DE GTT

Concernant GTT, les enjeux existent à différents niveaux.

D'abord, du point de vue macroscopique, la réalisation de ce projet doit permettre l'amélioration de certains KPI (comme la diminution du nombre d'ADC et ADGTT) mais également de gagner en efficacité, notamment par le gain de temps non-négligeable de l'automatisation du stroboscopy qui s'élève environ à 220 Jours-Homme (JH) par an.

De plus, du point de vu des employés, les différents dessinateurs n'auront plus à réaliser de stroboscopy, tâche qu'ils n'apprécient guère.

Enfin, cette mission en tant que première liée à l'IA permet également à GTT de mettre à l'épreuve ses infrastructures et de tester leur dimensionnement dans le cas où l'entreprise souhaiterait faire plus souvent appel à des entreprises extérieures ou des talents internes au lieu de faire appel à Ose Engineering.

3.4 RÉSULTATS ATTENDUS

Mon rôle quant à moi a consisté à la fois à épauler mon tuteur dans la réalisation de certains besoins de l'application mais également de réaliser une preuve de concept sur l'utilisation de l'IA afin de pouvoir améliorer le traitement des éléments variants dans la détection d'erreurs en comparaison. En effet, une méthode systématique se basant sur des algorithmes d'[OpenCV](#) existe déjà pour les projets « copie pure » et servira de baseline : l'IA devra faire mieux.

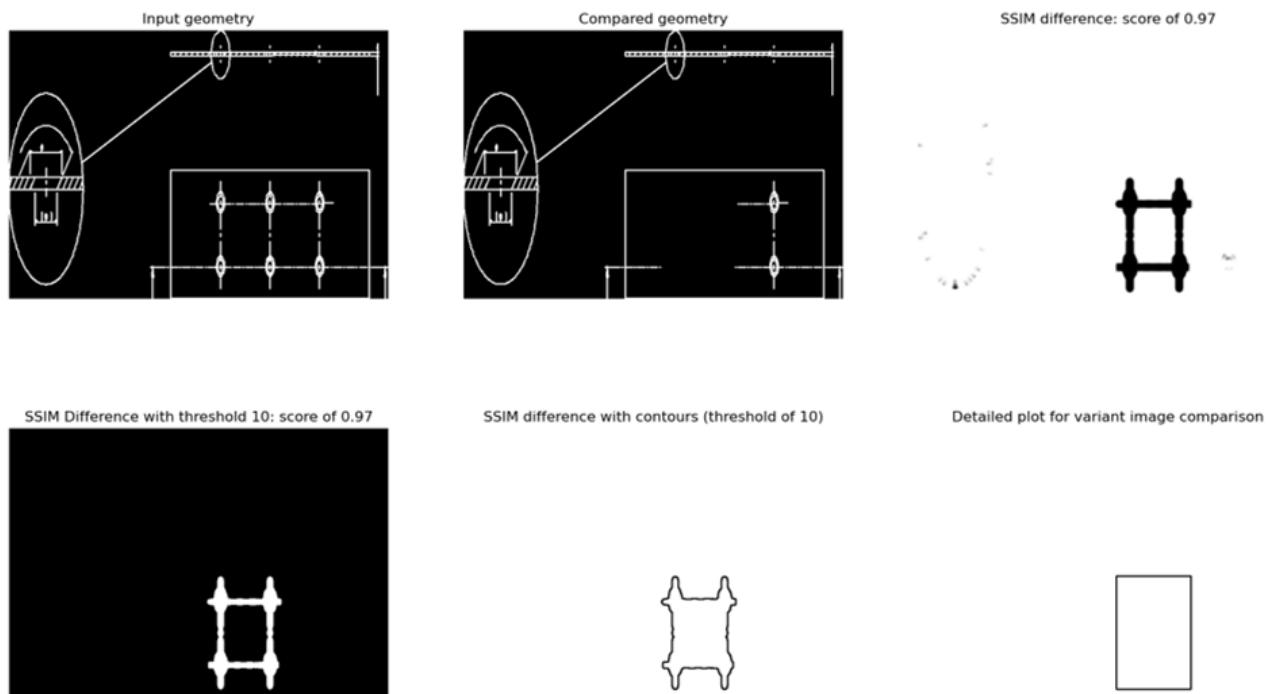


Figure 3.4.1 – Exemple de la mise en évidence de différences entre deux plans en « copie pure » à l'aide de l'**indice de similarité structurelle**. Dans un projet en copie pure, les différences sont suffisamment minimales pour pouvoir être détectées systématiquement (car le nombre de transformations invariantes est restreint : des éléments d'une page copiée ne vont pas être soudainement traduites...)

Notamment, mes divers objectifs ont été :

1. La mise en place de tests de scope et de non-régression sur une centaine de plans constituant un panel représentatif pour l'application durant mes 3 premiers mois ;
2. La mise en place d'un point API afin de récupérer les plans créés par GTT également durant mes 3 premiers mois ;
3. De contribuer à la réalisation d'une preuve de concept se basant sur l'IA et avec de meilleures performances que les outils systématiques préalablement utilisés avant la fin de mon stage ;
4. L'implémentation du maximum de critères d'explicabilité pour mieux comprendre les résultats de l'IA ;

Ces objectifs, et cela sera détaillé par la suite, sont liés entre eux. Notamment les deux premiers permettant de mettre en place le jeu de données pour entraîner l'IA.

*Le succès est la somme de petits efforts, répétés
jour après jour.*

- Leo Rober Collier

4

Réalisation de la mission

Dans un premier temps, une présentation des plans de GTT, afin de mieux comprendre les données à disposition sera faite. S'ensuivra une courte présentation de l'état de l'art et du modèle retenu, puis la phase de création des différents datasets ainsi que la prise en main d'un cluster de calculs pour entraîner l'IA avant d'enfin s'intéresser aux métriques d'explicabilité retenues.

4.1 FAMILIARISATION AVEC LES PLANS

4.1.1 EXPLICATION DÉTAILLÉE

Avant toute chose, il est nécessaire de comprendre au mieux les données. Une explication détaillée des plans de GTT est disponible Appendice A.

Il faut en retenir qu'il existe différents types de plans, différentes zones faisant intervenir

de nombreux composants sous différentes vues et que les composants possèdent eux-même de nombreux variants.

Une nomenclature, également détaillées Appendice A, régit la dénomination des pièces et des livrables.

Enfin, comme évoqué précédemment, il faut garder à l'esprit qu'un même plan possède une infinité de représentations, étant invariant par translation, rotation et déplacement de certains éléments comme illustré Figure 3.1.3.

4.1.2 CHAMP D'APPLICATION ACTUEL

La SDP produit différent livrables concernant les cuves, mais d'autres départements en produisent également ! Pour le moment le champ d'application d'AutoCheck se limite à certaines zones mais des élargissements successifs de ce champs d'application sont d'ores et déjà prévus pour l'avenir.

Les livrables pris en compte par l'application pour le moment sont les suivants :

- FW DB | PR | AR | MO [...] ;
- SO AR | DE [...] ;
- TC PR | AR | DE [...] ;
- DB PR | DE [...] ;

C'est donc autour de ces livrables que mon travail s'articulera, en gardant à l'esprit que ce champ d'application est voué à s'agrandir et qu'il faut donc prévoir une flexibilité et adaptabilité du code.

4.2 ETAT DE L'ART

Étant donné la perte de qualité et d'information induites par la diminution de la résolution des images qui nous obligeait donc à travailler avec de grandes images, ainsi que les contraintes géométriques auxquelles sont contraintes les dessins techniques, la réalisation d'un état de l'art s'imposait.

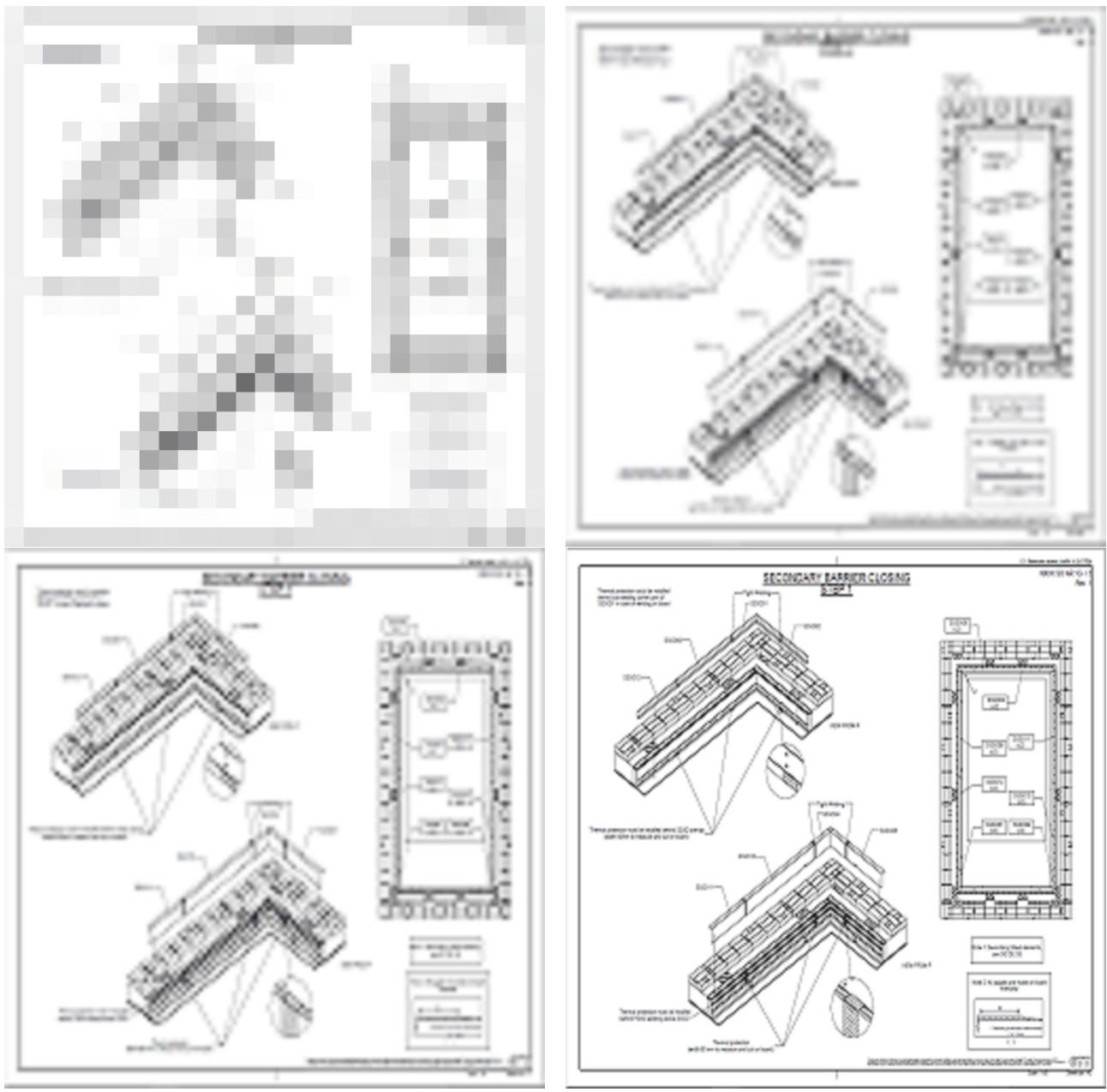


Figure 4.2.1 – Un même plan suivant les différentes résolutions couramment utilisées en ML : 28x28, 96x96, 128x128 & 256x256

La réalisation de l'état de l'art fut toutefois compliquée pour les raisons suivantes :

- La majorité des papiers étaient payants pour y accéder ;
- MP DATA ne possédait pas d'accès aux différentes librairies regroupant

des papiers de recherche ;

— Il y avait peu de papiers sur ce sujet étant donné que c'est un sujet de recherche privé ;

— Les rares papiers d'intérêt possédaient des lacunes quant à la reproductibilité des résultats : méthodologies et paramètres utilisés peu voire pas détaillés...

— Notamment, aucun code open-source n'était à disposition ;

Ainsi, sur la cinquantaine de papiers liés de près ou de loin à notre problématique, faisons un tour des plus intéressants :

1. Il existe des applications systématiques permettant de vérifier des dessins techniques réalisés par ordinateur, cependant cette approche semble utilisable pour des dessins simples mais ne permet pas d'appréhender l'essence d'un plan [7] ;

2. Des approches IA sont utilisables pour segmenter et extraire différentes parties d'un plan, notamment les dessins [15] [25], symboles [15] [14] et textes [19], cependant cette approche nous est peu utile étant donné que le travail préalablement réalisé par Quentin André permet d'extraire les différents éléments ;

3. Plusieurs méthodes d'OCR sont utilisées [23], dans notre cas, les PDFs étant recherchables l'intérêt est limité [19] ;

4. Des approches basées sur des Graph Neural Network, type de réseau permettant de traiter des graphes, ont également été appliquées, mais cela demande beaucoup de pré-processing, réflexions préalables et nous doutions que cela était applicable étant donné la complexité de nos plans [24] [26] ;

5. Enfin des approches basées sur des réseau siamois semblent obtenir des résultats prometteurs [10] [18], y compris avec de grandes images [10] ;

4.3 INTELLIGENCE ARTIFICIELLE RETENUE

Nous nous sommes inspirés de [10] pour aboutir à notre IA pour plusieurs raisons :

1. L'utilisation d'un réseau siamois avait été envisagée au préalable ;
2. Les résultats obtenus semblaient prometteurs pour un cas d'usage similaire ;
3. Les auteurs ont également dû travailler avec de grandes images (de l'ordre de 10^6 pixels), ce qui est également notre cas. Les plans sont au format SVG et peuvent cacher des détails si la résolution n'est pas assez bonne et c'est pour cela que nous avons choisi de travailler avec des grandes images ;
4. L'architecture utilisée permet de mettre en évidence les différences entre deux plans, afin de faciliter le contrôle par un utilisateur humain ;

Ce papier nous a donc véritablement conforté dans l'utilisation d'un réseau siamois afin de détecter et mettre en évidence des erreurs entre deux livrables.

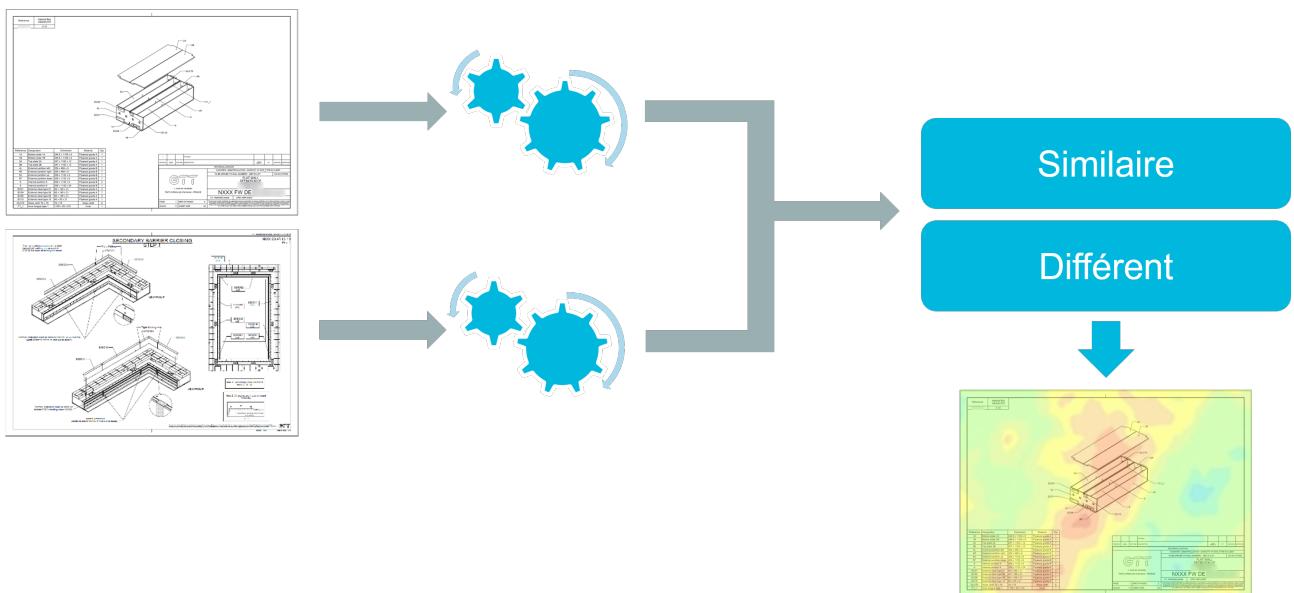


Figure 4.3.1 – Fonctionnement d'un réseau siamois avec mise en évidence des différences

Toutefois, à la différence de [10], nous avons souhaité entraîner le réseau de classification différemment. En effet, pour que ce dernier se familiarise avec les plans et en ait les meilleurs embeddings, représentation interne au réseau d'un objet sous forme vectorielle,

nous avons eu l'idée d'utiliser les résultats de la partie invariante d'AutoCheck pour labéliser les données, et donc de faire un entraînement supervisé au lieu de non-supervisé ! L'utilisation d'AutoCheck permet d'attribuer pour chaque page d'un livrable plusieurs valeurs pour différentes classes choisies, que notre réseau de classification devra deviner.

Ces classes sont les suivantes :

- La technologie : NO96 ou MARK III ;
- L'échelle : $\frac{1}{1}, \frac{1}{10} \dots$;
- La zone concernée : FW, DB ... ;
- Le type de plan : DE, PR ... ;
- L'élément concerné par le plan : BP, MS ... ;
- Le coding de cet élément : FPA 1-82-80A-03 ... ;

Donc, la première tâche a consisté à l'entraînement d'un réseau de classification dit multilabel, car il y avait plusieurs valeurs à prédire.

Suite à cela, nous nous sommes basés sur ce même réseau afin de former un réseau siamois, que nous avons affiné pour notre tâche de prédiction à l'aide d'un nouveau jeu de données créé spécialement pour cela (voir sous-section 4.5.2). En langage ML cette étape de partir d'un réseau entraîné et de le spécialiser dans une tâche se nomme « finetuning ».

Il aurait été possible de travailler directement sur les embeddings mais l'utilisation d'un réseau siamois possède plusieurs avantages :

1. Le réseau est entraînable et finetunable si l'on a le dataset adéquat : nul besoin de trouver la bonne métrique de similarité et les bons seuils manuellement ;
2. Les décisions sont explicables : on peut trouver les zones ayant menées à la décision ;

4.4 CRÉATION DU DATASET DE CLASSIFICATION

L'étape du choix ou de la création du dataset est cruciale en Machine Learning, car la qualité d'un modèle dépend énormément de la qualité des données sur lequel il s'entraîne. A minima, il faut que ces données soient **exactes, représentative et diverses**.

Explicitons ces termes, en prenant l'exemple d'un classifier devant déterminer si sur une

image se situe un chien ou un chat :

- **exactes** : les données sont correctement labelisées ie un chat n'est pas labelisé comme étant un chien ;
- **représentative** : les données correspondent au cas d'usage qui sera fait du modèle ie on ne l'entraîne pas sur des images de lama ;
- **diverses** : les données sont correctement distribuées ie on cherche à entraîner le modèle sur différentes races de chiens et chats afin que le modèle performe mieux et puisse davantage généraliser ;

Étant donné la quantité de données à disposition pour entraîner notre modèle, il a été choisi de labéliser automatiquement le dataset de classification. Pour opérer ainsi, nous avons eu besoin de :

1. Pouvoir labéliser de façon juste, précise et automatique les plans : pour cela nous utiliserons AutoCheck ;
2. Récupérer un ensemble de plans depuis PDM : le coffre-fort numérique utilisé par GTT ;
3. Transformer les fichiers pdf en images, et les stocker ainsi que les labels de manière cohérente ;

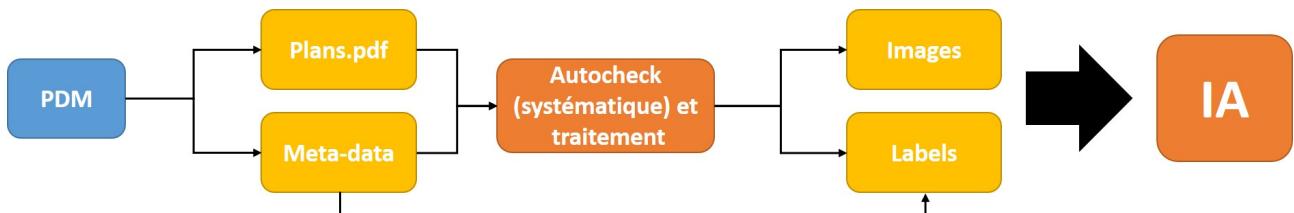


Figure 4.4.1 – Pipeline Crédit du dataset de classification

Nous allons traiter de ces différents points dans les sous-parties suivantes.

4.4.1 VÉRACITÉ DES PLANS

La méthode de labélisation automatique que nous avons souhaité utiliser se base sur AutoCheck qui, comme évoqué dans section 3.1, permet de vérifier les éléments invariants des plans. Mais il est primordial de s'assurer que les résultats de l'algorithme sont corrects.

La mise en place de tests constitue une étape importante du développement d'un outil ou d'une application. En effet, cela permet de vérifier que la solution proposée répond bien à la problématique posée, en s'assurant que les résultats obtenus concordent avec les résultats attendus.

Par conséquent, j'ai mis en place différents tests, à savoir des tests de scope et de non-régression, en ne rencontrant qu'une difficulté lors de ce travail cf section 5.1

TESTS DE SCOPE

Dans le cadre d'AutoCheck, j'ai implémenté les tests de **scope** pour la partie systématique : étant donné la diversité des plans produits par GTT au cours des années, il s'agit de s'assurer que l'application fonctionne pour les différents types de plans.

Pour mettre en oeuvre ces tests, il m'a fallu prendre un échantillon de plan et noter ce que l'application devrait renvoyer pour chaque plan de l'échantillon : on appelle cela la **vérité**. Il s'agit ensuite de comparer pour chacun de ces plans, la vérité et les résultats de l'application : s'ils sont égaux, alors le test est réussi. Mais s'il y a des différences, c'est que l'application ne fonctionne pas comme prévu. On met alors en évidence ces différences afin de pouvoir localiser le problème et le résoudre plus facilement.

La qualité de ce type de test dépend du choix de l'échantillon : il faut qu'il soit représentatif afin que chaque possibilité soit testée. Pour cela j'ai fait attention à prendre toutes les combinaisons de livrables possibles, sur différents projets afin de créer un échantillon de base d'une soixantaine de livrables (et donc plusieurs centaines voire milliers de pages). Puis cette base a été progressivement enrichie au fur et à mesure de la découverte de bugs ou d'échecs de l'application.

TESTS DE NON-RÉGRESSION

Les différents tests implémentés servent également de tests de **non-régression** : ils permettent de détecter si des changements non-voulus ont été introduits dans l'application lors de mises à jour.

Ainsi l'implémentation des différents tests permet de s'assurer du bon fonctionnement de

l'algorithme et donc de la véracité de la labélisation : les données du dataset seront bien **exactes**.

4.4.2 RÉCUPÉRATION DES PLANS

Pour pouvoir labéliser automatiquement des données, il nous faut des données et une manière de les labéliser. Nous venons de voir comment la labélisation allait s'opérer, reste à obtenir les données, ce que nous allons voir à présent.

Il faut savoir que GTT utilise Solidworks PDM (PDM), un coffre-fort numérique développé par Dassault Systèmes pour gérer les fichiers liés aux plans. Les livrables y sont stockées sous format .pdf. De plus, cet outil permet d'associer des « datacards », contenant des méta-données (capacité en m³, technologie, dates du projet...), à chaque fichier ou dossier. A ce stade du projet, nous avions décidé d'enrichir la labélisation à l'aide des méta-données et de choisir par la suite si ces dernières seraient conservées ou non.

Pour entraîner notre IA, nous souhaitions utiliser les plans créés par GTT au cours des 10 dernières années. Ce critère ayant été établi avec le métier, afin d'avoir le maximum de plans suivant les règles des plans d'aujourd'hui. En effet, des plans plus anciens sont radicalement différents car les règles, outils et méthodes utilisés par GTT pour les créer ont évolué.

S'il est facile d'extraire depuis l'interface utilisateur un unique projet, en extraire une centaine « à la main » s'est révélé être une tâche bien trop chronophage.

Heureusement, cette solution comprend également une [API](#) à partir de laquelle j'ai pu produire un code python pour extraire, non sans difficultés (cf sous-section 5.2.1), les plans voulus.

Une fois l'extraction effectuée, nous étions en possession de 84 projets qu'il nous a fallu transformer pour obtenir un dataset utilisable par l'IA. Puisque que nous avons récupéré l'ensemble des projets et livrables produits au cours des 10 dernières années, nous pouvons de facto considérer que les données de notre dataset sont **représentatives**.

4.4.3 CRÉATIONS DES IMAGES & LABELS

Les livrables obtenus à l'étape précédente sont en format .pdf et contiennent des plans au format SVG, ce qui n'est pas compatible avec les réseaux de neurones communément utilisés en Deep Learning.

Même s'il existe quelques réseaux de neurones utilisant des entrées en SVG [8], il n'est pas nécessaire ici d'y avoir recours comme l'a montré [10]. Néanmoins, le fait que tous les plans soient au format SVG a revêtu un énorme avantage car nous pouvions les redimensionner sans aucune perte de qualité.

Ainsi, nous avons transformé nos images au format le plus couramment utilisé en Computer Vision (CV) : le format .png. Cette transformation est appelée « rastérisation ». Chaque page d'un livrable deviendra une image, nommée de manière unique selon le numéro de ladite page et du coding du livrable :

$$\text{nom}_{\text{image}} = \text{nom}_{\text{livrable}} _\text{numéro page} \quad (4.1)$$

Cette nomenclature nous permet de facilement retrouver le livrable et la page d'origine et inversement si besoin.

Pour passer d'un livrable à des images, le *modus operandi* est le suivant :

1. Pour chaque livrable, nous utilisons AutoCheck pour récupérer les informations sur les parties invariantes de chaque page : leurs bounding boxes et labels ;
2. L'image est redimensionnée tant qu'elle est au format SVG, puis nous passons à des opérations sur les pixels ;
3. La première page est gardée si elle contient des éléments variants ;
4. Les tables sont gardées mais peuvent être masquées (des modifications sur AutoCheck étant nécessaires pour cela) ;
5. Les zones invariantes sont masquées car c'est grâce à elles que l'on labélise ;

6. Certaines zones variantes sont masquées car comportent des indices, notamment textuels, sur lesquels le modèle pourrait se baser pour deviner les labels (voir Figure 4.4.2);

7. L'image est recadrée de sorte à ne garder que le cadre (l'extérieur étant composé de pixels blanc c'est un gain d'efficacité de les enlever cf Figure 3.1.2);

8. L'image et ses labels associés sont enregistrés et regroupés par projet selon une arborescence pré-déterminée;

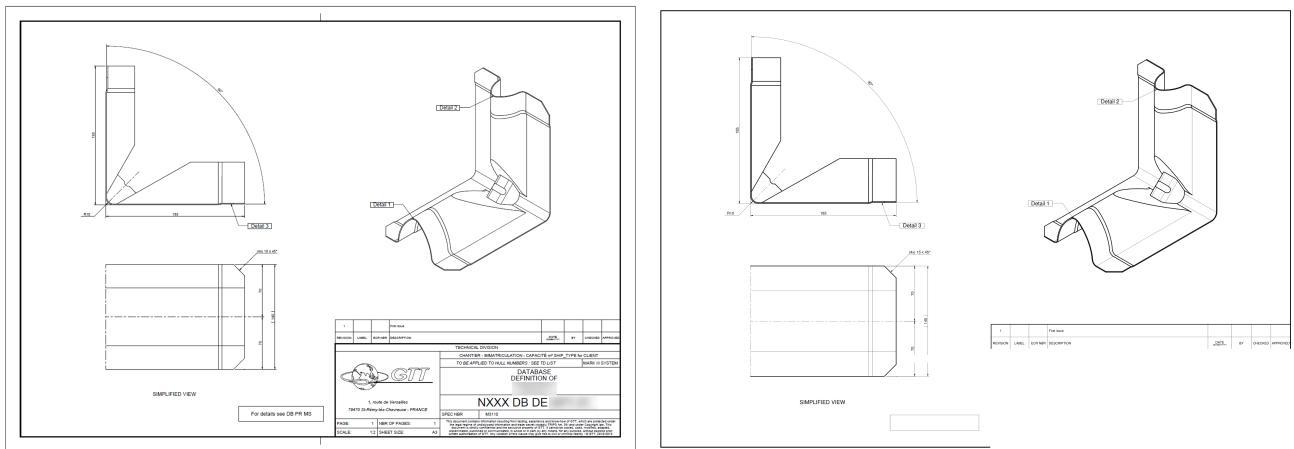


Figure 4.4.2 – Avant-après d'un plan auquel le processus de création du dataset a été appliqué. On observe qu'un variant pouvant aider le réseau de classification (« For details see DB PR MS ») a été masqué. Aussi, au-dessus du cartouche, une table reste : des modifications d'AutoCheck afin de mieux détecter les tables sont toujours en cours.

Du point de vue technique, le choix du stockage des labels s'est porté sur un encodage catégoriel mais en début d'entraînement, une conversion en « one-hot », également appelé 1 parmi n, est effectuée en fonction des classes d'entraînement choisies. Ceci permettant de ne pas limiter les classes sélectionnables lors de la création du dataset.

Un encodage one-hot consiste à encoder une variable catégorielle pouvant prendre n valeurs sur n bits. Le numéro du bit valant 1 correspondant à l'état de la variable.

Index	Couleur
1	Rouge
2	Bleu
3	Vert
4	Bleu

Table 4.4.1 – Encodage catégorielle

Index	Rouge	Bleu	Vert
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Table 4.4.2 – Encode one-hot

Table 4.4.3 – Exemple d'encodage one-hot à partir d'un encodage catégoriel

Au total, à partir des 84 projets, c'est plus 195 000 images labélisées qui ont été créées et qui vont constituer le dataset d'entraînement du modèle de classification. Comme nous avons généré des images à partir de tous les livrables historiques faisant partie du scope d'AutoCheck et au vu de leur nombre, nous pouvons affirmer que nos données sont **diverses**.

4.5 CRÉATION DU DATASET POUR LE RÉSEAU SIAMOIS

Afin que le réseau siamois remplisse son rôle, nous avons besoin de le finetune et pour cela nous avons besoin d'un dataset fait pour. Ce dernier doit-être composé de paires d'images ainsi que d'un label indiquant si les plans sont **similaires** c'est-à-dire identiques à un certains nombre de transformations invariantes près (translation, rotation, déplacement de légendes...) ou bien **différentes**.

De plus, nous souhaitons évaluer l'explicabilité de notre IA, et pour cela nous avons besoin d'être capable de localiser les différences et similarités (déplacement d'éléments n'entrant pas de différences du point de vue technique) au sein d'un plan afin de vérifier que l'IA se base bien sur cela pour prendre ses décisions. Il a été décidé que :

- Si le plan ne comporte que des différences, ou que des similarités, ces dernières seraient en rouge ;
- Dans le cas où un plan comporterait à la fois des similarités et des différences, les différences seraient en rouge et les similarités en bleu ;
- Enfin, seuls les éléments dont la position a changé seraient colorés ;

Ce jeu de données visant à vérifier l'explicabilité de notre modèle sera nommé pour la

suite « set de validation du réseau siamois ».

Ainsi, nous avons besoin de générer plusieurs sets de données :

1. Un set d'entraînement pour le réseau siamois composé d'exemples similaires et différents ;
2. Un set de validation pour le réseau siamois composé d'exemples similaires, différents et de cas limites, avec la localisation de tout changement ;

La notion de « cas limites » se référant ici des à des images similaires bien qu'impliquant de nombreuses transformations invariantes pour les plans.

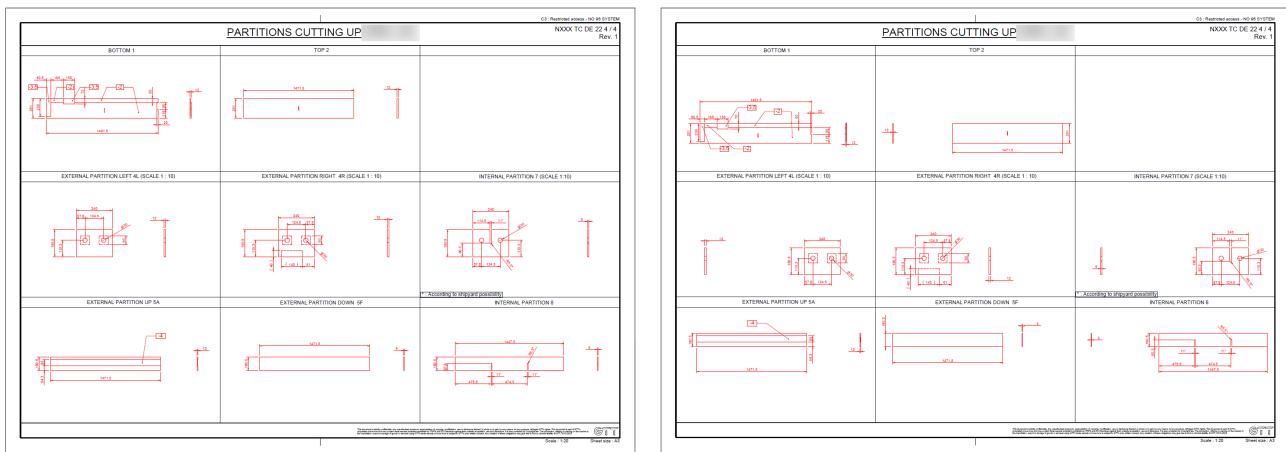


Figure 4.5.1 – Illustration d'un cas limite pour le réseau siamois : ces deux plans sont similaires : malgré de nombreuses transformations, ils représentent bel et bien le même plan

Il est important de noter qu'avec la méthodologie employée, nous aurons à disposition sur le set de validation les changements effectifs sur les images **au pixel près**, contrairement à si nous avions employé des bounding boxes. Notre métrique d'explicabilité sur le réseau siamois sera donc la plus précise possible.

4.5.1 CRÉATION DE PAIRES D'IMAGES DIFFÉRENTES

Il est extrêmement simple, en se basant sur le travail effectué lors de la Création du dataset de classification (section 4.4) de créer des paires d'images différentes. En effet, pour créer des paires d'images différentes, il suffit de prendre deux images du dataset de classification ayant au moins un label différent sur toutes les classes possibles.

Cependant, nous ne pouvons employer cette méthode pour obtenir des paires d'images similaires : si nous sommes certains qu'une différence de labélisation du dataset de classification résultera en effet sur des images différentes, la réciproque n'est pas vraie pour autant !

4.5.2 SUPERVISION D'UN LABELISATEUR

La complétion du dataset n'étant plus réalisable automatiquement, il a été décidé d'employer un labélisateur ayant des connaissances métiers.

L'été étant propice à de petits emplois saisonniers, ce fut donc le meilleur moment pour préparer la suite du projet à l'aide d'un ancien alternant ayant travaillé chez GTT. Ce fut mon rôle pendant 2 semaines : j'ai supervisé le travail d'un dessinateur qui devait créer les différents datasets évoqués.

Le dessinateur avait la connaissance métier et moi les attentes concernant les datasets. Je l'ai donc aiguillé notamment pour que les différentes classes et les différents sets soient bien équilibrés au niveau des livrables mais également en termes de quantités souhaitées.

A l'issue de ces deux semaines, 1 438 plans furent réalisés à partir de plans existants dont :

- 1 337 paires d'images similaires ;
- 49 paires d'images différentes ;
- 52 cas limites ;

Il est tout à fait logique que seules 49 et 52 paires d'images différentes et cas limites soient produites : en effet, avec le temps limité à disposition, la conception du set d'entraînement pour le réseau siamois a été privilégiée mais fut évidemment plus longue (avec la coloration et la réalisation de cas limites). De plus c'est une quantité suffisante pour élaborer une métrique d'explicabilité et en cas de dernier recours il restait la possibilité de créer davantage d'exemples en demandant à d'autres dessinateurs.

4.5.3 RÉCUPÉRATION DES PLANS & LABELS

La phase de récupération des plans et labels fut bien plus aisée concernant ce dataset étant donné que la localisation de chaque paire d'images ainsi que la labélisation associée

ont été inscrite dans un .csv lors de la labélisation. C'est un avantage de la constitution manuelle d'un dataset.

4.5.4 CRÉATION DES IMAGES & ZONES D'INTÉRÊT

La création des images et zones d'intérêts du réseau siamois fut assez simple car ressemblant énormément à ce qui a été fait précédemment (voir sous-section 4.4.3).

En se basant sur le *modus operandi* décrit précédemment, il a suffit de rajouter les étapes suivantes :

1. Déterminer les zones ayant été modifiés en se basant sur les différents canaux de couleurs de l'images ;
2. Créer et stocker les bounding boxes en conséquence ;
3. Remettre la couleur originale (noir) ;
4. Sauvegarder les images ;

A présent munis de nos deux datasets, concentrons-nous sur la mise en place de l'IA.

4.6 ENTRAINEMENT DE L'IA DE CLASSIFICATION

4.6.1 CHOIX DU MODÈLE

Le choix de notre modèle de classification s'est porté sur le Resnet, plus précisément sur le Resnet-50 comme utilisé dans [10].

Une réflexion sur l'utilisation de plusieurs autres modèles comme EfficientViT [16] ou EfficientNet [22] a été effectuée mais finalement il a été tranché que [10] ayant eu des résultats concluants avec un Resnet, nous allions tenter de reproduire ces résultats.

Pour la sortie de ce modèle, deux choix s'offraient à nous :

- Single-output : la sortie du modèle est unique et agrège les différentes classes
- Multi-output : le modèle possède plusieurs sorties (une par classe)

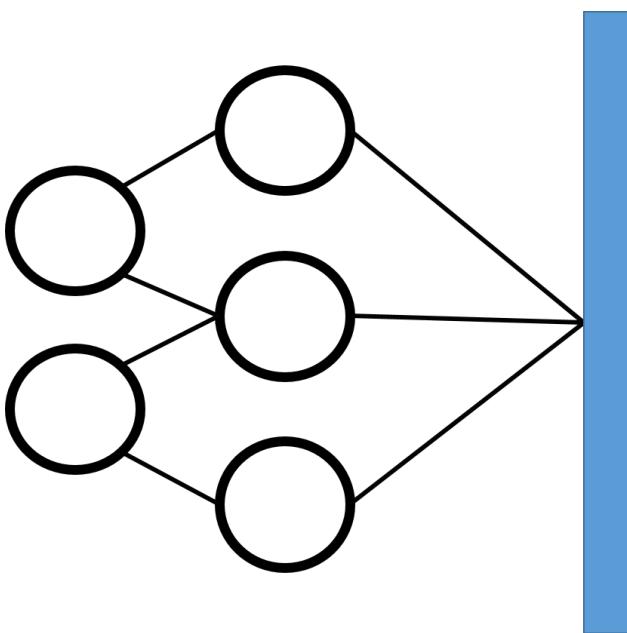


Figure 4.6.1 – Illustration d'un modèle de classification multi-label : il n'y a qu'une seule couche « fully connected » agrégeant toutes les classes

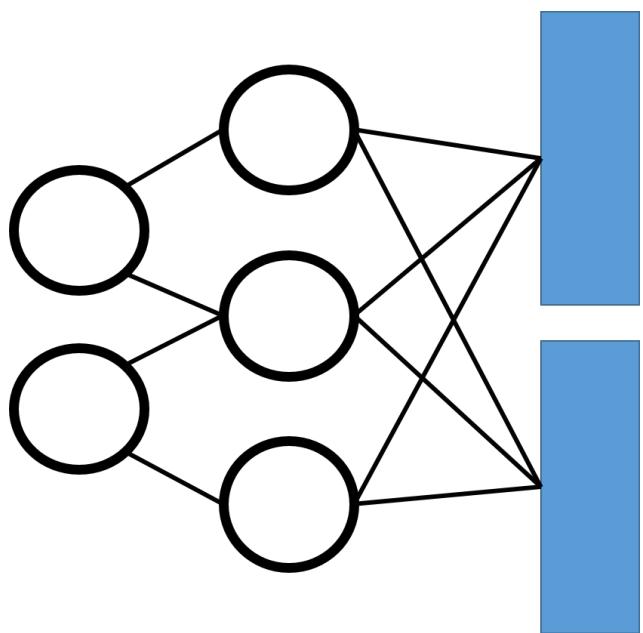


Figure 4.6.2 – Illustration d'un modèle de classification multi-output : il y a autant de couches « fully connected » que de classes

Nous avons choisi d'entrainer le modèle avec une unique sortie, en multilabel multiclass après discussions avec les équipes techniques de MP DATA.

4.6.2 UTILISATION D'UN CLUSTER DE CALCULS

Pour effectuer les différents entrainements, j'ai pu utiliser le cluster de calcul interne de GTT dénommé Jarvis, en référence à l'intelligence artificielle du même nom dans [Iron Man](#). Ce cluster de cluster est composé de 4 noeuds pour un total de 128 processeurs, 2 To de RAM et 8 cartes graphiques [NVIDIA V100 PCIe](#) chacune équipée de 16 GB RAM.

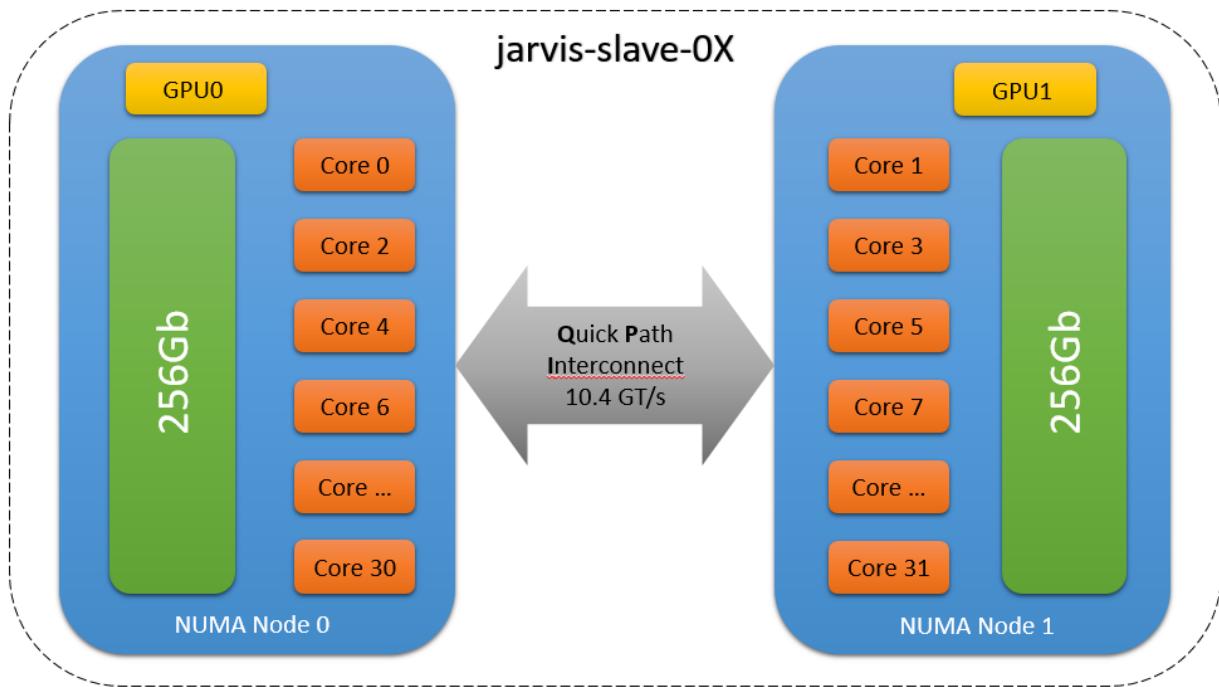


Figure 4.6.3 – Topologie d'un noeud de Jarvis

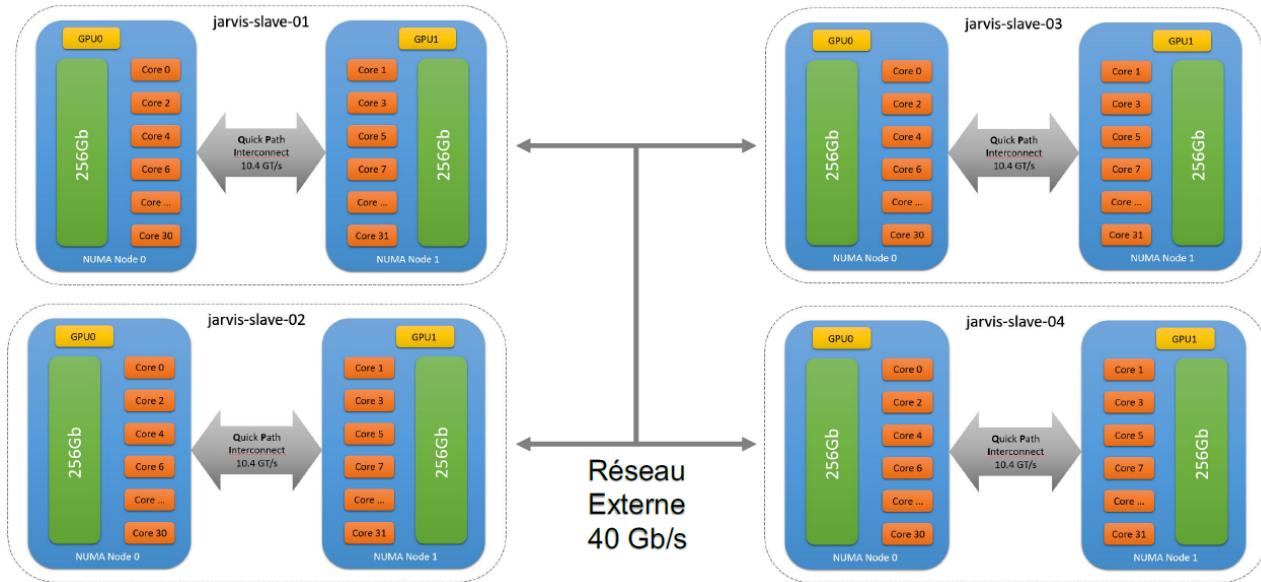


Figure 4.6.4 – Topologie de Jarvis dans son ensemble. Les 4 noeuds sont reliés par un débit de 40 Gb/s ce qui est supérieur aux 32 Gb/s maximum des V100 [9] : l'utilisation de plusieurs noeuds de calculs semble donc possible

Jarvis utilise deux programmes, Torque et MOAB, pour gérer l'activité du cluster. Plus précisément :

1. Torque est un système Portable Batch System (PBS) qui s'occupe du placement des jobs sur les noeuds de calculs, les démarrer, les arrêter et rapporter l'activité du cluster à MOAB ;
2. MOAB quant à lui planifie l'activité en fonction des demandes et de la politique d'accès aux ressources ;

On notera donc que Jarvis utilise PBS et non SLURM alors que ce dernier est normalement plus démocratisé dans le domaine de l'intelligence artificielle. Cela s'explique par le fait que la raison d'être du cluster est la réalisation de calculs Abaqus, nécessitant des processeurs et non des cartes graphiques.

Dans les faits, je suis la première personne à utiliser Jarvis pour entraîner des réseaux de neurones, bien que le cluster soit opérationnel depuis mai 2021. Ce qui, d'emblée, m'a confronté à de nombreuses difficultés cf :

- Prise en main (sous-section 5.3.1);
- Firewall (sous-section 5.3.2);
- Ordonnancement (sous-section 5.3.3);

4.6.3 STRATÉGIES DE PARALLÉLISATION

Lorsque l'on souhaite utiliser un cluster de calcul, il existe plusieurs manières d'entraîner une IA, afin de mobiliser plus ou moins toutes les ressources de calcul dudit cluster.

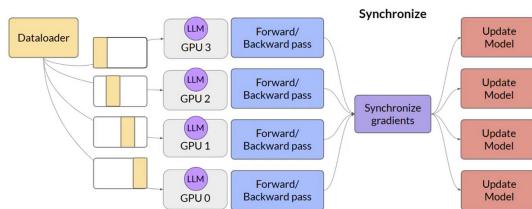
Notamment, nous avons essayé deux stratégies de parallélisation pour entraîner nos modèles, à savoir :

1. Distributed Data Parallel (DDP) : où chaque GPU utilisé va héberger une copie du modèle que l'on souhaite entraîner. Le dataset est divisé selon le nombre d'unités de calculs utilisées de sorte que chaque copie du modèle ne verra qu'une partie du dataset tout au long de l'entraînement. A chaque passe, les résultats sont agrégés afin de mettre à jour les poids de toutes les copies du modèle. Cette stratégie demande d'avoir suffisamment de VRAM sur chaque GPU pour pouvoir

héberger le modèle complet dessus, mais est autrement efficace et relativement facile à mettre en place ;

2. Fully Sharded Data Parallel (FSDP) : où l'on va répartir les différentes couches du modèles sur différents GPU. De plus, un transfert sur le CPU de certains résultats intermédiaires (concernant les optimiseurs, gradients, paramètres du modèle et activations) peut-être ajouté. Cette stratégie est utilisée notamment lorsque les modèles demandent une utilisation de VRAM plus importante que ce que possèdent les cartes graphiques ;

Distributed Data Parallel (DDP)



Fully Sharded Data Parallel (FSDP)

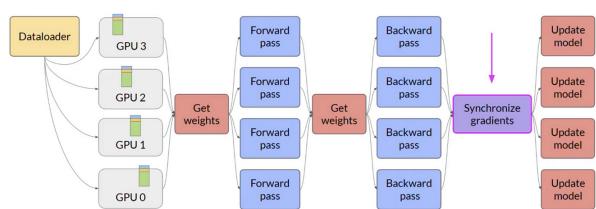


Figure 4.6.5 – Fonctionnement de la Distributed Data Parallel

Figure 4.6.6 – Fonctionnement de la Fully Sharded Data Parallel

Dans notre cas, nous avions commencé par nous orienter vers la DDP avant de nous rendre compte que les cartes graphiques à disposition ne possédaient pas assez de VRAM pour traiter des images de taille 2272×1576 .

Il est possible de trouver la quantité de VRAM nécessaire pour entraîner notre modèle en DDP. Le détail de ce calcul a été décrit Appendice B et donne le résultat suivant :

$$mémoire_{VRAM} = 34 \quad [\text{Gb}] \quad (4.2)$$

Il existe [d'autres stratégies de distribution](#) mais demandant soit des modifications de codes plus poussées que je n'avais pas le temps d'implémenter soit des versions de Pytorch / CUDA que nous ne pouvions pas utiliser pour des raisons de compatibilités évoquées sous-section 5.3.4.

Après avoir longuement essayé la FSDP, nous avons au dernier moment changé d'avis pour utiliser la DDP avec des images de taille 512×512 , afin de pouvoir mener à bien l'implémentation de certaines métriques comme cela sera détaillé section 5.5.

4.6.4 CHOIX DES PARAMÈTRES D'ENTRAINEMENT

CLASSES NON-ÉQUILIBRÉES

Le dataset de classification possédait un fort déséquilibre : si certaines classes ne comportaient que quelques valeurs et bien distribuées (*exempli gratia* : 2 valeurs réparties presque équitablement pour la technologie), d'autres comportaient plusieurs dizaines voire centaines de valeurs dont certaines n'étaient représentées que quelques dizaines de fois (comme les éléments ou leurs références).

Nous avons pondéré la fonction de perte pour prendre en compte cela comme détaillé sous-section 5.4.3.

LOSS FUNCTION

La fonction de perte utilisée pour entraîner notre modèle est la [BCELossWithLogits](#) possède l'avantage d'être numériquement stable et performante comparé à la simple utilisation d'une BCELoss.

D'autres fonctions de perte, comme la Focal Loss ont été envisagées, notamment afin de remédier au problème de déséquilibre des classes.

MÉTRIQUE

Étant donné le cas d'usage qu'il sera fait du modèle, nous avons choisi d'utiliser le F_{β} -score avec $\beta = 2$ afin de mesurer l'efficacité de nos modèles.

$$F_{\beta}\text{-score} = \frac{(1 + \beta^2) \times \text{précision} \times \text{rappel}}{(\beta^2 \times \text{précision}) + \text{rappel}} \quad (4.3)$$

$$F_{\beta}\text{-score} = \frac{(1 + \beta^2) \times TP}{(1 + \beta^2) \times TP + \beta^2 \times FN + FP} \quad (4.4)$$

Avec

- TP : Vrais-positifs ;
- FP : Faux-positifs ;
- FN : Faux-négatifs ;

Ce choix n'est pas anodin car plus cette métrique est élevée et plus le nombre de faux-négatifs est bas. Or, étant donné qu'une erreur non-détectée peut avoir des conséquences désastreuses pour GTT, il vaut mieux que notre modèle détecte toutes les erreurs, quitte à ce que certaines n'en soient en réalité pas !

SCHEDULER

Différents scheduler, permettant de choisir l'évolution de certains paramètres en fonction des résultats rencontrés au fil de l'entraînement, ont été mis en place, ceci permet un entraînement plus ergonomique et automatique. Ont notamment été mis en place :

1. Une mise à jour du *learning rate* en cas de non-amélioration du f_{β} _score à l'aide d'un [ReduceLROnPlateau](#) ;
2. Un arrêt total de l'entraînement une fois un minimum global durablement atteint à l'aide d'un EarlyStopping ;

AUTRE

Par mesure de précaution, les poids du modèle sont sauvegardés en fonction de différentes métriques afin d'être sûr de pouvoir retrouver l'état du modèle nous intéressant le plus. Ainsi, pour à chaque entraînement, étaient sauvegardés les poids correspondants :

1. A la dernière epoch ;
2. Au meilleur f_{β} _score atteint ;
3. Au meilleur rappel atteint ;
4. Au meilleure f_1 _score atteint (pour comparer avec le f_{β} _score) ;
5. A la meilleure précision atteinte ;

4.7 EXPLICABILITÉ DE L'IA & MÉTRIQUES UTILISÉES

Comme évoqué dans section 3.4, l'explicabilité de l'IA est primordiale car elle faisait partie des objectifs de la mission et permettait de contrôler la pertinence des modèles. C'est pour cela que j'ai mis en place, que ce soit pour le réseau de classification ou le réseau siamois, divers outils contribuant à l'interprétation des résultats des modèles et d'essayer de comprendre pourquoi ils commettent des erreurs afin de les corriger.

Dans cette section, nous nous contenterons d'expliquer les principes de base des différents outils et métriques utilisés afin qu'ils aient été introduits pour Résultats (chapitre 6)

4.7.1 MATRICE DE CONFUSION

En Machine Learning, une matrice de confusion sert à mesurer la qualité d'un système de classification. Les lignes correspondent aux classes réelles et les colonnes aux classes prédites. L'avantage de cet outil réside dans sa lisibilité : il est simple de savoir si un classifier est performant ou non et, s'il ne l'est pas, de voir où il se trompe.

Dans notre cas, j'ai au sein de notre classification multilabel, généré les matrices de confusions au sein de nos différentes classes. Ceci, afin de suivre en détails les performances de notre modèle au fil des epochs.

4.7.2 COURBE DE PRÉCISION-RAPPEL

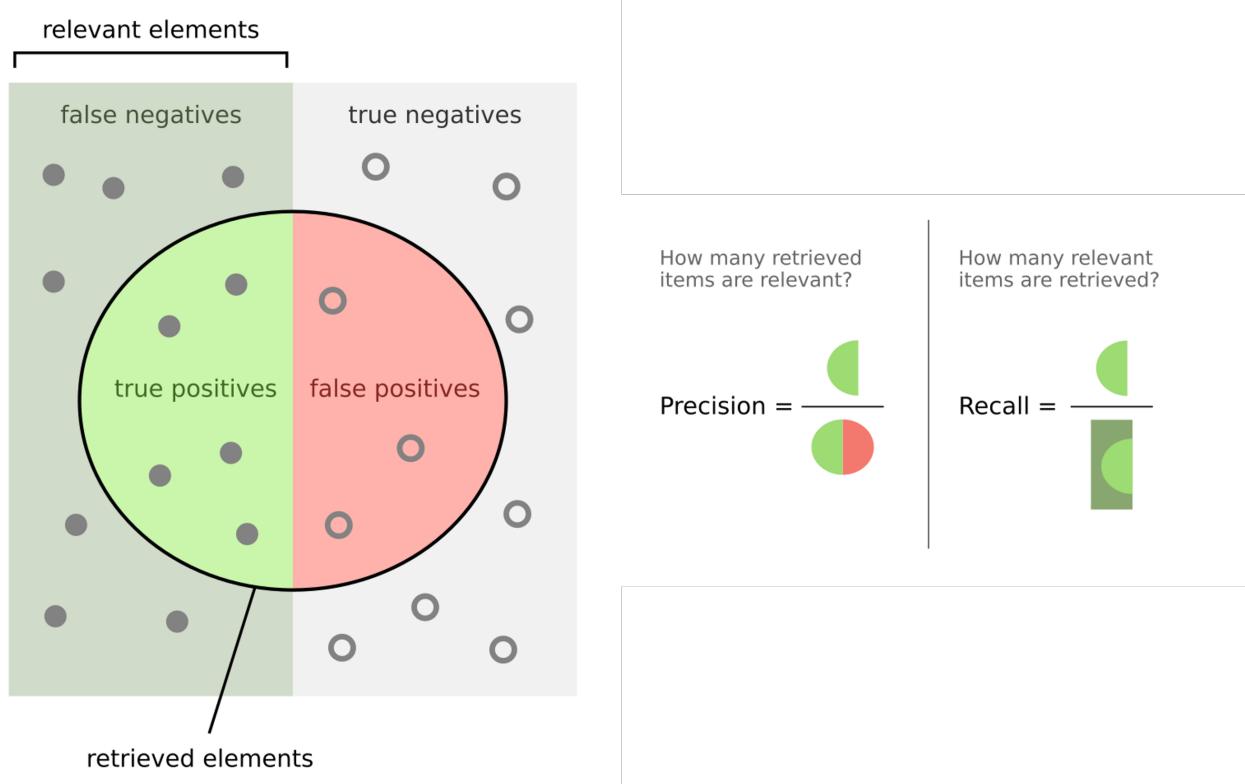


Figure 4.7.1 – Explication de la précision et du rappel

La précision et le rappel constituent deux métriques couramment utilisée en classification :

- La précision correspond au taux de prédictions correctes parmi les prédictions positives. Ainsi on a : $\text{précision} = \frac{n_{\text{vrais positifs}}}{n_{\text{vrais positifs}} + n_{\text{faux positifs}}}$
- Le rappel (également appelé « sensibilité ») correspond au taux d'individus positifs détectés par le modèle et est donné par la formule $\text{rappel} = \frac{n_{\text{vrais positifs}}}{n_{\text{vrais positifs}} + n_{\text{faux négatifs}}}$

Penchons-nous sur le fonctionnement d'un modèle de classification. Ce dernier attribut pour un échantillon une probabilité d'appartenance pour chaque label i. Un seuil de classification s_i est utilisé pour transformer une probabilité p_i en appartenance prédite. De

sorte que pour tout label i , on ait :

$$\begin{cases} L'\text{échantillon appartient au label } i, & \text{si } p_i \geq s_i \\ L'\text{échantillon n'appartient pas au label } i, & \text{sinon} \end{cases}$$

En fonction du choix de ces seuils, les résultats du modèle peuvent donc grandement varier et notamment, lorsque les seuils augmentent, la précision a tendance à augmenter et le rappel à baisser. Il y a donc un compromis à trouver et c'est cela que permet la courbe de précision-rappel, qui trace la précision en fonction du rappel pour différentes valeurs de seuil.

Ainsi, la courbe de précision-rappel nous permet donc de trouver les meilleures valeurs de seuils pour nos différents labels afin de trouver le meilleur compromis entre précision et rappel en fonction de notre cas d'utilisation. Dans notre cas, souhaitant minimiser le nombre d'ADC, c'est donc le rappel que l'on cherchera à maximiser.

4.7.3 SALIENCY MAP

Comme nos réseaux utilisent des images, et notamment le réseau siamois qui doit repérer des différences entre deux images, il apparaît utile de savoir les zones de l'image sur lesquelles se basent nos réseaux pour établir leur classification.

Cela est rendu possible par les Saliency Map, qui produisent une heatmap qui permet de visualiser le degré d'importance de chaque pixel grâce à différentes techniques ou algorithmes.

Parmi les différentes possibilités, le choix s'est porté sur l'utilisation des « Gradient Class Activated Maps » (Grad-CAM), qui produisent une heatmap grâce à la somme pondérée de l'activation de la couche convolutionnelle par les gradients [20]. Pour une même image, la grad-CAM de différents labels peuvent être totalement différentes.

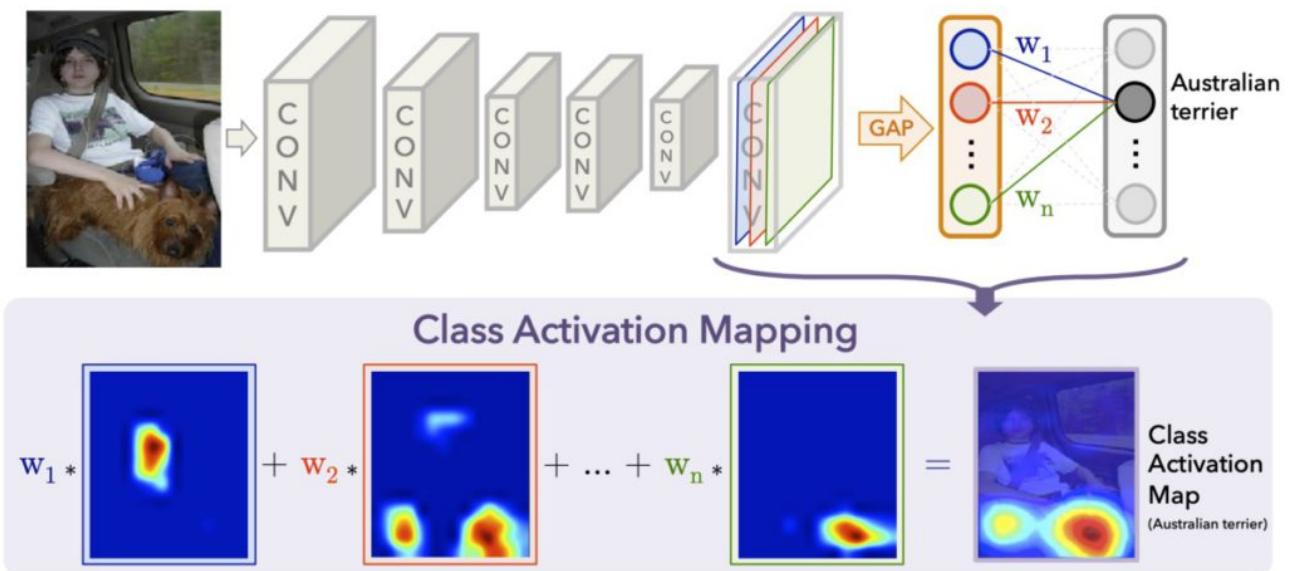


Figure 4.7.2 – Illustration de l’obtention des Class Activation Map dans un CNN [27]

Le choix de la saliency map n'est pas anodine et dépend de plusieurs facteurs : structure du réseau utilisé, doit se placer sur une couche précise et sa qualité doit être quantifiée (cf sous-section 7.2.1). Dans notre contexte, étant donné le choix du modèle de classification, des résultats empiriques et exemples d'utilisations convergeaient vers l'utilisation de la Grad-CAM sur la 4ème couche du Resnet [27].

A partir des heatmap obtenues, superposées aux images, il est aisément de localiser les zones de l'image ayant menées à la décision du modèle.

Dans notre cas, les grad-CAM ont été implémentées pour le réseau de classification afin de mieux comprendre comment ce dernier attribuait les labels parmi la centaine de possibilités mais surtout sur le réseau siamois pour les deux raisons suivantes :

1. Pour pouvoir mesurer la performance du modèle à partir du set de validation créé section 4.5 ;
2. Pour le confort des utilisateurs, en leur montrant directement où se trouvent les différences entre les deux images ;

4.7.4 EMBEDDINGS DU MODÈLES

Un réseau en Deep Learning passe par une phase de représentation vectorielle des données d'entrées : c'est ce que l'on nomme embeddings.

Les embeddings constituent un élément clé de nombreux domaines en Machine Learning. Ils permettent de mesurer la similarité entre plusieurs éléments avec une mesure comme la distance euclidienne et permettent également de facilement visualiser ces dernières une fois projetés en 2 ou 3 dimensions.

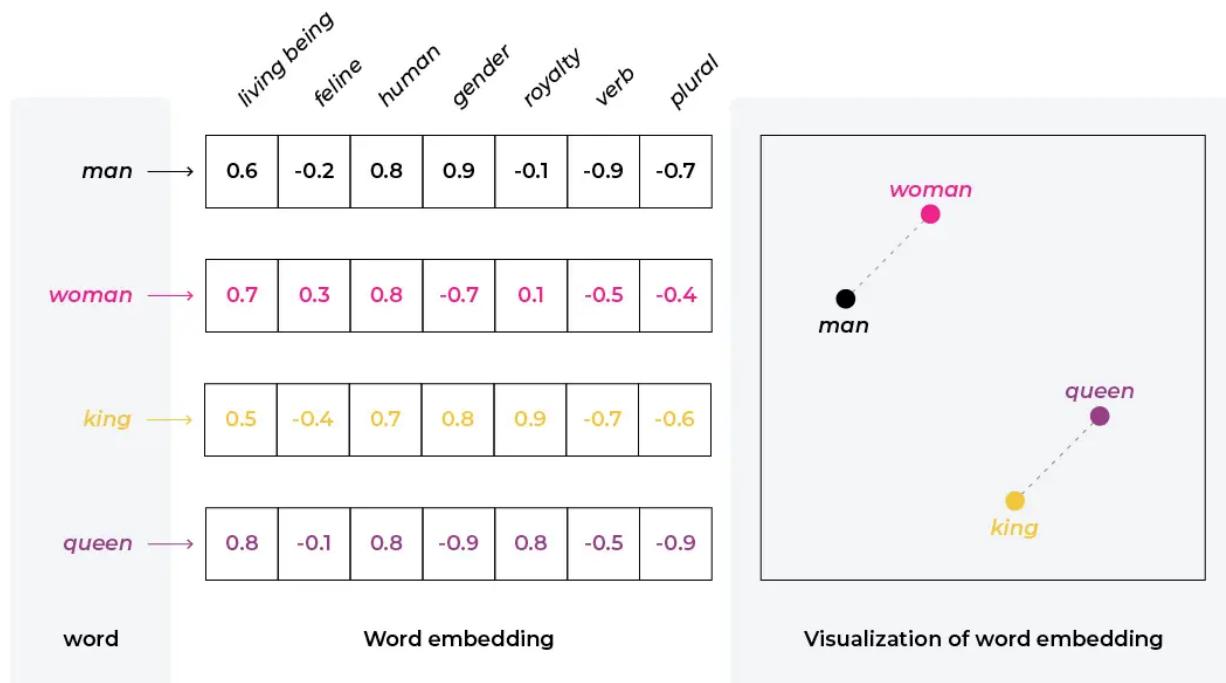


Figure 4.7.3 – Exemple connu de représentation et de mesure de similarités entre des mots en Natural Language Processing

Dans le cadre de notre fausse tâche intermédiaire, nous souhaitons obtenir les meilleurs embeddings possibles, c'est donc tout naturellement que nous avons implémenté la visualisation de ces derniers malgré quelques problèmes sous-section 5.5.1.

4.7.5 HISTOGRAMME DES POIDS & BIAIS

Le fonctionnement d'un neurone constitue le fondement du Machine Learning, en s'inspirant du fonctionnement des neurones biologiques qui fournissent une sortie en fonction d'une entrée et dont le grand nombre permet l'émergence de comportement complexes par le biais d'interactions qui nous dépassent.

Cela dit, en ML, le fonctionnement est plutôt simple car il s'agit de « simples » additions et multiplications :

$$output = input \times W + b \quad (4.5)$$

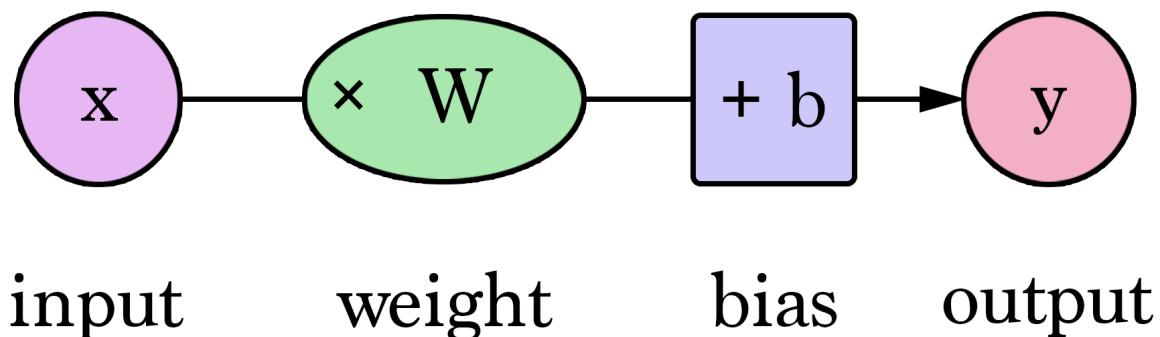


Figure 4.7.4 – Fonctionnement d'un neurone en Machine Learning

La visualisation des poids et biais des différentes couches du réseau permet d'observer leur distribution et peut permettre de déceler par exemple un grand nombre de neurones « morts », signifiant que le dimensionnement du modèle n'est pas correct et qu'une étape de pruning semble être nécessaire.



Figure 4.7.5 – Exemple de distribution de poids & biais au sein d'un réseau neuronal n'apprenant vraiment semblablement pas

Pour des raisons évoquées sous-section 5.5.2, cette visualisation n'a finalement pas été fonctionnelle.

Tout ce qui est susceptible d'aller mal ira mal.

- Edward A. Murphy Jr

5

Difficultés rencontrées

Dans ce chapitre, sont rassemblées les différentes difficultés que j'ai pu rencontrer lors de mon stage.

5.1 DIFFICULTÉS LIÉES AUX TESTS LOGICIELS

Les attentes concernant les tests étant d'abord simples, je les ai implémenté en utilisant [unittest](#), un framework de test implémenté nativement dans Python.

Cependant, les besoins des tests ont évolué au fur et à mesure de leurs utilisations, il a donc été jugé préférable de coder une classe de tests sur laquelle nous aurions un contrôle plus fin. C'est donc ce que j'ai fait, et techniquement ce que nous aurions dû faire dès le départ si nous avions mieux cerné les besoins, par exemple en établissant un cahier des charges.

Retour Véracité des plans (sous-section 4.4.1)

5.2 DIFFICULTÉS LIÉES À LA CRÉATION DES DATASETS

5.2.1 EXTRACTION DES PLANS

Extraire la quasi- totalité des plans créés par GTT au cours de ces 10 dernières années fut laborieux pour plusieurs raisons :

1. La documentation de PDM est inexacte et incomplète : toutes les versions ne sont pas répertoriées et des fonctionnalités de certaines versions sont non-fonctionnelles. Ceci a rajouté du temps de développement ;
2. Notamment, les fonctions de l'API censées permettre l'extraction des « datacards » n'étaient pas utilisables, nous avons cependant pu les extraire sous format .csv pour ensuite les intégrer à la labélisation effectuée par AutoCheck
3. Le code n'effectuait qu'une requête toutes les 15s, ce qui au vu du nombre de données à télécharger aurait pris environ 220 jours. Diagnostiquer le problème à l'aide de la DSI n'aboutissant pas, un code VBA temporaire a été réalisé pour l'extraction. Finalement, contrairement à ce que les faits laissaient penser, la cause était une mauvaise configuration de la VM, sur laquelle nous n'avions pas la main. Les différentes tentatives de résolution ainsi que la mise en place d'une seconde solution fut chronophage ;

Retour Récupération des plans (sous-section 4.4.2)

5.3 DIFFICULTÉS LIÉES AU CLUSTER

5.3.1 PRISE EN MAIN

Tout d'abord, l'accès au cluster de calcul de l'entreprise a été tardif, n'ayant pu y accéder qu'à partir de la moitié du stage.

Aussi, comme évoqué dans Utilisation d'un cluster de calculs (sous-section 4.6.2), j'ai été la première personne à utiliser Jarvis pour effectuer des calculs utilisant les cartes graphiques du cluster, ce qui implique qu'il y avait peu de ressources à disposition pour m'aider.

Notamment la documentation utilisateur du cluster sur l'utilisation des GPU était inexistante et sans exemples à disposition, il a fallu que je me forme et questionne l'administrateur, ce qui a pris du temps.

En plus de cela, il m'a fallu résoudre de nombreux soucis de compatibilité liés aux différentes versions des drivers NVIDIA et CUDA par rapport aux éléments de Pytorch que je souhaitais utiliser, les versions du cluster datant s'il y a 3 ans et une mise à jour de ces derniers n'étant pas souhaitée par l'administrateur.

Finalement, mon retour d'expérience a été utile à l'entreprise car cela a permis l'étoffement de la documentation interne ainsi que la mise en place d'exemples fonctionnels pour des futurs utilisateurs de Jarvis souhaitant effectuer du Machine Learning.

Retour Utilisation d'un cluster de calculs (sous-section 4.6.2)

5.3.2 FIREWALL

La DSI a oublié de renouveler certaines de ses licences, notamment concernant les pare-feu. Ces derniers définissent la politique de sécurité des réseaux en surveillant et contrôlant les flux applicatifs et de données.

Ainsi, Jarvis s'est retrouvé avec une configuration de pare-feu radicale : aucun package python n'était autorisé à être installé. Ce qui s'est évidemment répercuté sur des tests de bon fonctionnement du code produit lors de l'ajouts de nouveaux packages...

Retour Utilisation d'un cluster de calculs (sous-section 4.6.2)

5.3.3 ORDONNANCEMENT

L'algorithme d'ordonnancement de Jarvis n'était pas adapté à l'utilisation qui était faite des GPU. En effet, en ML, peu de CPU sont nécessaires car la seule utilité de ces derniers est de charger et transformer les données avant de les transférer sur les GPU, et cela est réalisable facilement avec un nombre restreint de CPU car le temps qu'ils ont pour effectuer ces actions sur un batch correspond au temps que prend les GPU pour traiter le batch précédent.

Or, le système de queue de Jarvis détaillé Appendice C ne possédait pas de file pour cela et il fallait faire un compromis entre les files short, long et big :

- En choisissant la file short, le job avait une meilleure priorité mais rentrait en concurrence avec les jobs Abaqus de par l'utilisation de 8 CPU qui n'étaient pas forcément disponibles. De plus, le temps maximal alloué (18 minutes) n'était pas compatible à la réalisation d'un entraînement mais uniquement à tester du code ;
- En choisissant la file long, il n'y avait pas de minimum de ressources à utiliser mais la priorité du job était inférieure à ceux des autres files ;
- En choisissant la file big, l'entraînement était sûr de tourner mais cette file n'était utilisable que le week-end ;

Le cas de figure où un noeud possédait moins de 8 CPU disponibles et où mon calcul aurait pu tourner sans les obligations de ressources imposées par l'ordonnanceur s'est produit de nombreuses fois comme le montre l'exemple Appendice E.

Retour Utilisation d'un cluster de calculs (sous-section 4.6.2)

5.3.4 STRATÉGIES DE PARALLÉLISATION

L'utilisation de stratégies de parallélisation pour effectuer nos entraînement sur Jarvis a été source de quelques difficultés.

En effet, il a d'abord fallu me former en autonomie à cela, car cela ne faisait pas partie des enseignements au sein d'IMTA mais également car personne au sein de GTT ou MP DATA n'avait ces connaissances.

Aussi, le fait que le cluster utilise PBS et non SLURM a rendu l'utilisation des différents noeuds de calculs impossible dans le cadre du stage, étant donné que des configurations existent sur SLURM mais que pour PBS il fallait les faire. Au vu du temps à disposition, je n'avais pas le temps de me documenter pour effectuer cette implémentation.

Ainsi, les ressources à disposition pour les calculs se limitaient à un seul noeud du cluster, à savoir 2 cartes graphiques **NVIDIA V100** de 16 Gb.

Retour Stratégies de parallélisation (sous-section 4.6.3)

5.4 ENTRAINEMENT

5.4.1 NOMBRE D'ENTRAINEMENTS

Comme expliqué sous-section 5.3.3, étant donnée la configuration de Jarvis et notamment de ses algorithmes d'ordonnancement (cf Appendice C), la seule manière d'effectuer un entraînement de manière sûre était d'utiliser la file « big ». Or cette dernière ne fonctionnait que le week-end.

En prenant en compte la durée du stage, ce nombre d'entraînement s'élève à 24.

A cela il faut soustraire le temps de de mettre en place les tests et de créer le dataset, ainsi que la résolution des divers problèmes liés à Jarvis ainsi que la mise en place du code.

Ainsi, nous n'avions en pratique qu'une dizaine, si ce n'est moins, fenêtres d'entraînement.

5.4.2 TEMPS D'ENTRAINEMENT

Le temps d'entraînement fut également une énorme difficulté pour mener à bien les objectifs du stage. Ce dernier est fonction de nombreux facteurs : modèle choisi, taille des images, carte graphiques à disposition...

De par les contraintes en jeu ainsi que les choix effectués et la taille du dataset, le temps d'entraînement pouvait varier 2h30 à plus de 8h par epoch.

5.4.3 ENTRAINEMENTS NON-PROBANTS

APPARITION DE LOSS LIÉE AU POS_WEIGHT

Pour tenter de contrebalancer le déséquilibre des classes, nous avons tenté d'utiliser l'argument *pos_weight* de notre [fonction de loss](#), dont le principe est d'accorder un poids plus grand aux classes les moins représentées dans le calcul de la loss. Chaque label se voit assigné un poids pos_i tel que :

$$pos_i = \frac{n_{\text{exemples négatifs}}}{n_{\text{exemples positifs}}} \quad (5.1)$$

Mais ceci produisait des disparités énormes : les poids allant de 10^{-2} à 10^5 . Ces poids bien trop grands étaient source de NaN lors des étapes de backpropagation des entraînements.

Pour résoudre ce problème, nous avons essayé d'attribuer un poids total W_i à chaque classe, de sorte que :

$$pos'_i = pos_i \times \frac{W_i}{\sum_j pos_j} \quad (5.2)$$

$$W_i = \sum_i pos'_i \quad (5.3)$$

Mais cela fut sans succès et causa au contraire une disparition des gradients, les poids allant de 10^{-6} à 10^{-1} , n'ayant pas réussi à trouver des valeurs appropriées pour les W_i .

Une nouvelle solution fut de repartir de l'Équation 5.1, d'appliquer un logarithme et de déplacer le minimum à 1; afin de prendre en compte le déséquilibre de classe tout en évitant les valeurs aberrantes :

$$pos''_i = \log(pos_i) - \min_i \log(pos_i) + 1 \quad (5.4)$$

APPARITION DE LOSS LIÉE À LA PRÉCISION

Nous avons été longuement bloqués par l'apparition de NaNs liée à la précision machine :

- En float32 : Aucun NaN
- En float16 : Apparition de NaN

Après avoir disséqué notre modèle, ses gradients et la mise à jour de ses poids, le changement d'optimiseur d'un Adam à un SGD résolu ce problème.

En effet, nous tenions à utiliser une précision en float16 afin de gagner du temps sur des entraînements déjà bien longs.

APPARITION D'ERREURS

Les problèmes liés aux visualisations (sous-section 5.5.1 & sous-section 5.5.2) ont été source d'erreurs mettant fin à des entraînements alors que nous en avions déjà qu'un nombre limité.

Retour Choix des paramètres d'entraînement (sous-section 4.6.4)

5.5 DIFFICULTÉS LIÉES AUX MÉTRIQUES

5.5.1 EMBEDDINGS

La visualisation des embeddings n'est vraiment utile qu'une fois le modèle entraîné, car la manière dont le modèle aboutit à ces derniers va évoluer au fil de l'entraînement. Pendant l'entraînement, il est plus pratique de s'intéresser aux matrices de confusion, ces dernières demandant bien moins de ressources et de temps pour être calculées et étant plus facilement lisibles.

Le code produit avait été pensé de sorte d'utiliser un « Early Stopper », qui permet de finir l'entraînement une fois que la métrique choisie ne s'améliore plus. Une fois ce dernier signalant la fin de l'entraînement, le calcul des embeddings devait se faire.

Cependant, du à l'implémentation de la FSDP dans [Pytorch Lightning](#) et sa manière de stocker les poids du modèle efficacement, il n'était pas possible de calculer proprement les embeddings en fin d'entraînement. Nous avons créé, sans réponse à ce jour, une [issue Github à ce propos](#).

Dans l'attente d'une réponse, nous avions mis cette visualisation de côté malgré un code fonctionnel en théorie, mais il a finalement fallu l'intégrer de manière moins efficace et pratique dans le code.

Retour Embeddings du modèles (sous-section 4.7.4)

5.5.2 HISTOGRAMME DES POIDS & BIAIS

Pour des raisons similaires à la visualisation des embeddings, la visualisation des histogrammes des poids a été retardée, malgré l'existence d'un code théoriquement fonctionnel, avant d'être abandonnée après une évaluation de son utilité par rapport au coût de ré-implémentation.

[Retour Histogramme des poids & biais \(sous-section 4.7.5\)](#)

Il n'y a pas de réussite facile, ni d'échecs définitifs

- Marcel Proust

6

Résultats

6.1 RÉSEAU DE CLASSIFICATION

Au terme du stage, un entraînement avec les labels suivant a été probant :

- Technologie ;
- Zone spéciale ;
- Type de plan

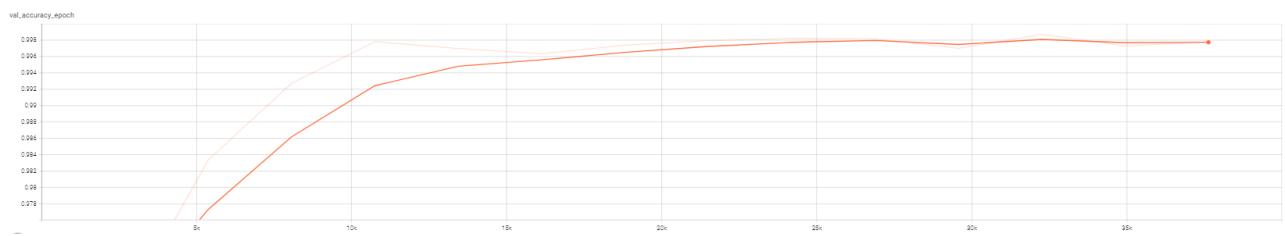


Figure 6.1.1 – Évolution de la précision sur le set de validation

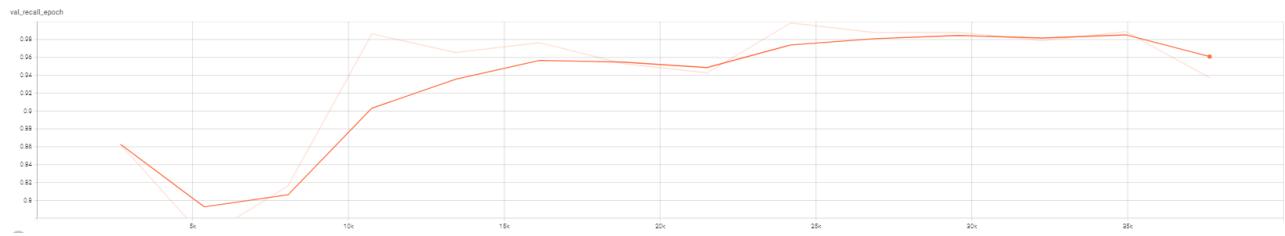


Figure 6.1.2 – Évolution du rappel sur le set de validation

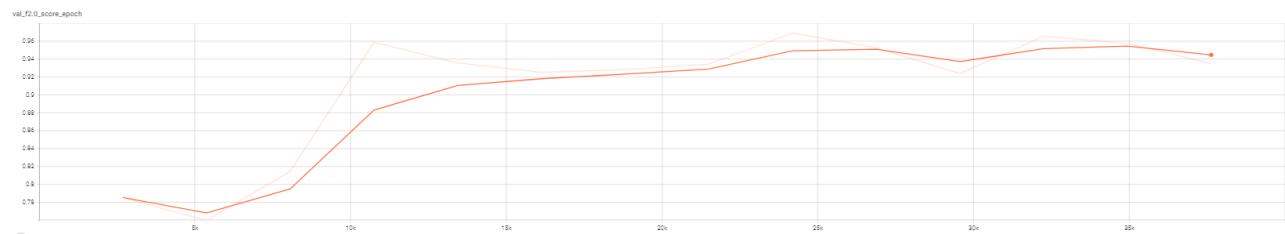


Figure 6.1.3 – Évolution du f_2 _score sur le set de validation



Figure 6.1.4 – Évolution du learning rate

Figure 6.1.5 – Résultat d'un entraînement sur 14 epochs

On observe de très bons résultats avec, à la 13^{me} epoch :

- Un f_2 _score de 0.955;
- une précision de 0.997;
- Un rappel de 0.989;

Et cela avec un learning rate qui n'a pas changé : la convergence ayant été plus rapide

qu'anticipée pour que le ReduceLROnPlateau fasse effet.

Des soucis d'écriture concurrentielle sur le tensorboard, liés au changement de dernière minute de stratégie, nous obligent à regarder les matrices de confusion binaires au lieu des matrices de confusion par classe et expliquent les mauvaises visualisations liées à la gradcam.

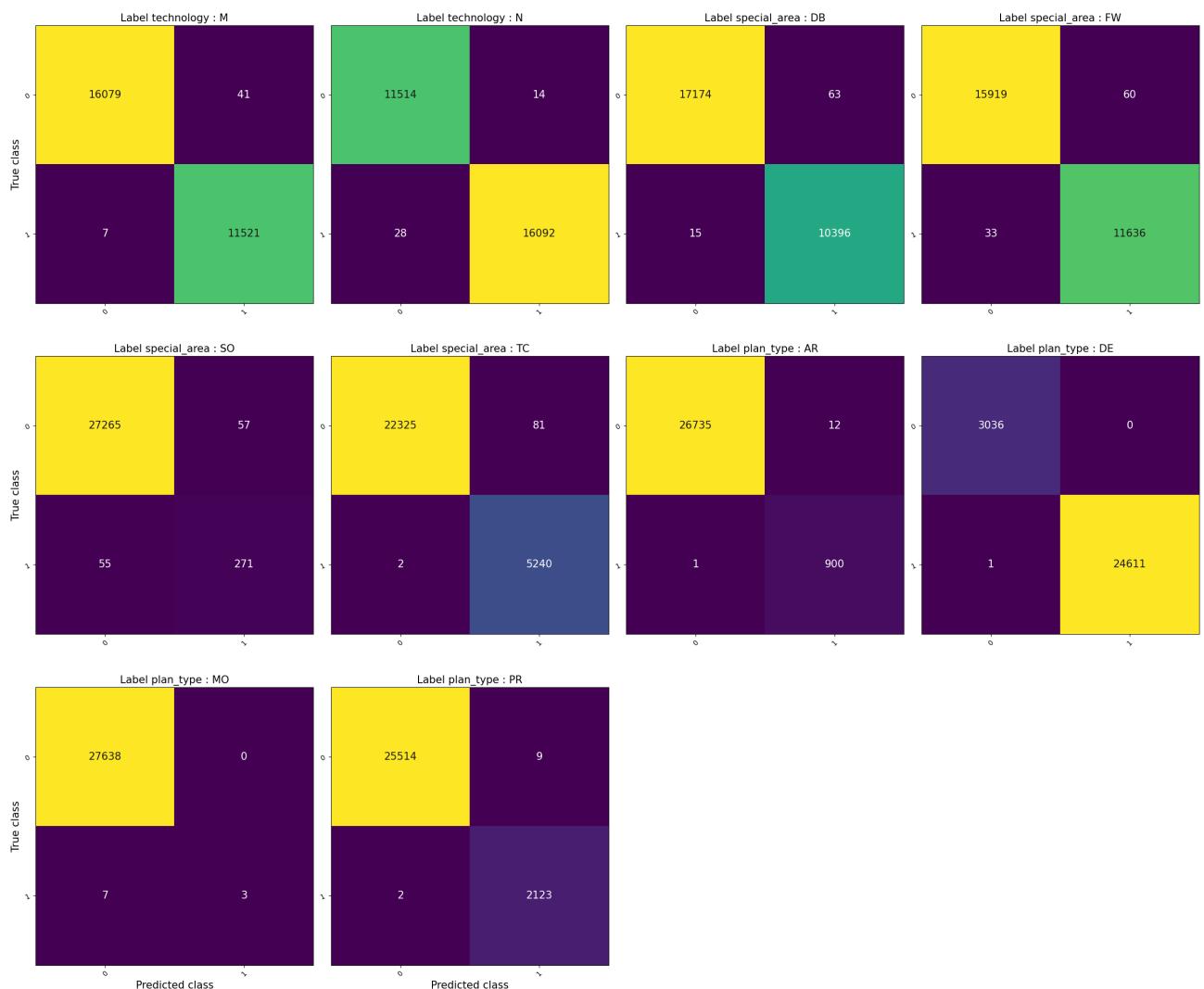


Figure 6.1.6 – Matrice de confusion binaire du set de validation

On observe que les matrices de confusion reflètent le résultats des métriques : elles sont

très bonnes, étant quasiment diagonales. Cette visualisation pourrait être améliorée en normant les éléments de la matrice pour rendre compte de cela.

Le déséquilibre des données se reflète bien dans ces différentes matrices, notamment par le nombre prédominant de plans de définition (DE) et au contraire le peu de plan de type MO (une dizaine, mais cela est lié au fait qu'il n'y en a qu'un par projet !)

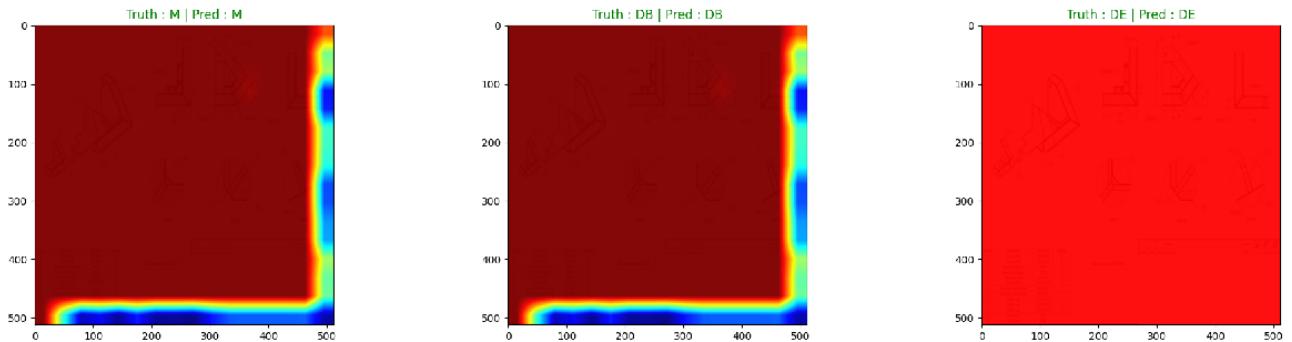


Figure 6.1.7 – Gradcam des différentes classes d'un plan correctement prédit

Le problème d'écriture concurrentielle mentionné ci-dessus est parfaitement illustré par les images obtenues : malgré qu'elles doivent être composées à 90% du plan de base et de 10% des gradcams, elles apparaissent quasi-totalement rouge, ce qui n'est pas normal.

Ces résultats sont très encourageants pour la suite mais ont été obtenus à la suite d'un entraînement « simple » sur une fraction des labels.

La prochaine étape consiste résoudre les problèmes d'écriture concurrentielle et d'effectuer un entraînement prenant en compte l'échelle des plans ainsi que les éléments, si cela s'avère possible également leurs références et enfin si AutoCheck parvient à récupérer le titre des pages, regénérer le dataset et ajouter cette donnée aux labels.

6.2 RÉSEAU SIAMOIS

Par manque de temps, aucun résultat relatif au réseau siamois n'a pu être produit.

Cependant, le dataset nécessaire à son entraînement a été réalisé et il ne manquerait réellement qu'un « bon » entraînement du réseau de classification pour continuer ce travail.

Un échec est un succès si on en retient quelque chose

- Malcom Forbes

7

Prise de recul

7.1 BILAN DU TRAVAIL EFFECTUÉ

Pour commencer, la majorité des missions ont été menées à leur terme, à savoir :

1. Les tests ont été implementés dans les délais impartis ;
2. Le point API pour télécharger les plans a été implementé dans les délais impartis ;
3. La preuve de concept n'a pas pu être menée à son terme ;
4. Toutes les métriques d'explicabilité ont été implémentées ;

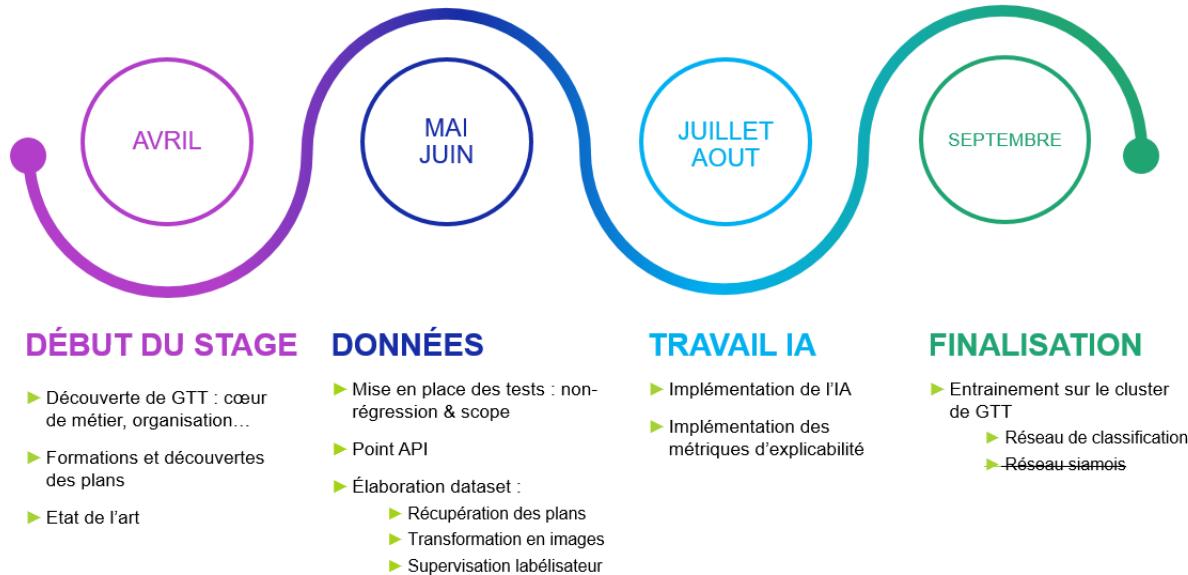


Figure 7.1.1 – Résumé du travail effectué au fil du stage

Concernant la preuve de concept, même si elle n'a pu être menée à son terme, notamment en raison de la lenteur des entraînements et du peu de temps finalement à disposition, la base de code nécessaire pour la réaliser est cependant présente et sera utilisée par Quentin ANDRÉ pour obtenir des résultats.

Enfin, par rapport à la visualisation des embeddings et histogramme de poids, une fois notre [issue github résolue](#), le code sera facilement remis de manière propre et fonctionnelle dans le code d'entraînement.

Le travail envisagé était ambitieux pour un stage mais je suis content d'avoir pu en effectuer le maximum.

7.2 CRITIQUE DES MÉTHODES UTILISÉES

7.2.1 TRAVAUX PRÉALABLES

Tout au long du stage, le travail effectué a utilisé bon nombre de connaissances empiriques disponibles en ligne, comme le choix du modèle pour le réseau siamois [10] ou

le choix du type de « saliency map », qui s'est porté sur les gradCAM, ainsi que la couche sur laquelle l'appliquer.

Ces différents choix auraient pu être remis en cause à l'aide d'un travail plus poussé et de davantage de temps, par exemple :

- Concernant les modèles, en testant plusieurs architectures comme EfficientNet [22] ou EfficientViT [16], mais également leur dimensionnement : dans le cas de Resnet ce dernier est disponible en plusieurs réseau de tailles différentes (Resnet18, Resnet50...);
- Pour les saliency map : en comparant les différentes méthodes à disposition (HiRes CAM, GradCAMElementWise, GradCAM++, XGradCAM, EigenCAM...) à l'aide de [métriques d'explicabilité](#) pour savoir laquelle est la plus performante et serait donc la plus judicieuse ;
- ...

7.2.2 STRATÉGIES DE PARALLÉLISATION

De nombreux aller-retours ont été effectués quant au choix de la stratégie de distribution et un choix plus assertif dès le départ aurait pu être bénéfique au projet car moins de temps aurait été perdu. En effet dans notre cas, énormément de temps a été consacré à comprendre la FSDP et à adapter le code pour l'implémenter, avant de finalement utiliser la DDP.

7.3 ENSEIGNEMENTS ET PROJET PROFESSIONNEL

Je tire de cette expérience de nombreux enseignements, notamment à davantage prendre en considération les facteurs externes, étant donné les nombreux problèmes et retards que cela m'a causé dans ce projet (chapitre 5). Ces enseignements sont également techniques, avec la prise en main et utilisations de nombreuses bibliothèques de Machine Learning ([Pytorch Lightning](#), [grad-CAM](#), [Einops](#)...), méthodes / concepts (réseau siamois, focal loss...) et bonnes pratiques de codes (documentation [Sphinx](#), concepts avancées de Programmation Orientée Objet en Python...).

Certaines de ces connaissances ont été le fruit d'un apprentissage en autonomie,

comme les différentes stratégies de distributions de calculs et l'utilisation de Jarvis, tandis que d'autres, plutôt liées à l'organisation du projet, m'ont été transmises par les différents membres de l'équipe d'AutoCheck.

Ces différents enseignements me serviront à la fois dans le quotidien de ma vie professionnelle mais également pour me démarquer : les connaissances et expériences dans le Deep Learning et la Computer Vision étant particulièrement prisées et c'est sans mal que ce sujet de stage singulier contribuera à cela.

Ayant particulièrement apprécié travailler sur des modèles de deep learning, c'est dans ce domaine, que ce soit en Computer Vision ou en Natural Language Processing que je pense m'orienter !

7.4 LIENS AVEC CURSUS SUIVI AU SEIN D'IMT ATLANTIQUE

Les Unités d'Enseignements suivies au sein d'IMTA ont constitué des bases de connaissances solides, me permettant de rapidement assimiler des concepts non-vus en cours.

Aussi, j'ai pu mobiliser des connaissances provenant de mes deux TAFs, à savoir COPSI et DaSci : concernant la première, j'ai en effet pu faire usage du [problème de couverture par ensembles](#) afin de sélectionner le plus faible nombre d'exemple pour effectuer les grad-CAM. Concernant la seconde, j'ai notamment beaucoup utilisé les connaissances acquises lors de l'UE D : Deep Learning, de par les concepts éponymes largement mis en pratique lors de ce stage, mais également car ce cours présentait de nombreux concepts d'explicabilité. Enfin, de manière plus globale les connaissances générales et bonnes pratiques de Data Scientist apprises au fil des différents cours de la TAF DaSci.

Toutes les connaissances théoriques évoquées, et même certaines non-mentionnées, ont pu être mises en pratique, ce qui est formateur dans le cheminement d'apprentissage.

7.5 IMPACT DU PROJET & INDIVIDUEL

Le projet AutoCheck est certain d'avoir un impact dans plusieurs domaines.

7.5.1 IMPACT ECONOMIQUE & ORGANISATIONNEL

Aux niveaux économique et organisationnel, le gain d'efficacité va permettre à l'entreprise d'économiser de l'argent, mais cela peut également produire une réduction de la masse salariale car moins de salariés seront nécessaires pour produire la même quantité de travail. Cependant, l'application aura un impact positif certain sur le confort de travail des employés, leur permettant de ne plus avoir à réaliser de stroboscheck.

De plus, ce projet pourrait constituer une première étape vers la mise en place d'une base de données requêtéable des plans grâce aux embeddings. Par exemple, un Large Language Model pourrait être utilisé pour vectoriser une requête sous forme de texte afin de trouver le plan le plus proche correspondant ; tous les plans ayant été au préalable vectorisés grâce au travail effectué dans le cadre de ce stage.

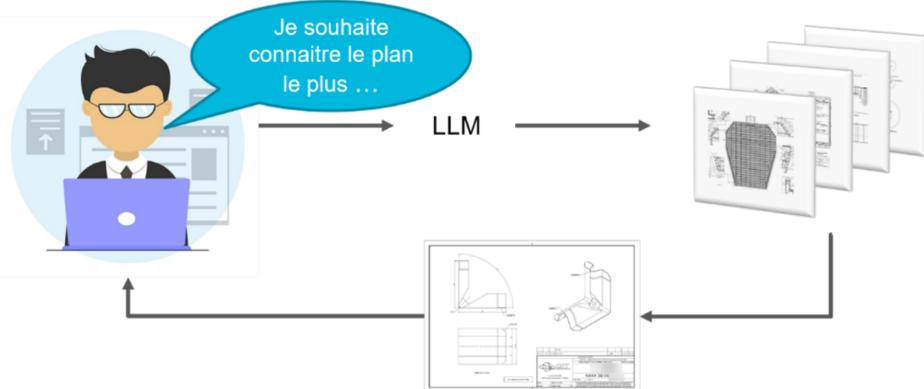


Figure 7.5.1 – Exemple d'une base de données de plans requêtéables : un LLM va embedder la requête d'un utilisateur afin de trouver le(s) plan(s) le(s) plus proche(s)

7.5.2 IMPACT ENVIRONNEMENTAL

PROJET

Au niveau environnemental, l'entraînement ainsi que l'hébergement pour inférence des différents réseaux aura un terme un bilan carbone non-négligeable, étant donné que l'application a vocation à être utilisée pour plusieurs années, avec certainement des ré-entraînements plus ou moins fréquents.

Cependant, le gain en efficacité induit par l'utilisation de l'application ne se traduira pas

en rejet de davantage de CO₂ (lié à la fabrication de davantage de cuves, se répercutant sur le volume de GNL transporté / consommé ...) car GTT ne refuse jamais de contrat.

Le calcul de l'impact environnemental de l'IA a été effectué et est complètement détaillé dans l'Appendice D. Il en résulte les chiffres suivantes :

1. Utilisation jusqu'à présent (tests, débuggage...) : 2 kg_{CO₂} ;
2. Coût de l'entraînement du réseau de classification : 33 kg_{CO₂} ;
3. Coût de l'entraînement du réseau siamois : non-fait par manque de données sur l'entraînement mais vraisemblablement du même ordre de grandeur que le reste ;
4. Coût de l'utilisation projetée par an (inférences) : 114 kg_{CO₂} ;

En comparaison des 550 tonnes de CO₂ émises par l'entraînement de ChatGPT-3 lors de son entraînement [17], l'impact de notre projet est comparable à une goutte d'eau dans l'océan, cependant la démarche écologique reste louable !

DÉCISIONS INDIVIDUELLES

Du point de vu individuel, de nombreuses décisions relatives à l'impact environnement de l'IA et du projet en général ont été prises en adéquation avec Quentin André. Notamment :

1. La réflexion quant aux modèles : même si le choix s'est porté sur le Resnet et a été justifié précédemment, EfficientViT et EfficientNet avaient été sélectionnés car permettant un contrôle fin des ressources utilisables et du dimensionnement du réseau [22] [16] ;
2. La génération tardive du dataset : le code étant quasiment prêt depuis 6 semaines au moment de la génération finale, il a été décidé d'attendre que le maximum de features d'AutoCheck soient prêts afin de ne le générer qu'une fois. Le processus total durant 2 jours. Cependant cela nous a également ralenti car il a fallu finaliser le code et le débugger une fois la décision de lancer la génération prise ;
3. Enfin, si le temps l'avait permis et les résultats probants, nous aurions mené une étape de [distillation des connaissances](#), visant à spécialiser un modèle réduit à imiter le fonctionnement d'un modèle de taille plus importante. Ceci afin d'avoir un modèle moins énergivore ;

7.6 CONCLUSION

Pour finir, la mission AutoCheck a été renouvelée pour une année supplémentaire, ce qui permettra à Quentin ANDRE de repartir de mes bases pour achever mon travail, et également mettre en place un autre type de détection d'erreurs, sans comparaison, pour lequel nous avons déjà réfléchi à une solution technique.

C'est à la fois avec un sentiment de fierté, au vu de tout le travail accompli durant ces quelques mois, mais également avec un goût d'inachevé, étant donné que je n'ai pu aller au bout du projet et véritablement voir les résultats finaux, que se conclut ce projet pour moi.

J'ai été content de faire partie de l'équipe d'AutoCheck, qui fut véritablement une expérience enrichissante et profite de ces dernières lignes pour remercier une dernière fois Quentin ANDRÉ, Guillaume MORIN et Juliette DARNAL pour ces quelques mois à travailler ensemble.

A

Détail des données

Si GTT possède deux technologies principales, nous n'en détaillerons qu'une par soucis de concision, mais présenterons néanmoins rapidement les deux.

A.1 LES TECHNOLOGIES DE GTT

GTT possède 2 technologies principales :

— MARK III : système d'isolation fixé sur la coque interne du navire, composé d'une membrane primaire ondulée en acier inoxydable disposée sur des panneaux isolants en mousse préfabriquée intégrant une membrane secondaire composite. Cette technologie a l'avantage d'utiliser des composants préfabriqués standards, pouvant donc facilement être produits et assemblés ;

— NO96 : également fixé sur la coque interne du navire, composé d'une isolation primaire de caissons de bois contreplaqué remplis de matériaux isolants et espacés entre eux avec deux membranes métalliques d'invar, alliage de fer-nickel utilisé pour ses propriétés physiques. Ces éléments sont moins standardisés et doivent être fabriqués sur place par les chantiers ;

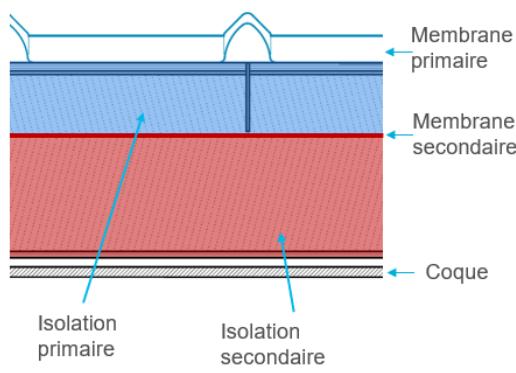
Les technologies de GTT

● NO96



● Un principe commun

- Système de confinement fixé à la coque intérieure du navire
- Deux couches d'isolation
- Deux systèmes d'étanchéité



● Mark III

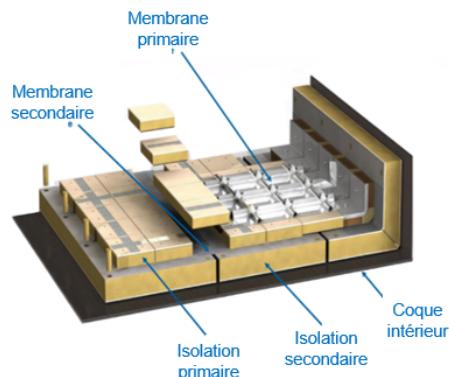


Figure A.1.1 – Les technologies de GTT

Penchons-nous en détail sur la technologie MARK III.

A.2 LES CUVES

A.2.1 LA GÉOMÉTRIE

Le nombre de cuves installées dans un navire dépend évidemment de sa taille. De manière standard, les navires équipés mesurent 300m de long pour 40m de largeur et 30m de hauteur. En général, 4 cuves sont installées et leur géométrie est variable, pouvant être trapézoïdale, prismatique ou parallélépipédique, le tout afin de s'adapter au mieux au navire.

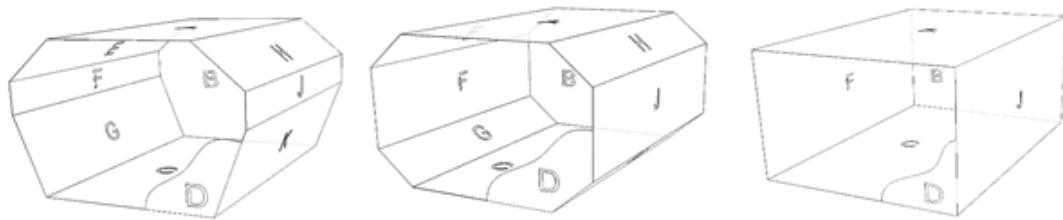


Figure A.2.1 – Les différentes géométries des cuves

La cuve numéro une est située à l'avant du navire et diffère généralement des trois autres de par sa forme et ses dimensions.

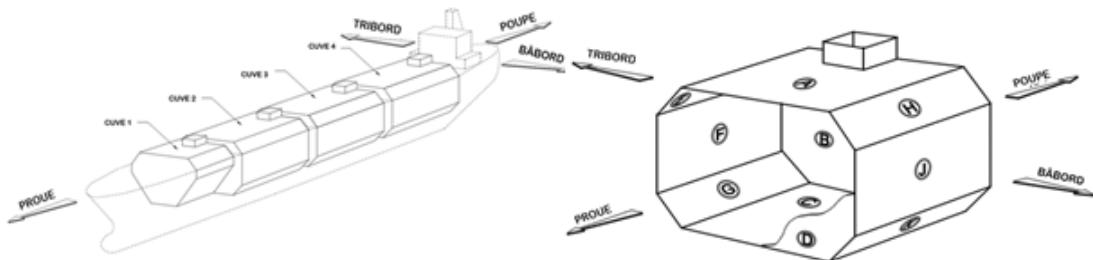


Figure A.2.2 – Localisation et numérotation des cuves

A.2.2 LES DIFFÉRENTES ZONES

Au sein d'une cuve, on distingue différentes zones « normales », représentant des faces ou intersections de ces dernières :

- Les Flatwalls (FW) : faces d'une cuve, numérotées par des lettres allant de A à P en fonction de la forme (voir Figure A.2.1) et de l'orientation (voir Figure A.2.2) ;
- Les Dièdres (DR) : intersection entre deux faces ;

- Les Trièdres (TR) : intersection entre 3 faces ;

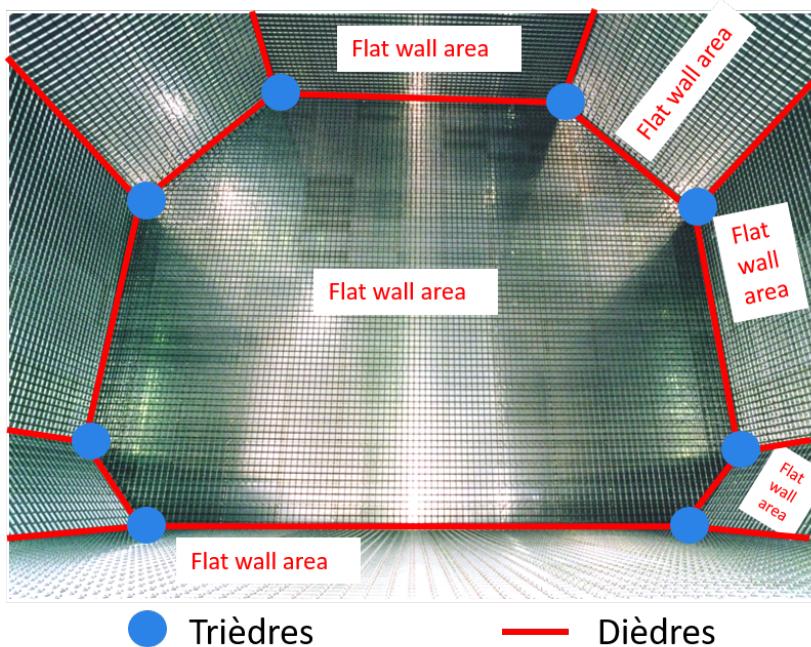


Figure A.2.3 – Localisation des Flatwall, Dièdres et Trièdres au sein d'une cuve

Ces zones sont en opposition aux zone dites « spéciales », localisant des éléments venant induire de nouvelles contraintes et nécessitant des règles de calepinage spécifiques :

- Liquid Dome (LD) : ouverture sur la face supérieure (A) permettant de relier la cuve à l'extérieur (passage des tuyaux) ;
- Pump Tower Base Support (PTBS) pièce fixée sur le plancher de la cuve, permettant une liaison glissière au mat contenant les tuyaux ;
- Gas Dome (GD) : ouverture sur la face supérieure (A) permettant des échanges gazeux ;
- Side Opening (SO) : ouverture temporaire permettant de faciliter l'acheminement des matériaux de construction ;
- Gas Pocket (GP) : passage situé dans la membrane et permettant au gaz d'éviter une compression trop importante liée au sloshing (dans le cas contraire, le mouvement du liquide pouvant être amené à comprimer une poche de gaz si cette dernière ne possède pas d'échappatoire) ;

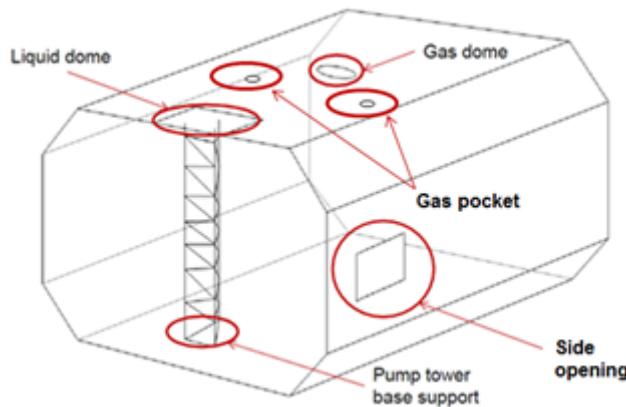


Figure A.2.4 – Localisation des différentes zones spéciales

A.3 LES DIFFÉRENTS COMPOSANTS

La technologie MARK III possède l'avantage d'être adaptable à tout type de zones de la cuve grâce à la répétition d'un motif, ce qui permet de limiter le nombre d'éléments, de calculs et donc les coûts. Les principaux éléments de cette technologies sont :

- Les Flat Panels (FP) : installés dans les FW ils représentent jusqu'à 90% des éléments d'isolation utilisés ;
- Les Corner Panels (CP) : installés au niveau des DR ;
- Les Trihèdres (TR) : installés à l'intersection de 3 faces ;
- Les Top Bridge Pads (BP) : comblant les espaces au niveau de l'isolation primaires entre deux FP ;
- Les Erection-on-Boards (EB) : comblant les espaces au niveau de l'isolation primaire entre deux CP et autour des TR ;
- Les Membrane Sheets (MS) : assurant l'étanchéité entre le GNL et les FP et caractérisées par des ondulations pour ne pas se rompre en raison des forces en vigueur ;

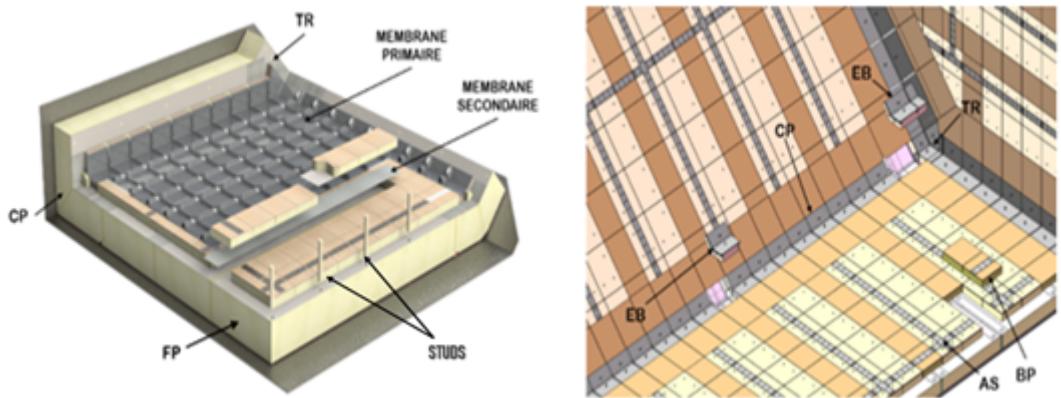
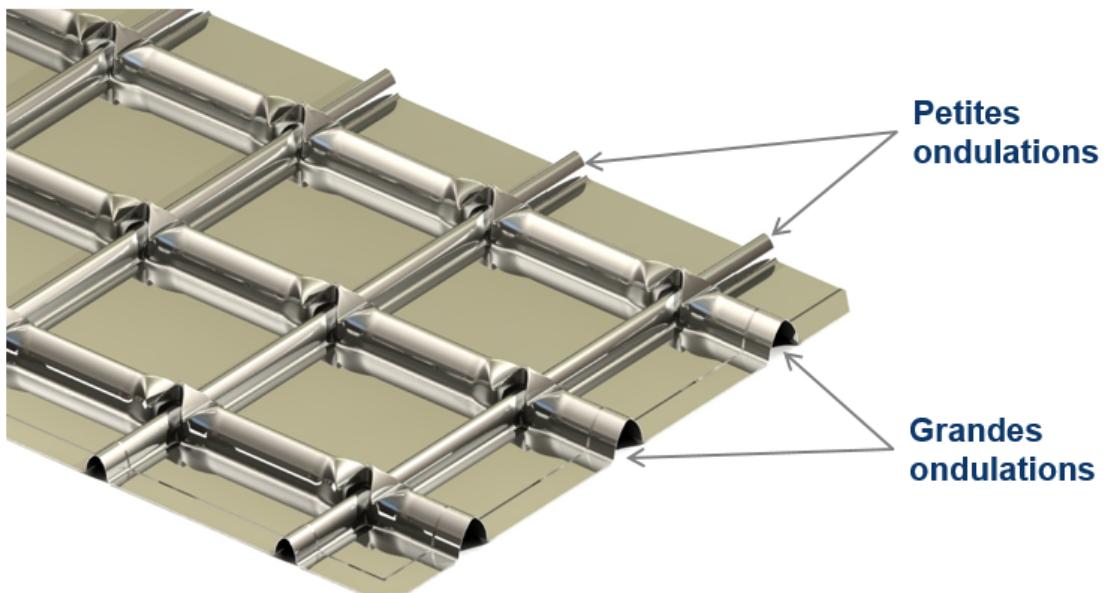


Figure A.3.1 – Localisation des différentes éléments de la technologie MARK III



- Standard size : 3m x 1m
- Standard pitch : 340mm

Figure A.3.2 – Apparence d'une Membrane Sheet. On remarque les courtes et longues ondulations caractéristiques permettant la déformations de la cuve lorsqu'elle est soumise aux contraintes longitudinales et perpendiculaires.

... mais encore d'autres éléments existent ! De plus, ces éléments, possèdent différentes

références avec des dimensions ou géométries propres d'où la mise en place d'une nomenclature afin de pouvoir identifier chaque pièce.

A.4 CODING

La nomenclature au sein de GTT est communément appelée « coding ». En ce qui concerne la technologie Mark, il existe environ 13 500 références d'éléments utilisés pour les projets. Près de 1 000 éléments composent systématiquement chaque projet. Ainsi, une modification de l'épaisseur d'isolation sur un projet provoque la création d'environ 1 700 nouvelles références d'éléments.

C'est pourquoi il est nécessaire d'avoir un système de nomenclature précis permettant d'identifier sur les plans la position et la nature de chacun des composants. A chaque code correspond une seule pièce. Par exemple, pour les FP le code de nomenclature est composé de :

- Deux caractères identifiant l'élément, donc ici « FP » ;
- Deux lettres définissant la hauteur ainsi que la densité ;
- Un chiffre définissant la forme ;
- Deux chiffres pour la taille : ici le nombre de caissons dans les deux directions ;
- Un chiffre pour la particularité (déterminé par le schéma des bandes métalliques sur les panneaux primaires) ;
- Une lettre (A ou B) pour définir l'existence d'un autre élément symétrique ;
- Un compteur indiquant la position d'éléments spécifiques ;

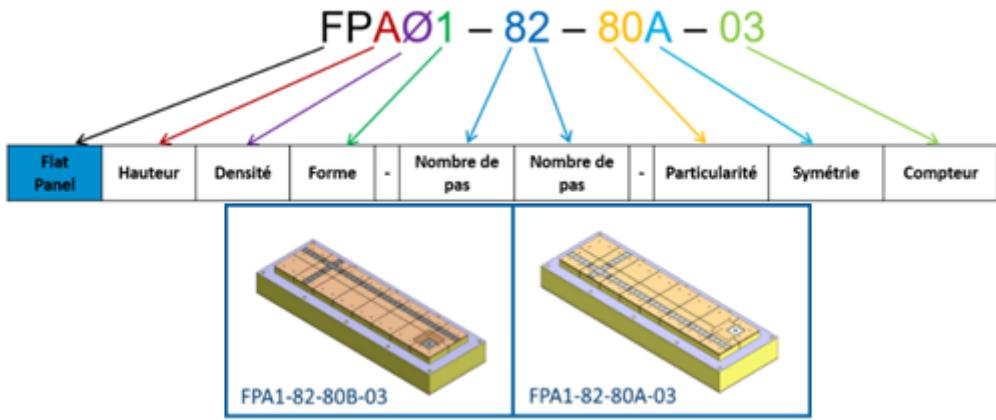


Figure A.4.1 – Nomenclature des Flat Panels (FP)

A.5 LES DIFFÉRENTS TYPES DE PLAN

Il existe différents types de plans, communs aux deux technologies :

- **Principe (PR)** : qui définit un type de pièce de manière standard sous tous ses angles ;
- **Définition (DE)** : qui définit les dimensions et compositions des différentes références (variations) de la pièce standard utilisées dans le projet ;
- **Arrangement (AR)** : qui explicite la position de chaque élément ainsi que l'ordre de montage des différents éléments ;
- **Listing (LG)** : qui liste le nombre de chaque référence nécessaire ;

A.6 LA NOMENCLATURE DES LIVRABLES

En raison de la quantité pléthorique d'informations définies ci-dessus, il est impératif de pouvoir naviguer aisément parmi les différents livrables. Pour cela, la nomenclature suivante, utilisée en MARK, permet de se rendre compte du nombre de livrables existant :

NXXX TS AR 0C

Zone	Type of drawing	Tank	Bulkhead
GL General	AR ARrangement	0 All Tank	0 All Bulkhead
FW Flat Wall	PR PRinciple	1 Tank 1	A Bulkhead A
LD Liquid Dome	DE DEfinition	2 Tank 2	B Bulkhead B
TS pump Tower base Support	LG ListinG	3 Tank 3	C Bulkhead C
GD Gas Dome		4 Tank 4	D Bulkhead D
SO Side shell Opening			E Bulkhead E
SW Sump Well			F Bulkhead F
DR DRaining			G Bulkhead G
AI Ammonia Injection			H Bulkhead H
ST Temperature Sensor			J Bulkhead J
ME Messenger			K Bulkhead K
DB Data Base			L Bulkhead L
			M Bulkhead M
			N Bulkhead N
Definition of elements			
AP Angle Piece			
AS Anchoring Strip			
BP Top Bridge Pad			
...			
CP Corner Panel			
DL Dog Leg			
EB Erection on Board			
EC End Corrugation			
FJ Flat Joint			
FP Flat Panel			
...			

Examples

- GL PR 00
- LD AR 00
- LD PR PI
- TS DE 00
- GL LG AP
- DB PR FP
- FW AR 2BP
- DR PR OC

Figure A.6.1 – Nomenclature des livrables, NXXX définissant un numéro de projet

A.7 EXEMPLES DE PLAN

Sur la figure suivante se trouve un échantillon de plans de GTT afin que le lecteur puisse se rendre compte de la complexité et de la diversité des plans.

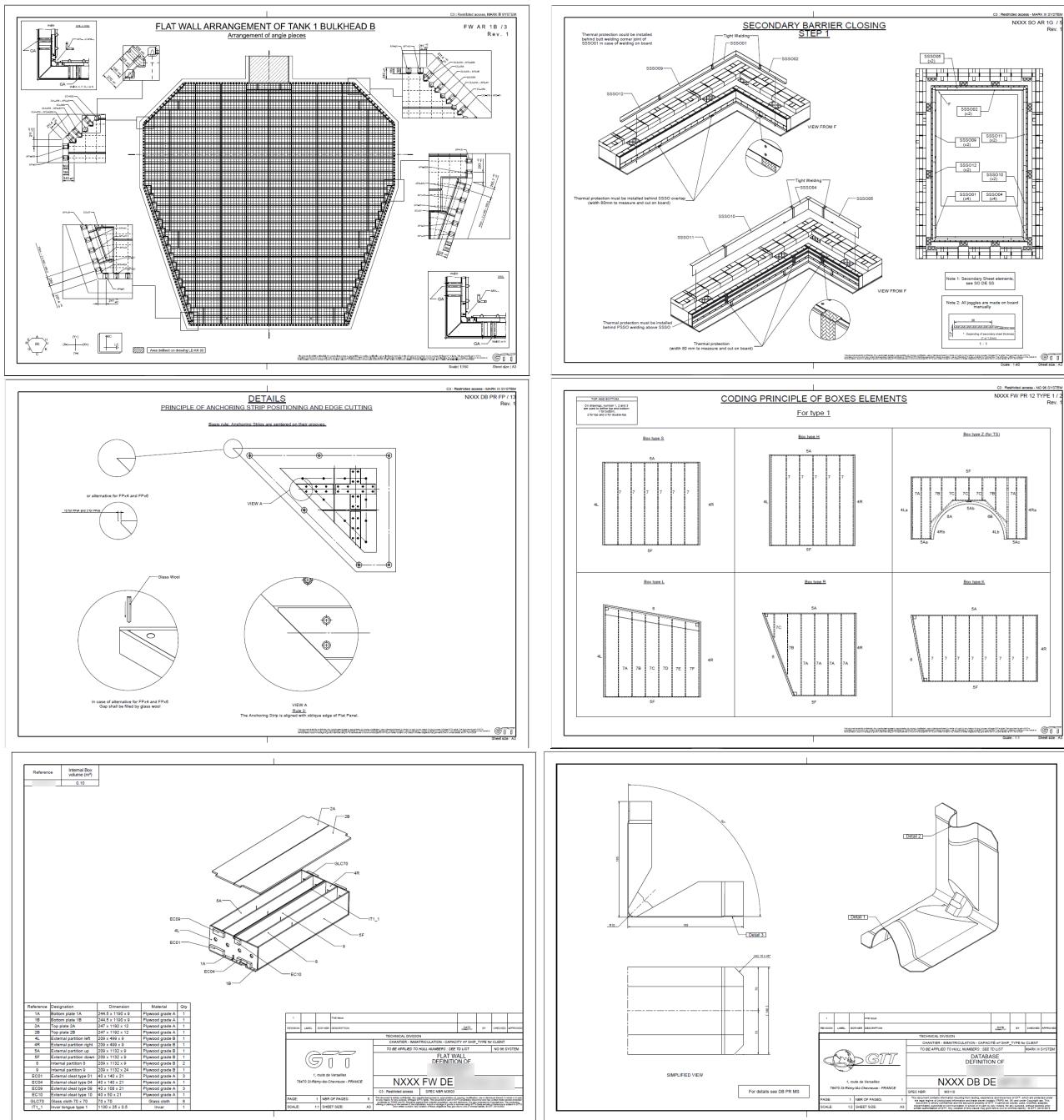


Figure A.7.1 – Échantillon de plans de GTT

Retour Familiarisation avec les plans (section 4.1)

B

Calcul de la taille mémoire d'un réseau

Le calcul de la taille mémoire d'un réseau, plus simple à réaliser *a posteriori* car devant tenir compte des résultats de chaque couche intermédiaire, est le suivant

$$\text{mémoire}_{\text{VRAM}} = (n_{\text{input}} + n_{\text{params}} + n_{\text{activations}}) \times \frac{n_{\text{bytes}}_{\text{précision}}}{1024^3}$$

Avec :

- n_{input} : le nombre de paramètres en entrée. Dans notre cas, avec un tenseur de dimensions (B,C,H,W) cela revient à $n_{\text{input}} = B \cdot C \cdot H \cdot W$;
- n_{params} : le nombre de paramètres de notre modèle ;
- $n_{\text{activations}}$: le nombre d'activations intermédiaires, nécessaires pour le calcul des gradients lors de la mise à jour des poids du réseaux. C'est donc la somme du nombre de paramètres en sortie de chaque couche et c'est dans la grande

majorité des cas le facteur limitant. On a $n_{activations} = \sum_i n_{outputs,i}$ où $n_{outputs,i}$ représente le nombre de paramètres en sortie de la couche i;

— $\frac{n_{bytes_{précision}}^3}{1024}$: le facteur multiplicatif servant à convertir le nombre de paramètres en nombre de Gigabytes ;

— avec $n_{bytes_{précision}}$ le nombre d'octets nécessaire pour stocker les chiffres en fonction de la précision ;

Dans notre cas :

Soit

$$mémoire_{VRAM} = 34 \quad [\text{Gb}] \quad (\text{B.1})$$

Retour Stratégies de parallélisation (sous-section 4.6.3)

C

Algorithme d'ordonnancement de Jarvis

MOAB, l'ordonnanceur de Jarvis oblige les jobs à déclarer les ressources nécessaires à leur exécution afin d'optimiser au mieux l'utilisation du cluster. Doivent notamment être déclarés :

- La mémoire disque
- La mémoire RAM
- Le nombre de CPU
- Le nombre de GPU [Nota Bene : la manière de déclarer cette ressource n'était même pas explicité dans la documentation interne cf sous-section 5.3.1]
- Une estimation du temps d'exécution
- La file d'attente

Nom	Type	Ressources minimum	Ressources maximum	Priorité	Horaires d'accès
small	Petits calculs demandant peu de temps	Pas de ressource minimum	4 CPU, 5 minutes	Haute	7/7j - 24/24h
short	Calculs moyens de durée raisonnable	Pas de ressource minimum	16 CPU, 18 minutes	Faible	7/7j - 24/24h
long	Calculs longs	8 CPU	32 CPU, pas de limite de temps	Très Faible	7/7j - 24/24h
big	Très gros calculs	4 CPU	64 CPU, 72 heures	Maximale lors des horaires d'accès, nulle sinon	24/24h le week-end, 19h30-8h30 sinon

Table C.0.1 – Description des différents types de calculs

Retour Ordonnancement (sous-section 5.3.3)

D

Calculs de l'impact environnemental de l'IA

Pour avoir un meilleur regard critique sur l'impact environnemental du projet, et du travail produit au cours de ce stage, j'ai effectué le calcul du coût carbone du réseau de neurone. en me basant sur diverses publications [12] [21] et cours donnés par Gabor BELLA [5].

D.1 MÉTHODOLOGIE

Calculer l'impact écologique d'un réseau de neurone correspond à quantifier la quantité de carbone qu'il a rejeté pour son entraînement et qu'il rejettéra au cours de son utilisation lors de ses inférences.

Dans le cadre de l'IA, on relie l'énergie utilisée sous forme d'électricité par le réseau au

rejet de CO_2 induit par la production de l'électricité avec l'équation suivante :

$$CO_2 \text{eq}_{\text{réseau}} = E_{\text{élec}} \times CO_2 \text{eq}_{kWh} \quad [\text{kg}_{CO_2}] \quad (\text{D.1})$$

avec, en France, grâce à son énergie bas-carbone :

$$CO_2 \text{eq}_{kWh} = 32 \cdot 10^{-3} \quad [\text{kWh kg}_{CO_2}^{-1}] \quad (\text{D.2})$$

Enfin, on obtient l'énergie électrique consommée par le réseau grâce à :

$$E_{\text{élec}} = \frac{PUE}{1000} \times (P_{CPU} + P_{RAM} + P_{GPU}) \times temps \quad [\text{kWh}] \quad (\text{D.3})$$

où PUE (Power Usage Effectiveness) correspond à un facteur de correction permettant de tenir compte des sources additionnelles d'énergie, notamment utilisées pour refroidir les composants électroniques, afin d'obtenir la consommation effective. La moyenne de ce coefficient avait comme valeur en 2018, d'après [4] :

$$PUE = 1.58 \quad [\text{sans unité}] \quad (\text{D.4})$$

D.2 DONNÉES À DISPOSITION

On possède les données suivantes, dont notamment n_{FLOP} grâce à [Calflops](#) et $v_{NVIDIA_{V100}}$ grâce à [12]; de plus, on estime les paramètres du réseau siamois à deux fois ceux du

réseau de classification :

$P_{GPU_{actif}} = 300$	[W]	(D.5)
$P_{GPU_{inactif}} = 50$	[W]	(D.6)
$P_{CPU} = 100$	[W]	(D.7)
$P_{RAM} = 8$	[W]	(D.8)
$m_{GPU} = 16 \cdot 10^9$	[Bytes]	(D.9)
$m_{réseau}^{classification} = 0.167$	[Gb]	(D.10)
$m_{réseau}^{siamois} \approx 2 \cdot m_{réseau}^{classification}$	[Bytes]	(D.11)
$m_{image} = \cdot 10^3$	[Bytes]	(D.12)
$n_{images}^{classification} = 195 \times 10^3$		(D.13)
$n_{FLOP_{inférence}}^{classification} = 686 \cdot 10^9$	[FLOP batch ⁻¹]	(D.14)
$n_{FLOP_{entraînement}}^{classification} = 2.06 \cdot 10^{12}$	[FLOP batch ⁻¹]	(D.15)
$n_{images}^{siamois} = 2500 \times 10^3$		(D.16)
$n_{FLOP_{entraînement}}^{siamois} \approx 2 \cdot n_{FLOP_{entraînement}}^{classification}$	[FLOP batch ⁻¹]	(D.17)
$n_{FLOP_{inférence}}^{siamois} \approx 2 \cdot n_{FLOP_{inférence}}^{classification}$	[FLOP batch ⁻¹]	(D.18)
$V_{NVIDIA V100_{précision}} = 31.4 \cdot 10^{12}$	[FLOPs s ⁻¹]	(D.19)

D.3 ENTRAINEMENTS

Les entraînements, que ce soit pour la classification ou le réseau siamois, ont été ou seront réalisés à l'aide de 2 GPU, chacun avec 2 CPU et le minimum de RAM possible (de puissance totale P_{RAM}) donc

$$P_{entraînement} = (2 \times P_{GPU_{actif}} + 2 \times P_{CPU} + P_{RAM})$$

$$P_{entraînement} = 808 \quad [\text{W}] \quad (\text{D.20})$$

D.3.1 CALCUL D'UN ENTRAINEMENT

CLASSIFICATION

Soit $n_{epoch}^{classification} = 100$ (hypothèse sur un entraînement avec tous les labels) et avec une proportion de train/val/test de respectivement 0.7, 0.15 et 0.15 ainsi qu'un batch size de 24, il vient :

$$\begin{aligned} n_{train \ steps}^{classification} &= \frac{n_{images}^{classification} \times 0.7}{batch \ size} &= 6906 \\ n_{val \ steps}^{classification} &= \frac{n_{images}^{classification} \times 0.15}{batch \ size} &= 1218 \\ n_{test \ steps}^{classification} &= \frac{n_{images}^{classification} \times 0.15}{batch \ size} &= 1218 \end{aligned}$$

Soit une quantité totale de FLOPs :

$$\begin{aligned} n_{totFLOPs}^{classification} &= n_{epoch}^{classification} \times \left(n_{train \ steps}^{classification} \times n_{FLOP_{entraînement}}^{classification} + n_{val \ steps}^{classification} \times n_{FLOP_{inférence}}^{classification} \right) \\ &\quad + n_{test \ steps}^{classification} \times n_{FLOP_{entraînement}}^{inférence} \\ n_{totFLOPs}^{classification} &= 1.5 \cdot 10^{18} \end{aligned} \quad [\text{FLOP}] \quad (\text{D.21})$$

Pour un temps d'entraînement :

$$\begin{aligned} t_{entraînement} &= \frac{n_{totFLOPs}^{classification}}{V_{NVIDIA V100_{précision}}} \\ t_{entraînement} &= 796 \end{aligned} \quad [\text{h}] \quad (\text{D.22})$$

Ainsi, en se basant sur (D.3), on obtient :

$$E_{entraînement}^{classification} = \frac{PUE}{1000} \times P_{entraînement} \times t_{entraînement} \quad [\text{kWh}] \quad (\text{D.23})$$

$$E_{entraînement}^{classification} = 1016 \quad [\text{kWh}] \quad (\text{D.23})$$

$$CO_2eq_{entraînement}^{classification} = 33 \quad [\text{kg CO}_2] \quad (\text{D.24})$$

SIAMOIS

Par manque de données sur le nombre d'epochs nécessaires à l'obtention de résultats, le calcul ne peut-être effectué. Si cette donnée avait été disponible au moment de l'écriture de ce rapport, en suivant la démarche utilisée précédemment, le calcul aurait pu être mené à son terme.

D.3.2 ESTIMATION DES RÉSULTATS PRÉALABLES / TESTS / DÉBUGGAGES

Avant d'obtenir un réseau fonctionnel sans bugs, de nombreux tests et entraînements intermédiaires ont été réalisés.

On estime :

$$t_{tests} = 500 \quad [\text{h}] \quad (\text{D.25})$$

d'où avec (D.3) et (D.20) :

$$E_{tot_{tests}} = \frac{PUE}{1000} \times P_{entraînement} \times t_{tests}$$
$$E_{tot_{tests}} = 638.32 \quad [\text{kWh}] \quad (\text{D.26})$$

$$CO_2eq_{tests} = 2 \quad [\text{kg CO}_2] \quad (\text{D.27})$$

D.4 INFÉRENCES

Calculons à présent l'énergie électrique utilisée par le réseau pour effectuer des inférences. Pour cela, nous allons effectuer un calcul sur une année sous les hypothèses suivantes :

1. L'année est moyenne : il y a 20 projets MARK et 5 NO
2. Chaque plan nécessite d'être vérifié 3 fois
3. L'application est disponible 12 heures par jour
4. L'application utilise 2 GPU, chacun avec 2 CPU et le minimum de RAM possible (de puissance totale P_{RAM})
5. Une inférence est suffisante pour checker un plan, ainsi le nombre

d'inférences à effectuer correspond au nombre de plans à checker fois le nombre de vérifications à faire

Même si les réseaux ne tournent pas, ils doivent être chargés sur Jarvis et restés à disposition. Ils consomment donc de l'énergie.

On peut calculer le temps d'utilisation et d'inactivité des GPU par :

$$t_{actif} = n_{inférence} \times t_{inférence} \quad [s] \quad (D.28)$$

$$t_{inactif} = t_{disponible} - t_{actif} \quad [s] \quad (D.29)$$

$$t_{disponible} = n_{GPU} \times 365,25 \times 12 \times 60 \quad [s] \quad (D.30)$$

et le temps d'inférence peut se calculer grâce au nombre d'opération nécessaires à une inférence, dont l'unité est le FLOP, et à la vitesse de traitement du matériel utilisé en FLOPS. Ainsi on a :

$$t_{inférence} = \frac{n_{FLOP}}{V_{GPU}} \quad [s] \quad (D.31)$$

Ce qui finalement, nous permet de calculer, à l'aide de (D.3)

$$E_{tot_{inférence}} = \frac{PUE}{1000} \left(t_{disponible} \times (P_{CPU} + P_{RAM}) + t_{actif} \times P_{GPU_{actif}} + t_{inactif} \times P_{GPU_{inactif}} \right) [\text{kWh}] \quad (D.32)$$

Ce qui donne, après calculs et avec (D.1) :

$$E_{tot_{inférence}} = 3\,589 \quad [\text{kWh}] \quad (D.33)$$

$$CO_2eq_{inférence} = 114 \quad [\text{kg CO}_2] \quad (D.34)$$

Dans ce calcul, l'impact des inférences est minime comparé à l'uptime de l'infrastructure (en ODG : moins de 1 %)

Retour Impact du projet & individuel (section 7.5)

E

Exemple d'entraînement en attente

Pour expliquer les problématiques d'entraînement liées à Jarvis, suivons une tentative d'entraînement en journée.

Etant donné que l'on souhaite effectuer un entraînement, et non tester le code, la file choisie est « long ».

Lançons un premier job, qui portera le numéro 482 436, avec les caractéristiques suivantes :

- 8 processeurs ;
- 2 cartes graphiques ;
- 6 heures ;
- File long ;

Notons t le temps de lancement du job.

active jobs-----						
JOBID	USERNAME	STATE	PROCS	REMAINING	STARTTIME	
482447		Running	16	00:14:23	Fri Sep 20	10:08:06
482361		Running	4	8:48:18	Fri Sep 20	08:30:01
482362		Running	2	8:48:18	Fri Sep 20	08:30:01
482421		Running	32	14:57:14	Thu Sep 19	20:38:57
4 active jobs		54 of 144 processors in use by local jobs (37.50%)				
		4 of 6 nodes active (66.67%)				
eligible jobs-----						
JOBID	USERNAME	STATE	PROCS	WCLIMIT	QUEUETIME	
482436	pae	Idle	8	6:00:00	Fri Sep 20	09:28:07
482442		Idle	16	00:18:00	Fri Sep 20	09:46:36
482443		Idle	16	00:18:00	Fri Sep 20	09:46:44

Figure E.0.1 – Etat du cluster à t+44

Le job est placé en attente, tentons de comprendre pourquoi.

pae@jarvis-m:~> mdiag -p 482436
diagnosing job priority information (partition: ALL)
Job
Weights
482436
482443

Figure E.0.2 – Priorité des jobs

Les règles de calcul de la priorité et les stratégies d'ordonnancement sont choisies par l'administrateur. Notamment, le calcul de la priorité prend en compte les ressources demandées et plus une priorité numérique est basse et plus le job est prioritaire pour l'ordonnanceur. Ici, mon job passera après le job 482 443.

Pourtant en regardant l'onglet « active job » Figure E.0.1, peu de calculs et ressources

semblent être utilisés. Vérifions cela.

```
pae@jarvis-m:~> checknode jarvis-slave-03
node jarvis-slave-03

State:      Idle  (in current state for 14:40:06)
Configured Resources: PROCS: 32  MEM: 503G  SWAP: 8192M  DISK: 784G  GPUS: 2  socket: 2  numanode: 2  core: 32  thread: 32
Utilized   Resources: DISK: 29G
Dedicated  Resources: ---
```

Figure E.0.3 – On observe sur l'onglet « Utilized Ressources » qu'aucun calcul ne tourne sur le noeud 3

En effet cela se vérifie car le noeud 3 est disponible : aucun calcul ne tourne dessus et toutes ses ressources sont disponibles.

Mon calcul pourrait tourner mais ce n'est pas le cas.

```
pae@jarvis-m:~> showq

active jobs-----
JOBID          USERNAME      STATE  PROCS      REMAINING           STARTTIME
482443          pae        Running    16      00:13:11  Fri Sep 20 10:34:17
482361          pae        Running     4      8:20:55   Fri Sep 20 08:30:01
482362          pae        Running     2      8:20:55   Fri Sep 20 08:30:01
482421          pae        Running    32     14:29:51  Thu Sep 19 20:38:57

4 active jobs      54 of 144 processors in use by local jobs (37.50%)
                           4 of 6 nodes active      (66.67%)

eligible jobs-----
JOBID          USERNAME      STATE  PROCS      WCLIMIT           QUEUETIME
482436          pae        Idle     8      6:00:00  Fri Sep 20 09:28:07

1 eligible job
```

Figure E.0.4 – Etat du cluster à t+72

Le job 482 443 est bien lancé avant notre job.

A t+77, nous lançons un second job, portant le numéro 482 454 avec les caractéristiques suivantes :

- 2 processeurs ;
- 2 cartes graphiques ;

- 6 heures ;
- File long ;

```
pae@jarvis-m:~> showq

active jobs-----
JOBID          USERNAME      STATE  PROCS   REMAINING           STARTTIME
482361          Running       4      8:10:57  Fri Sep 20 08:30:01
482362          Running       2      8:10:57  Fri Sep 20 08:30:01
482421          Running      32     14:19:53  Thu Sep 19 20:38:57

3 active jobs      38 of 144 processors in use by local jobs (26.39%)
                      3 of 6 nodes active      (50.00%)

eligible jobs-----
JOBID          USERNAME      STATE  PROCS   WCLIMIT           QUEUETIME
482436          pae        Idle    8      6:00:00  Fri Sep 20 09:28:07
482454          pae        Idle    2      6:00:00  Fri Sep 20 10:45:58

2 eligible jobs
```

Figure E.0.5 – Etat du cluster à t+82

Même ce job, nécessitant moins de processeurs, reste en attente sur le cluster.

Tentons un dernier essai en lançant un job, de numéro 482 461, à t+119 avec les caractéristiques suivantes :

- 8 processeurs (minimum de la file) ;
- 2 cartes graphiques ;
- 18 minutes ;
- File short ;

Ce job correspond donc à un « test de code » plutôt qu'à un entraînement, étant donné le temps maximal disponible.

```
pae@jarvis-m:~> showq

active jobs-----
JOBID          USERNAME      STATE PROCS      REMAINING           STARTTIME
482361          Running       4      7:27:44   Fri Sep 20 08:30:01
482362          Running       2      7:27:44   Fri Sep 20 08:30:01
482421          Running      32     13:36:40  Thu Sep 19 20:38:57

3 active jobs      38 of 144 processors in use by local jobs (26.39%)
                           3 of 6 nodes active      (50.00%)

eligible jobs-----
JOBID          USERNAME      STATE PROCS      WCLIMIT           QUEUETIME
482436          pae        Idle    8      6:00:00   Fri Sep 20 09:28:07
482454          pae        Idle    2      6:00:00   Fri Sep 20 10:45:58
482461          pae        Idle    8      00:18:00  Fri Sep 20 11:27:32

3 eligible jobs
```

Figure E.0.6 – Etat du cluster à t+125

6 minutes après la soumission du job, ce dernier n'est toujours pas exécuté.

```
pae@jarvis-m:~> showq

active jobs-----
JOBID          USERNAME      STATE PROCS      REMAINING           STARTTIME
482436          pae        Running  8      5:59:29   Fri Sep 20 11:35:49
482361          Running     4      7:23:41   Fri Sep 20 08:30:01
482362          Running     2      7:23:41   Fri Sep 20 08:30:01
482421          Running    32     13:32:37  Thu Sep 19 20:38:57

4 active jobs      46 of 144 processors in use by local jobs (31.94%)
                           4 of 6 nodes active      (66.67%)

eligible jobs-----
JOBID          USERNAME      STATE PROCS      WCLIMIT           QUEUETIME
482454          pae        Idle    2      6:00:00   Fri Sep 20 10:45:58
482461          pae        Idle    8      00:18:00  Fri Sep 20 11:27:32

2 eligible jobs
```

Figure E.0.7 – Etat du cluster à t+129

Enfin, à t+129, le premier job soumis est enfin exécuté.

Cette exemple démontre l'incertitude entourant l'exécution des jobs sur Jarvis et que des améliorations concernant les règles d'ordonnancement ou leur bon exécution sont possibles afin de pouvoir entraîner des algorithmes de Machine Learning.

Retour Ordonnancement (sous-section 5.3.3)

Glossaire

A | B | C | D | E | F | G | H | I | J | K | L | M | N | P | R | S | T | V

A

Abaqus —

Progiciel de calculs d'éléments finis développé par une filiale de Dassault Systèmes et utilisé par GTT 46, 60

ADC — Anomalie DéTECTée Client

Anomalie détectée sur les plans par le client une fois les plans livrés, ce qui baisse les KPI de GTT en plus de pouvoir lui coûter davantage de pénalités comparé à une ADGTT 22, 26, 52, 102, 110

ADGTT — Anomaliée DéTECTée GTT

Anomalie détectée sur les plans par GTT avant l'envoi au client; moins grave qu'une ADC 22, 26, 102, 110

AR — Arrangement

Type de plan explicitant la position de chaque élément ainsi que l'ordre de montage des différents éléments. 30, 83

AutoCheck —

Outil développé par MP DATA à la demande de GTT et visant à automatiser la vérification des plans et donc à remplacer le strobocheck; une version systématique a été développé par mon tuteur et le but de mon stage est d'y intégrer de l'IA afin de traiter les éléments variants xiii, 23–25, 30, 34–36, 38–40, 58, 68, 72, 74, 75, 110

B

Backpropagation — backpropagation

« Rétropropagation du gradient », étape de mise à jour des poids d'un réseau de neurone à l'aide des gradients en partant de la dernière couche et en remontant à la première 62

Batch — batch

Littéralement, « lot », correspond à un sous-ensemble des données, pouvant être traité en parallèle par le réseau et dont la taille peut influencer l'entraînement de ce dernier. 59

Bounding box — bounding box

Rectangle d'aire minimale englobant l'ensemble des points d'intérêt. Dans notre cas, la bounding box est non-orientée et donc définie par 2 points formant un rectangle. 38, 41, 43

BP — Top Bridge Pad

Élément de la technologie MARK III installé entre deux FP 34, 80

Bunker — bunker

Résidu du raffinage de l'essence ou du diesel, utilisé pour la propulsion navale et dont la combustion relâche des quantités importantes de CO_2 , CH – 4 et N_2O , les principaux gaz responsables du réchauffement climatique. 12

C**Calepinage** — calepinage

Disposition d'éléments sur un plan 79

CAO — Conception Assistée par Ordinateur 17, 23**Cluster** — cluster

Regroupement de ressources informatiques (cartes graphiques, processeurs, RAM...) afin de former un supercalculateur, pouvant dépasser les limitation d'une simple machine. Un cluster peut-être divisé en plusieurs noeuds. iv, 29, 44, 46, 58–60, 88, 99, 107, 110

Coding — coding

Nomenclature 34, 38, 82

CP — Corner Panel

Élément de la technologie MARK III installé au niveau des DR. C'est une variante du FP. 80, 104

CPU — processeur

Central Processing Unit, littéralement « processeur ». Unité de calcul de base d'un ordinateur 47, 59, 60, 88, 92, 94, 106

CSA — Containment System A

Division de la SDP. 16, 17

CSB — Containment System B

Division de la SDP. 16, 17, 23

.csv — Comma-separated values

Format de stockage de données tabulaires sous forme de valeurs séparées par des virgules. 43, 58

CV — Computer Vision

Domaine de l'IA traitant des images et vidéos. Littéralement, « Vision par ordinateur » iv, 38, 72

D

Dataset — dataset

Littéralement, « jeu de données » iv, 34, 35, 37, 39–43, 46, 48, 68, 74

DB —

30, 34

DDP — Distributed Data Parallel

Stratégie de distribution de calculs où chaque unité de calcul (GPU) héberge une copie du modèle. x, 46–48, 71

DE — Définition

Type de plan définissant les différentes références (variations) de la pièce standard utilisées dans le projet. 30, 34, 68, 83

DL — Deep Learning

Sous-domaine de l'IA utilisant des réseaux de neurones comportant de nombreuses couches pour résoudre des tâches complexes. 38, 54, 72, 106, 109

DR — Dièdre

Intersection de 2 faces d'une cuve 78, 80, 103

DSI — Direction Informatique 17, 58, 59

E

EB — Erection-on-Board

Élément de la technologie MARK III installé installé entre deux CP et au niveau des TR 80

Éléments invariants — éléments invariants

Partie d'un plan pouvant être traité de manière systématique car sa position et son type peut-être inférée. Ces éléments peuvent-être du texte présent dans le cartouche, des tables... Voir également Éléments variants 20, 104

Éléments variants — éléments variants

Partie d'un plan ne pouvant être traité de manière systématique car ni position ni sa composition ne peut-être inférée. Ces éléments peuvent-être des dessins, des médaillons, des cotes... Voir également Éléments invariants 20, 24, 26, 38, 102, 104

Embeddings — embeddings

Représentation vectorielle d'un objet complexe (images, mot, phrase...) dans un sous-espace. Souvent interne à un réseau neuronal. 33, 34, 54, 63, 64, 70, 73

Epoch — epoch

Littéralement « époque ». Désigne une passe complète de toutes les données utilisées lors de l'entraînement. 49, 50, 61, 66, 94

ETI — Entreprise de Taille Intermédiaire

Entreprise de Taille Intermédiaire, catégorie située entre les PME et les grandes entreprises et caractérisée par une masse salariale comprise entre 249 et 4999 individus ainsi qu'un chiffre d'affaires inférieur à 1.5 milliards d'euros. 7

Explicabilité — explicabilité

Ensemble des méthodes et processus permettant de comprendre et de se fier aux résultats des algorithmes de Machine Learning. Autrement, les IA sont souvent assimilées à des « boîtes noires » iv, 27, 29, 40, 41, 69, 72, 109

F

FLOP — Floating Point Operations

Littéralement « opérations en virgule flottante », de manière simplifiée, opération d'addition ou de multiplication 95

FLOPS — Floating Point Operations per Second

Littéralement « nombre d'opérations en virgule flottante par seconde », unité de mesure de la rapidité de calcul d'un système informatique 95

FP — Flat Panel

Élément de la technologie MARK III installé sur les FW. C'est un panneau de bois comportant plusieurs apposé sur l'isolation secondaire et sur lequel sera fixé l'isolation primaire. Leur taille classique est de 3m par 1m. 80, 82, 103

FSDP — Fully Sharded Data Parallel

Stratégie de distribution de calculs où un modèle est partitionné sur plusieurs unités de calculs (GPU) x, 47, 48, 63, 71

FW — Flatwall

30, 34, 78, 80, 105

G

GD — Gas Dome

Zone spéciale de la technologie MARK permettant des échanges gazaux. 79

GNL — Gaz Naturel Liquéfié

Gaz condensé sous forme liquide afin d'être plus facilement transportable ; pour cela le gaz doit être refroidi à -161° dans le cas du méthane, sous pression atmosphérique : il occupe alors un volume 600 fois moindre 5–8, 11–13, 74, 80

GP — Gas Pocket

Zone spéciale de la technologie MARK permettant d'éviter la compression du gaz. 79

GPU — carte graphique

Graphical Processing Unit, dont la traduction est « carte graphique ». L'utilisation des GPU en DL est privilégiée aux CPU étant donné que l'on peut capitaliser sur la parallélisation des calculs matriciels. 46, 47, 59, 61, 88, 92, 94, 95, 104, 105, 110

Grad-CAM — grad-CAM

Type de saliency map iv, 52, 53

GTT — Gaztransport et Technigaz

Entreprise cliente iii, iv, x, xiii, 5–13, 16, 20, 22, 23, 25–27, 29, 35, 37, 42, 44, 49, 58, 60, 74, 76, 82, 84, 85, 102, 108, 109

H**Heatmap** — heatmap

Littéralement « carte thermique ». C'est une représentation graphique de données statistiques, faisant correspondre l'intensité d'une grandeur variable sur une matrice à deux dimensions à l'aide de couleurs, comme par exemple la température sur une carte 52, 53

Hexapode — hexapode

Simulateur composé de 6 actionneurs agissant sur une plateforme mobile commune. Ceci permet la mise en position d'objets suivant les 6 degrés de libertés (translations et rotations selon 3 axes). Ils sont notamment utilisés pour reproduire les conditions de séismes ou de marées. 7

HS — Handling System 16, 17**HSE** — Hygiène Sécurité Environnement 14**I****IA** — Intelligence Artificielle iv, 2–5, 10, 23–28, 32, 37, 40, 43, 46, 50, 74, 90, 102, 104, 105, 108**IMTA** — IMT Atlantique

École d'ingénieur où étudie l'auteur de ce rapport iv, xiii, 1, 60, 72

J

Jarvis —

Cluster de calculs interne à GTT basé sur Torque et MOAB et comportant notamment 4 noeuds de calculs chacun possédant 2 NVIDIA V100 avec 16 GB de RAM 44–46, 58–61, 72, 88, 95, 96, 101

JH — Jours-Homme

Unité de mesure correspondant au travail effectué par un homme en une journée. 26

Job — job

Tâche informatique 60, 88, 96–101

K**KPI** — Indicateur Clé de Performance

Un indicateur clé de performance est utilisé pour l'aide à la décision au sein des organisations et entreprises 11, 22, 26, 102

L**LD** — Liquid Dome

Zone spéciale de la technologie MARK permettant de relier la cuve à l'extérieur. 79

LG — Listing

Type de plan listant le nombre de chaque référence nécessaire sur l'ensemble du projet. 83

LLM — Large Language Model

Réseau neuronal acceptant des entrées textuelles, comme ChatGPT 73

M**MASE** — MASE

Réseau d'association délivrant des certifications SSE 4

ML — Machine Learning

TODO 3, 31, 34, 50, 54, 55, 59, 71, 101, 105

MOAB —

Programme d'ordonnancement de tâches informatiques utilisé avec Torque pour gérer un Cluster de calculs 46, 88, 107, 110

MP DATA —

Entreprise d'accueil iv, xiii, 2–5, 23–25, 31, 44, 60, 102

MS — Membrane Sheet

34, 80

Mtpa — Millions de tonnes par an

Unité de mesure utilisé dans le transport maritime 6

N

NaN — Not A Number

Littéralement, « pas un nombre », valeur attribuée au résultat d'un calcul lorsque celui-ci dépasse la plage imposée ou résulte d'une forme indéterminée mathématiquement parlant. 62

NLP — Natural Language Processing

Domaine de l'IA impliquant la linguistique et permettant de traiter le langage naturel. 54, 72

P

PBS — Portable Batch System

Programme d'ordonnancement de tâches informatiques développé à la demande de la NASA dans les années 90 ; une alternative étant SLURM 46, 60, 109, 110

PDM — Solidworks PDM

Coffre-fort numérique développé par Dassault Systèmes et utilisé par GTT 35, 37, 58

Pipelines CI/CD — Pipeline d'intégration en continu et de distribution en continu

Séries d'étapes automatisées alliant des pratiques d'intégration en continu (CI) et de distribution en continu (CD) afin de simplifier le développement d'applications 3

PR — Principe

Type de plan définissant une pièce de manière standard sous tous ses angles. 30, 34, 83

Pruning — pruning

Action de simplification d'un réseau neuronal en supprimant les connexions inutiles et redondantes 55

PTBS — Pump Tower Base Support

Zones spéciale de la technologie MARK traversant la cuve. 79

R

R&D — Recherche & Développement 6

Rastérisation — rastérisation

Procédé de conversion d'une image vectorielle en une image matricielle (ie constituée de pixels) 38

Réseau Siamois — réseau siamois

Type de réseau neuronal à plusieurs entrées utilisé majoritairement pour obtenir une mesure de différence ou de similarité entre ces dernières. iv, 32–34, 40–43, 50, 52, 53, 68, 71, 74, 91

Resnet — resnet

Réseau très connu en Deep Learning ayant significativement amélioré les performances des modèles à la pointe de la technologie grâce aux connexions résiduelles, raccourci entre deux couches permettant d'éviter le problème de disparition des gradients. 43, 53, 71, 74

RO — Recherche Opérationnelle

Ensemble des méthodes et techniques rationnelles permettant de trouver la solution optimale, ou à défaut la meilleure possible, à un problème donné 3

ROI — Retour sur Investissement 3**RSE** — Responsabilité Sociétale des Entreprises

Prise en compte par les entreprises, sur une base volontaire, et parfois juridique, des enjeux environnementaux, sociaux, économiques et éthiques dans leurs activités 4, 11

S**Saliency Map** — saliency map

Technique d'explicabilité sous forme d'image mettant en évidence les régions d'importance ayant mené le modèle à sa décision. 52, 53, 71, 106

SDP — Sous-Direction des Plans

Direction de GTT ayant la charge d'élaborer les plans pour les chantiers. C'est à cette direction que mon tuteur et moi sommes rattachés xiii, 15, 16, 25, 30, 103

Sloshing — sloshing 7, 79**SLURM** — Simple Linux Utility for Resource Management

Programme d'ordonnancement de tâches informatiques open source sur linux utilisé par la majorité des supercalculateurs de la planète ; une alternative étant PBS 46, 60, 108

Smart Shipping —

Gestion d'une flotte afin de diminuer ses coûts et son impact environnemental (optimisation des routes...) 10

SO — Side Opening

Zone spéciale de la technologie MARK temporaire permettant le transport de matériaux. 30, 79

SSE — Santé, Sécurité et Environnement 4, 107

Strobocheck — strobocheck

Méthode de vérification des erreurs afin d'éviter les ADC et ADGTT ; cette tâche monotone et peu intéressante doit-être remplacée par AutoCheck 22, 25, 26, 73, 102

SVG — Scalable Vector Graphics

Format de stockage de données sous forme vectorielle possédant de fait la particularité de pouvoir être agrandi ou rapetissé sans perte de qualité 33, 38

T**TC** —

Zone spéciale de la technologie NO non-explicitée ici 30

Torque — Terascale Open-source Resource and Queue Manager

Programme gérant la distribution des ressources, le démarrage et l'arrêt des tâches informatiques des noeuds d'un Cluster de calculs ; basé sur PBS et open-source ; utilisé conjointement avec MOAB pour gérer un cluster de calcul 46, 107

TR — Trièdre

Intersection de 3 faces d'une cuve 79, 80, 104

V**VBA** — Visual Basic for Applications

Language informatique obsolète et limité mais intégré dans les applications de Microsoft Office et certaines autres comme Solidworks. 58

VM — Machine Virtuelle 58**VRAM** — Mémoire Vidéo

Virtual Random Access Memory, dont la traduction est « mémoire vidéo ». Liée au GPU, c'est ce qui lui permet de stocker les données à traiter 46, 47

Bibliographie

- [1] Visualisation des rapports annuels de l'ong shipbreaking platform. URL <https://www.offthebeach.org/>.
- [2] Les importations de gnl en europe en 2022. 2023. URL https://www.connaissancesdesenergies.org/importations-de-gnl-en-europe-un-nouveau-tracker-et-des-inquietudes-230323?utm_source=newsletter&utm_medium=fil-info-energies&utm_campaign=newsletter/cde-aujourd'hui-20-juin-2023.
- [3] Les importations de gnl en europe en 2024. 2024. URL <https://ieefa.org/articles/european-lng-import-terminals-are-used-less-demand-drops>.
- [4] Rhonda Ascierto. Uptime institute global data center survey, 2018. URL <https://datacenter.com/wp-content/uploads/2018/11/2018-data-center-industry-survey.pdf>.
- [5] Gabor BELLA. Ue g : Nlp and text mining, 2024. URL <https://moodle.imt-atlantique.fr/mod/resource/view.php?id=68813>. Moodle, [Online].
- [6] Pancevski Bojan. The real story of the nordstream pipeline sabotage. *Wallstreet Journal*, août 2024. URL https://www.wsj.com/world/europe/nord-stream-pipeline-explosion-real-story-da24839c?mod=europe_trendingnow_article_pos1. Consulté en août 2024.
- [7] Josiah A. Bryan. Automatic grading software for 2d cad files. *Computer Applications in Engineering Education*, 28(1) :51–61, 2020. doi : <https://doi.org/10.1002/cae.22174>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22174>.
- [8] D. Cao, Z. Wang, J. Echevarria, and Y. Liu. Svgformer : Representation learning for continuous vector graphics using transformers. In *2023 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), pages 10093–10102, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society. doi : 10.1109/CVPR52729.2023.00973.
URL <https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.00973>.

- [9] NVIDIA Corporation. Nvidia v100 specifications.
<https://images.nvidia.com/content/technologies/volta/pdf/volta-v100-datasheet-update-us-1165301-r5.pdf>, 2020. Accessed : 2024-09-01.
- [10] Dries Van Daele, Nicholas Decleyre, Herman Dubois, and Wannes Meert. An automated engineering assistant : Learning parsers for technical drawings. *CoRR*, abs/1909.08552, 2019. URL <http://arxiv.org/abs/1909.08552>.
- [11] Mathilde Damgé. Cop27 : le fret maritime est l'un des plus grands émetteurs de co2, et il tarde à changer de cap, 2023. URL
https://www.lemonde.fr/les-decodeurs/article/2022/11/11/cop27-le-transport-maritime-un-secteur-polluant-qui-tarde-a-changer-de-cap_6149485_4355770.html.
- [12] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. Compute and energy consumption trends in deep learning inference. *CoRR*, abs/2109.05472, 2021. URL <https://arxiv.org/abs/2109.05472>.
- [13] Justine Dumont. Quelle est l'empreinte carbone du transport maritime ?, 2023. URL
<https://greenly.earth/fr-fr/blog/actualites-ecologie/quelle-est-l-empreinte-carbone-du-transport-maritime>.
- [14] Benedikt Faltin, Damaris Gann, and Markus König. A comparative study of deep learning models for symbol detection in technical drawings. pages 877–886, 01 2023. ISBN 979-12-215-0289-3. doi : 10.36253/979-12-215-0289-3-87.
- [15] Yi-Hsin Lin, Yu-Hung Ting, Yi-Cyun Huang, Kai-Lun Cheng, and Wen-Ren Jong. Integration of deep learning for automatic recognition of 2d engineering drawings. *Machines*, 11(8), 2023. ISSN 2075-1702. doi : 10.3390/machines11080802. URL
<https://www.mdpi.com/2075-1702/11/8/802>.
- [16] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit : Memory efficient vision transformer with cascaded group attention, 2023. URL <https://arxiv.org/abs/2305.07027>.
- [17] Alexandra Sasha Luccioni, Sylvain Viguer, and Anne-Laure Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model, 2022. URL
<https://arxiv.org/abs/2211.02001>.

- [18] Haitao Luo, Jinming Zhang, Xiongfei Liu, Lili Zhang, and Junyi Liu. Large-scale 3d reconstruction from multi-view imagery : A comprehensive review. *Remote Sensing*, 16 (5), 2024. ISSN 2072-4292. doi : 10.3390/rs16050773. URL <https://www.mdpi.com/2072-4292/16/5/773>.
- [19] Tobias Schlagenhauf, Markus Netzer, and Jan Hillinger. Text detection on technical drawings for the digitization of brown-field processes, 2022. URL <https://arxiv.org/abs/2205.02659>.
- [20] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam : Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128 (2) :336–359, October 2019. ISSN 1573-1405. doi : 10.1007/s11263-019-01228-7. URL <https://arxiv.org/pdf/1610.02391>.
- [21] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp, 2019. URL <https://arxiv.org/abs/1906.02243>.
- [22] Mingxing Tan and Quoc V. Le. Efficientnet : Rethinking model scaling for convolutional neural networks, 2020. URL <https://arxiv.org/abs/1905.11946>.
- [23] Javier Villena Toro, Anton Wiberg, and Mehdi Tarkian. Optical character recognition on engineering drawings to achieve automation in production quality control. *Frontiers in Manufacturing Technology*, 3, 2023. ISSN 2813-0359. doi : 10.3389/fmtec.2023.1154132. URL <https://www.frontiersin.org/journals/manufacturing-technology/articles/10.3389/fmtec.2023.1154132>.
- [24] Liuyue Xie, Yao Lu, Tomotake Furuhata, Soji Yamakawa, Wentai Zhang, Amit Regmi, Levent Kara, and Kenji Shimada. Graph neural network-enabled manufacturing method classification from engineering drawings. *Computers in Industry*, 142 :103697, 2022. ISSN 0166-3615. doi : <https://doi.org/10.1016/j.compind.2022.103697>. URL <https://www.sciencedirect.com/science/article/pii/S016636152200094X>.
- [25] Muhammad Yazed, Ezak Shaubari, and Moi Yap. A review of neural network approach on engineering drawing recognition and future directions. *JOIV : International Journal on Informatics Visualization*, 7 :2513, 12 2023. doi : 10.30630/joiv.7.4.01716.
- [26] Wentai Zhang, Joe Joseph, Yue Yin, Liuyue Xie, Tomotake Furuhata, Soji Yamakawa, Kenji Shimada, and Levent Burak Kara. Component segmentation of engineering drawings using graph convolutional networks. *Computers in Industry*, 147 :103885, May 2023. ISSN 0166-3615. doi : 10.1016/j.compind.2023.103885. URL

<http://dx.doi.org/10.1016/j.compind.2023.103885>.

- [27] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. 12 2016. doi : 10.1109/CVPR.2016.319. URL <https://arxiv.org/pdf/1512.04150.pdf>.

Colophon

THIS REPORT WAS TYPESET using L^AT_EX, was first modified by AYARI Anis, later by PERRIN–DELORT Antoine and was originally developed by Leslie Lamport and based on Donald Knuth's for Harvard T_EX. The body text is set in 11 point Arial, designed by Microsoft. A template, which can be used to format a IMTA report, has been released under the permissive MIT (x11) license, and can be found online at github.com/anisayr/IMTA-report-latex or from the author at anis.ayari@imt-atlantique.net