

**Intitulé : IA pour une imagerie médicale enrichie et non-invasive**

**Université de Poitiers Année universitaire -**

**SP2MI, UFR Sciences Fondamentales et Appliquées**

**Master Objets Connectés**

**Rapport de stage**

**Adressé pour la validation du Master : Ingénierie des objets intelligents**

**Intelligence artificiel pour une imagerie médicale enrichie et non-invasive**

Période de stage : mars - septembre

Soumis par

Marwan AL OMARI **Tuteur de stage**

BOURDON Pascal

Laboratoire XLIM CNRS

Bât. H - SP2MI

Enseignant référent : Bd Marie et Pierre Curie

Noël Richard Poitiers Cedex

**Table des matières**

[**Liste des figures** II](#_Toc110947532)

[**Liste des tableaux** V](#_Toc110947533)

[**Résumé** VI](#_Toc110947534)

[**Abréviations et acronymes** VI](#_Toc110947535)

[**I. Laboratoires et présentations de stage** IX](#_Toc110947536)

[**I.1 Laboratoire XLIM** X](#_Toc110947537)

[**I.2 Laboratoire I3M** X](#_Toc110947538)

[**I.3 Présentation des stage** XI](#_Toc110947539)

[**I.3.1 Cahier des charges** XII](#_Toc110947540)

[**I.3.2 Description** XII](#_Toc110947541)

[**I.4 Gestion de projet** XIII](#_Toc110947542)

[**I.4.1 Les systèmes d'exploitation** XIII](#_Toc110947543)

[**I.4.2 Programmes de logiciels** XIII](#_Toc110947544)

[**I.4.3 Plateformes de codage** XVI](#_Toc110947545)

[**1. Introduction** 16](#_Toc110947546)

[**2. Travaux relatifs** 2](#_Toc110947547)

[**3. Background** 8](#_Toc110947548)

[**3.1 L'apprentissage profond dans le diagnostic médical** 8](#_Toc110947549)

[**3.2 CNNs** 2](#_Toc110947550)

[**3.2.1 Convolution (C1, C3, and C5)** 3](#_Toc110947551)

[**3.3.2 Couche de Pooling (S2 and S4)** 2](#_Toc110947552)

[**3.3.3 FCL (F6)** 4](#_Toc110947553)

[**3.3.4 Fonctions d'activation** 4](#_Toc110947554)

[**3.3 U-Net** 2](#_Toc110947555)

[**3.4 GANs** 4](#_Toc110947556)

[**3.5 U-Net et GAN** 6](#_Toc110947557)

[**3.5 Jeu de données** 6](#_Toc110947558)

[**4. Approache** 7](#_Toc110947559)

[**4.1 Pré-traitement** 1](#_Toc110947560)

[**4.1.1 Recadrage des images** 1](#_Toc110947561)

[**4.1.2 One-hot encoding** 2](#_Toc110947562)

[**4.1.3 Normalization** 2](#_Toc110947563)

[**4.1.4 Filtration** 3](#_Toc110947564)

[**4.1.5 Correction de l'intensité linéaire** 3](#_Toc110947565)

[**4.1.6 Correction du biais** 3](#_Toc110947566)

[**4.1.7 Augmentation des données** 3](#_Toc110947567)

[**4.1.8 Répartition des données** 3](#_Toc110947568)

[**4.2 Algorithme** 4](#_Toc110947569)

[**4.2.1 SIMO** 5](#_Toc110947570)

[**4.2.2 MIMO** 5](#_Toc110947571)

[**4.2.3 MISO** 5](#_Toc110947572)

[**4.3 Evaluation Metrics** 6](#_Toc110947573)

[**5. Expériences et résultats** 6](#_Toc110947574)

[**6. Conclusion** 19](#_Toc110947575)

[**6.1 Professionnel** 19](#_Toc110947576)

[**6.2 Personnel** 19](#_Toc110947577)

[**6.3 Travaux futurs** 19](#_Toc110947578)

[**6.4 Challenges** 19](#_Toc110947579)

[**6.5.2 Utilisateur administratif non subventionné** 20](#_Toc110947580)

[**6.5.3 L'ambiguïté du cahier des charges** 20](#_Toc110947581)

[**6.5.3 Entraînement prolongé des architectures DL** 21](#_Toc110947582)

[**7. Références** 21](#_Toc110947583)

[**8. Annexe** 24](#_Toc110947584)

[**8.1 Ensembles de données :** 24](#_Toc110947585)

[**8.2 DLOps** 24](#_Toc110947586)

[**8.2.1 Modèle de SIMO** 24](#_Toc110947587)

# **Liste des figures**

[**Figure 1**- Représentation du laboratoire XLIM, qui se compose de 3 pôles et de 6 axes scientifiques X](#_Toc110947349)

[**Figure 2**- Images FLAIR représentatives d'un sujet unique à trois intensités de champ : 1,5 T, 3 T et 7 T, extraites de [23] XI](#_Toc110947350)

[**Figure 3**- Diagramme de GANT - gestion du temps des tâches XIII](#_Toc110947351)

[**Figure 4**- Graphique PERT classique - gestion du temps des tâches XIV](#_Toc110947352)

[**Figure 5**- Établissement d'une connexion SSH avec le serveur en question pour la manipulation SFTP XV](#_Toc110947353)

[**Figure 6**- Connexion SFTP à l'aide de FileZilla XVI](#_Toc110947354)

[**Figure 7**- Méthodes SISO, SIMO, MISO et MIMO dans le traitement des modalités IRM 2](#_Toc110947355)

[**Figure 8**- LeNet-5, un exemple d'architecture CNN, extrait de [21] 3](#_Toc110947356)

[**Figure 9**- Une déconstruction de l'opération convolutive dans les CNNs 2](#_Toc110947357)

[**Figure 10**- Exemple de suréchantillonnage 2](#_Toc110947358)

[**Figure 11**- Exemple de max-pooling avec fliter 2×2 et stride 2×2 3](#_Toc110947359)

[**Figure 12**- Exemple de pooling moyen avec filtre 2×2 et stride 2×2 3](#_Toc110947360)

[**Figure 13**- Exemple de Max-pooling global avec une taille de pool égale à la taille d'entrée 3](#_Toc110947361)

[**Figure 14**- Exemple de pooling moyen global avec une taille de pool égale à la taille d'entrée 4](#_Toc110947362)

[**Figure 15**- Fonction ReLU en représentation graphique 4](#_Toc110947363)

[**Figure 16**- Fonction LeakyReLU en représentation graphique 5](#_Toc110947364)

[**Figure 17**- Fonction sigmoïde en représentation graphique 2](#_Toc110947365)

[**Figure 18**- Architecture du réseau U-Net 2](#_Toc110947366)

[**Figure 19**- L'architecture U-Net, qui est extraite de [19][20] 3](#_Toc110947367)

[**Figure 20**- Le concept des GANs 4](#_Toc110947368)

[**Figure 21**- Démonstration de la cartographie générateur-discriminateur, extraite de [13]. 5](#_Toc110947369)

[**Figure 22**- L'architecture du GAN 6](#_Toc110947370)

[**Figure 23**- L'architecture d'assemblage de U-Net et des GANs 6](#_Toc110947371)

[**Figure 24** – Echantillon de données illustrant les quatre modalités avec carte de segmentation (vérité terrain) 7](#_Toc110947372)

[**Figure 25**- Architecture du programme, spécifiant les différentes étapes de la saisie des données, du prétraitement, de l'augmentation, de l'algorithme, des résultats et des mesures d'évaluation. Note : le processus marqué en rouge a été suspendu sur les architectures finales 8](#_Toc110947373)

[**Figure 26** – Le résultat de l'étape de prétraitement, y compris la correction des biais, la normalisation, le filtrage et le recadrage. 1](#_Toc110947374)

[**Figure 27**- Recadrage de l'image de 128×128 à 80×80, mettant en évidence la zone de la tumeur. 2](#_Toc110947375)

[**Figure 28**- Augmentation des données, y compris : des rotations à 90° et 180° et des retournements de gauche à droite. 3](#_Toc110947376)

[**Figure 29**- Données divisées en ensembles de test et de formation pour les prototypes SIMO (T1 → T2, FLAIR et T1c), MISO (T1 et T2 → FLAIR) et MISO (T1, T2 et FLAIR → T1c) avec une taille d'entrée d'image de 128×128. 4](#_Toc110947377)

[**Figure 30**- Données divisées en ensembles de test et de formation pour le prototype MISO (T1, T2 et FLAIR → T1c) avec un recadrage de 80×80 à l'entrée de l'image. 4](#_Toc110947378)

[**Figure 31**- L'architecture SIMO prend T1 en entrée, et synthétise T2, FLAIR, T1c, et la carte de segmentation pour la discrimination 5](#_Toc110947379)

[**Figure 32**- L'architecture MIMO prend T1 et T2 en entrée, et synthétise T1c et FLAIR pour la discrimination 5](#_Toc110947380)

[**Figure 33**- MISO prend T1 et T2 en entrée, et synthétise FLAIR pour la discrimination 6](#_Toc110947381)

[**Figure 34**- MISO prend T1 et T2, et FLAIR en entrée, et synthétise T1c pour la discrimination 6](#_Toc110947382)

[**Figure 35**- Score de dés U-Net, une comparaison entre SIMO, MISO (sortie FLAIR), MISO (sortie T1c 128×128), et MISO (sortie T1c 80×80), 18](#_Toc110947383)

[**Figure 36**- Matrice de probabilité/gravité du risque de non-disponibilité de l'ensemble de données 20](#_Toc110947384)

[**Figure 37**- Matrice de probabilité/gravité du risque de non-subventionnement de l'utilisateur de l'administration 20](#_Toc110947385)

[**Figure 38**- Matrice de probabilité/gravité du risque d'ambiguïté des spécifications 21](#_Toc110947386)

[**Figure 39**- Matrice de probabilité/sévérité du risque de formation prolongée des modèles 21](#_Toc110947387)

[**Figure 40**- Le modèle SIMO, en tant que microservice, accessible sur le service local pour le débogage 127.0.0.1:5000 24](#_Toc110947388)

[**Figure 41**- Le modèle SIMO, en tant que microservice, accessible sur le service local pour le débogage 127.0.0.1:5000 25](#_Toc110947389)

[**Figure 42**- Le modèle SIMO, en tant que microservice, accessible sur le service local pour le débogage 127.0.0.1:5000 25](#_Toc110947390)

# **Liste des tableaux**

[**Tableau 1**- Diagramme PERT - gestion des tâches XIII](#_Toc110947133)

[**Tableau 2**- Résumé des articles en ce qui concerne le numéro de l'étude, le jeu de données, l'approche utilisée, les résultats de l'expérience, les défis et les travaux futurs possibles. 5](#_Toc110947134)

[**Tableau 3**- BRATS2018 modality distribution over 285 patients 7](#_Toc110947135)

[**Tableau 4**- One-hot encoding des étiquettes de tumeur et de non-tumeur. 2](#_Toc110947136)

[**Tableau 5**- Spécifications de l'ordinateur de laboratoire 6](#_Toc110947137)

[**Tableau 6**- Score de dés U-Net des 4 architectures, SIMO a atteint 0,8708 ACC, MISO (sortie FLAIR) a atteint 0,4140, MISO (sortie T1c 128×128) a atteint 0,4147, et MISO (sortie T1c 80×80) a atteint ‼‼ 17](#_Toc110947138)

**Remerciements**

J'adresse mes sincères remerciements tout d'abord à mon superviseur principal Pascal BOURDON, et à ma co-superviseuse Christine FERNANDEZ pour le suivi constant et les instructions directives tout au long du projet. J'aimerais également remercier mon tuteur Noël RICHARD pour son enseignement sur les différents aspects de la gestion de projet. Je dois également remercier le responsable du programme de master, Clency PERRINE, pour ses conseils et son organisation ferme. De plus, j'aimerais remercier le personnel d'XLIM, car j'ai eu l'occasion d'y travailler et d'établir de nouvelles connexions.

Enfin, j'aimerais adresser mes sincères remerciements à Campus France pour avoir financé mes études de master et une année entière de langue française en CFLE (Centre de Français Langue Etrangère) à Poitiers. J'apprécie vraiment l'opportunité de changement de vie qui m'a été donnée, à Poitiers, en France. En outre, je n'oublierais pas ma famille, qui est restée à mes côtés et m'a soutenu dans les moments difficiles pendant mes études de master ainsi que pendant mon stage. Je dois mentionner que je n'avais pas beaucoup de connaissances en langue française au début du programme de master. Je me souviens encore des jours que je passais à être bloquée, sans aucune avancée, j'essayais de comprendre quelques mots de la langue. Mais comme tout le reste dans ce monde, ça s'améliore au fur et à mesure que le temps passe.

# **Résumé**

L'intelligence artificielle a progressé au cours des dernières années, montrant des capacités accrues dans le traitement des données pour diverses applications de la vision par ordinateur, notamment la classification, la segmentation, la détection, etc. Notamment, le réseau neuronal génératif a prouvé son efficacité dans la capture de caractéristiques plus profondes au cours du processus de synthèse d'images. Sur la base de l'état de l'art, les réseaux de générations sont compétitifs par rapport aux algorithmes traditionnels d'apprentissage automatique, dans l'apprentissage de structures similaires entre les images d'entrée. Par conséquent, ce projet de recherche développe des architectures améliorées à une entrée unique et multiple et à sortie unique dans le but de synthétiser l'imagerie médicale. Ces architectures, dont SIMO (sortie T, FLAIR, Tc), MISO (sortie FLAIR), MISO (sortie Tc ), et enfin MISO (sortie T1c ), sont importantes, car elles permettent de gagner beaucoup de temps sur le diagnostic des tumeurs et elles permettraient d'éviter l'injection d'une substance supplémentaire, par exemple du gadolinium, dans le cerveau lors de l'extraction de la modalité Tc. Les modèles sont entraînés et expérimentés en utilisant des techniques d'hyper-optimisation pour affiner les paramètres sur le jeu de données BRATS. Ensuite, les architectures ont obtenu des résultats compétitifs en mettant en évidence les performances de l'état de l'art.

**Mots-clés :** Intelligence artificielle, réseaux neuronaux génératifs, apprentissage profond, synthèse médicale

# **Abréviations et acronymes**

Dans cette section, les abréviations et acronymes utilisés dans ce rapport sont clarifiés afin d'éviter la répétition des mêmes termes. Ces termes sont les suivants :

***M****achine* ***L****earning* (ML)

***D****eep* ***L****earning* (DL)

***S****ingle-****I****nput* ***S****ingle-****O****utput* (SISO)

***M****ulti-****I****nput* ***S****ingle-****O****utput* (MISO)

***M****ulti-****I****nput* ***M****ulti-****O****utput* (MIMO)

***S****ingle-****I****nput* ***M****ulti-****O****utput* (SIMO)

**C**entre **H**ospitalier **U**niversitaire (CHU)

**I**magerie **M**étabolique **M**ulti-noyaux **M**ulti-organes (IM)

***BRA****in* ***T****umor* ***S****egmentation Challenge*  (BRATS)

***BRA****in* ***T****umor* ***S****egmentation Challenge*  (BRATS)

***B****rain* ***T****umor* ***S****egmentation* ***C****hallenge* (BRATS)

*The* ***Celeb****Faces* ***A****ttributes* (CelebA)

*The* ***Ra****dboud* ***F****aces* ***D****atabase* (RaFD)

***T****-weighted* (T)

***T*** *and* ***C****ontrast-enhance* (Tc)

***T****-weighted* (T)

***F****luid-****A****ttenuated* ***I****nversion* ***R****ecovery* (FLAIR)

***H****igh* ***G****rade* ***G****lioma* (HGG)

***L****ower* ***G****rade* ***G****lioma* (LGG)

***C****onvolutional* ***N****eural* ***N****etwork****s*** (CNNs)

***F****ully-****C****onnected* ***L****ayers* (FCL)

***G****enerative* ***A****dversarial* ***N****etwork* (GAN)

***G****enerative* ***A****dversarial* ***N****etwork****s*** (GANs)

**C**entre **N**ational de la **R**echerche **S**cientifique (CNRS)

***C****onvolutional Patched****GANs*** (CGANs)

***A****rtificial* ***N****eural* ***N****etwork****s*** (ANNs)

***D****eep* ***N****eural* ***N****etwork****s*** (DNNs)

***R****ecurrent* ***N****eural* ***N****etwork****s*** (RNNs)

***Re****ctified* ***L****inear* ***U****nit* (ReLU)

***CO****rona****VI****rus* ***D****isease*  (Covid-)

***C****ross* ***V****alidation* (CV)

***ADA****ptive* ***M****oment estimation* (ADAM)

**I**magerie par **R**ésonance **M**agnétique nucléaire (IRM)

**T**raitement **A**utomatique des **L**angues(TAL)

***ACC****uracy* (ACC)

**I**ntelligence **A**rtificiel (IA)

***B****lack and* ***W****hite* (BW)

Taux d'apprentissage (TA)

Technique d'evaluation et d'examen de programmes**/*P****rogram* ***E****valuation and* ***R****eview* ***T****echnique* (PERT)

***M****achine* ***T****ranslation* (MT)

***S****ecure* ***SH****ell* (SSH)

***S****SH* ***F****ile* ***T****ransfer* ***P****rotocol* (SFTP)

***S****treet* ***V****iew* ***H****ouse* ***N****umber* (SVHN)

***N****ormalized* ***M****ean* ***A****bsolute* ***E****rror* (NMAE)

***P****eak* ***S****ignal-to-****N****oise* ***R****atio* (PSNR)

***S****ignal-to-****N****oise* ***R****atio* (SNR)

***S****tructural* ***S****imilarity* ***I****ndex* ***M****easurement* (SSIM)

***V****isual* ***I****nformation* ***F****idelity* (VIF)

***N****aturalness* ***I****mage* ***Q****uality* ***E****valuator* (NIQE)

***V****ariational* ***A****uto****E****ncoder****s*** (VAEs)

***C****ycle-****C****onsistency* (CC)

***A****mazon* ***M****echanical* ***T****urk* (AMT)

***I****ntersection-****O****ver-****U****nion* (IOU)

***C****omputed* ***T****omography* (CT)

***M****ean-****S****quared* ***E****rror* (MSE)

***D****eep* ***L****earning* ***OP****eration****s*** (DLOps)

***C****entral* ***P****rocessing* ***U****nit****s*** (CPUs)

***G****raphics* ***P****rocessing* ***U****nit****s*** (GPUs)

***R****andom* ***A****ccess* ***M****emory* (RAM)

***S****hared* ***W****ireless* ***A****ccess* ***P****rotocol* (SWAP)

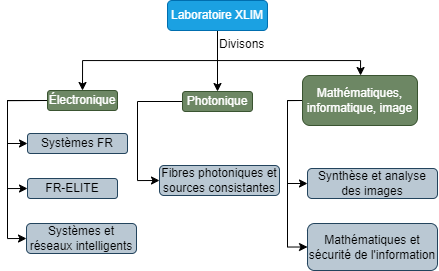
# **I. Laboratoires et présentations de stage**

Dans cette section, une brève introduction est représentée sur les laboratoires XLIM et IM ainsi que leurs différentes spécialités de domaine. Ensuite, une présentation générale du poste de stage sera abordée avec ses objectifs souhaités et les missions requises, et surtout la gestion du projet.

## **I.1 Laboratoire XLIM**

XLIM, numéroté , est un des centres de recherche français, connu sous le nom de CNRS . Le laboratoire est largement centré sur l'électronique, l'optique, la photonique, les mathématiques, l'informatique, le traitement d'images, les télécommunications, la sécurité des réseaux, la bio-ingénierie et l'énergie. Une représentation des divisions de XLIM est présentée ci-dessous dans la figure ().

**Figure 1**- Représentation du laboratoire XLIM, qui se compose de pôles et de axes scientifiques



XLIM est un institut de recherche pluridisciplinaire, implanté sur plusieurs sites géographiques, à Limoges sur les sites de la Faculté des Sciences et Techniques, de l'ENSIL, d'Ester-Technopole, sur le Campus universitaire de Brive ; et sur le site de la Technopole du Futuroscope à Poitiers. Il rassemble plus de enseignants-chercheurs, chercheurs CNRS, ingénieurs, techniciens administratifs, doctorants et post-doctorants.

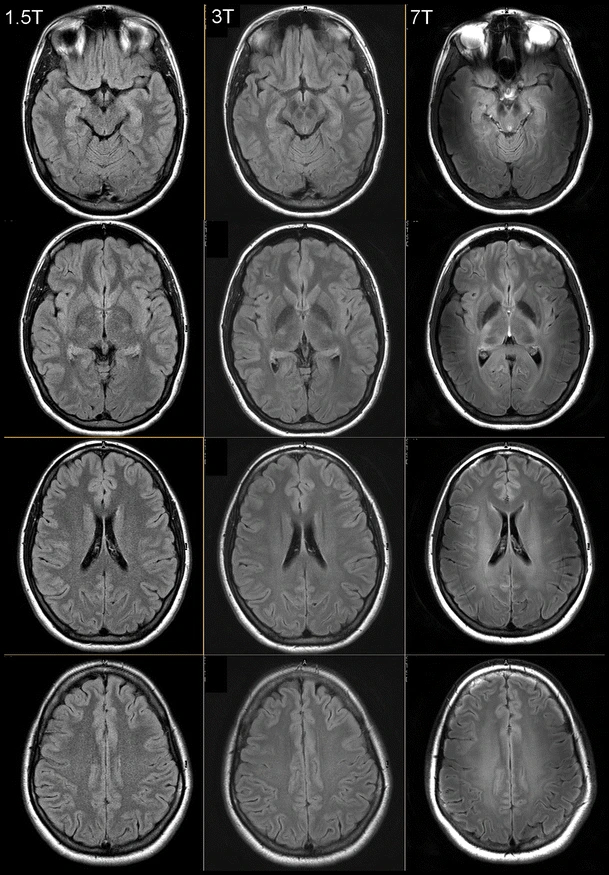
La recherche principale d'XLIM s'appuie principalement sur deux plateformes. D'une part, une plateforme permet aux chercheurs d'accéder à des équipements technologiques pour la création de structures optiques. Par exemple, un grand nombre de lasers et d'instruments sont disponibles pour la caractérisation expérimentale de dispositifs électroniques, optiques, électromagnétiques et radiants. D'autre part, le laboratoire s'engage dans des activités de modélisation et de simulation car il est équipé d'une forge logicielle, qui est à la disposition de ses membres et de leurs collaborateurs externes. Pour cela, il propose une simulation multi-échelle de systèmes complexes basée sur des modèles d'interface entre la physique, les mathématiques et l'informatique.

## **I.2 Laboratoire IM**

IM a été fondé en février , une collaboration entre le CNRS, SIEMENS HEALTHINNERS, le CHU de Poitiers et l'Université de Poitiers . C'est un centre de recherche transdisciplinaire distinctif basé sur l'imagerie métabolique des organes en France et même en Europe. Le laboratoire utilise le traitement automatique et l'analyse multimodale avec l'aide de l'IA pour l'objectif d'images IRM multivariées ainsi que l'aide au suivi diagnostique et thérapeutique dans les maladies du cerveau, du cœur et des reins. Par ailleurs, l'IM utilise une IRM à ultra-haut champ de Tesla, considérés comme la e du monde entier .

L'IRM Tesla permet d'accéder à une imagerie de très haute résolution des molécules et du métabolisme, en plus de l'anatomie et de la fonction des organes. Elle permet de mesurer la fonction et la structure des organes d'une manière inégalée. Par exemple, la figure () représente une comparaison d'images FLAIR entre T, T et T d'un même sujet. Par conséquent, les images FLAIR, capturées par le T, ont un SNR plus élevé et un contraste bien meilleur par rapport aux autres générations d'IRM Tesla.

**Figure 2**- Images FLAIR représentatives d'un sujet unique à trois intensités de champ : T, T et T, extraites de []



## **I.3 Présentation de stage**

Pendant mon stage, j'ai été un ingénieur de recherche, travaillant sur le développement et l'expérimentation d'architectures DL pour résoudre le problème de la génération de différentes séquences IRM, principalement sur la génération de fausses modalités sur BRATS. J'ai acquis de nombreuses compétences, qui seront ma prochaine source de connaissances et de développement de stratégies à l'avenir. Certaines des compétences que j'ai acquises sont résumées dans ce qui suit :

1. Gestion du réseau et programmation d'un ordinateur à distance en utilisant SSH et SFTP
2. Prototypage d'un algorithme pour l'amélioration de la recherche.
3. Planification et autogestion de projets

Le stage se déroule dans deux laboratoires différents et à des intervalles de temps différents :

* XLIM : présence du mars au juillet et du août au er septembre.
* IM : présence du juillet au août.

### **I.3.1 Cahier des charges**

Les principales spécifications des stages sont les suivantes :

1. Développement d'une solution pour la construction de séquences IRM à l'aide d'algorithmes d'intelligence artificielle.
2. Réalisation d'un prototype en langage Python.

Dans la section suivante, je vais décrire et donner des détails supplémentaires sur les missions durant toute la période de stage.

### **I.3.2 Description**

Pour commencer, j'ai été chargé de découvrir le jeu de données BRATS dans un contexte médical, en dehors de sa complexité structurelle et représentationnelle. Ensuite, j'ai l’analysé en effectuant différentes manipulations à l'aide du langage de programmation Python. Par exemple, j'ai découvert les données plus en détail avec des graphiques de représentation en utilisant la visualisation Matplotlib.

En outre, j'ai développé des architectures de DL comme SIMO et MISO, en utilisant des GANs et des algorithmes U-Net, pour prédire certaines modalités à partir d'une ou plusieurs entrées. Par conséquent, j'ai identifié les solutions et les algorithmes les plus récents pour améliorer les résultats, y compris les méthodes de prétraitement et les structures d'algorithme. J'ai expérimenté différentes méthodes pour affiner l'algorithmes GANs, y compris le pipeline et les tests d'échantillons.

Lors des réunions en face-à-face et sur le web, je devais présenter les résultats obtenus à chaque étape ; ainsi, à travers ces présentations, ma langue française s'améliorait régulièrement voilà que je rédigeais des rapports hebdomadaires et mensuels et des courriels à mon superviseur ainsi qu'à mon superviseur pédagogique.

En plus des tâches techniques, mes qualifications, dans le cadre du programme de stage en imagerie médicale d'XLIM et IM, se sont massivement développées en matière de communication, d'organisation, d'autonomie de recherche, de vision analytique, de curiosité et de créativité. J'ai appris à présenter des idées avec de bonnes capacités de communication, tant à l'oral qu'à l'écrit. Néanmoins, l'écriture n'est pas exempte de limites bien que j'aie fait de mon mieux pour garder les écrits sans erreurs.

Dans les bases quotidiennes, j'apprenais de nouvelles choses qui avaient façonné mon esprit dans la science des données et l'IA. Je pense avoir prouvé que j'étais un membre essentiel en apportant des idées révolutionnaires et des solutions innovantes pour améliorer la synthèse d'images. Cependant, j'ai été confronté à de nombreux problèmes pour mettre en place l'environnement d'entraînement des algorithmes, car je me déplace d'un laboratoire à l'autre et j'ai à chaque fois un équipement différent à utiliser, sans droit d'administration. Je pourrais dire que tout dépend d'une hiérarchie donc si je voulais quelque chose, comme installer un programme, je dois l'attendre car je n'ai aucune autorité sur l'ordinateur. Ainsi, pour la première phase, j'ai découvert que l'utilisation de mon ordinateur personnel permettrait de gagner beaucoup de temps sur le développement des différents prototypes d'architectures GANs.

Néanmoins, j'ai été un peu limité dans mon ordinateur personnel, qui se plantait lors de l'entraînement avec une énorme quantité de données. La spécification de mon ordinateur, détaillé dans la tableau () suivant :

**Tableau 1**- Spécifications de l'ordinateur personnelle×

|  |  |
| --- | --- |
| **CPUs** | Intel Core i7-10750H processor (Hexa-Core 2.6 GHz/5 GHz Turbo - 12 Threads - Cache 12M) |
| **RAM** | 16 GB of 2666 MHz DDR4 RAM (2x8 GB) |
| **GPUs** | Nvidia GeForce RTX 2060 graphics chip with 6 GB of dedicated GDDR6 memory |

Plus tard, j'ai eu accès à un serveur à distance équipé de Quadro RTX, sur lequel j'ai effectué toutes les expérimentations. Enfin, j'ai effectué les dernières expérimentations sur un autre serveur à distance équipé de Nvidia Tesla.

En ce qui concerne l'importance, j'ai constaté que l'organisation est l'un des éléments essentiels de la gestion du temps et du traitement des différentes tâches à chaque étape de l'avancement. J'ai remarqué que la science des données et la DL requièrent un esprit dynamique et analytique pour visualiser des solutions adaptées tout en couvrant un large éventail de littérature dans les travaux connexes en imagerie médicale. En pratique, la revue de la littérature a élargi mes horizons pour relever certains défis et tirer des leçons de ce que d'autres chercheurs ont vécu.

En somme, j'ai vraiment obtenu de grandes connaissances sur les clés et les capacités potentielles de la synthèse d'images grâce à l'analyse des données et au développement algorithmique. Je crois qu'elles se manifestent rarement dans les laboratoires de recherche français, surtout après la période difficile de la pandémie de Covid-.

### **I.4 Gestion de projet**

Dans cette section, je vais présenter les logiciels que j'ai utilisés pendant le stage, en commençant par les systèmes d'exploitation, les programmes et les plateformes de codage.

### **I.4.1 Les systèmes d'exploitation**

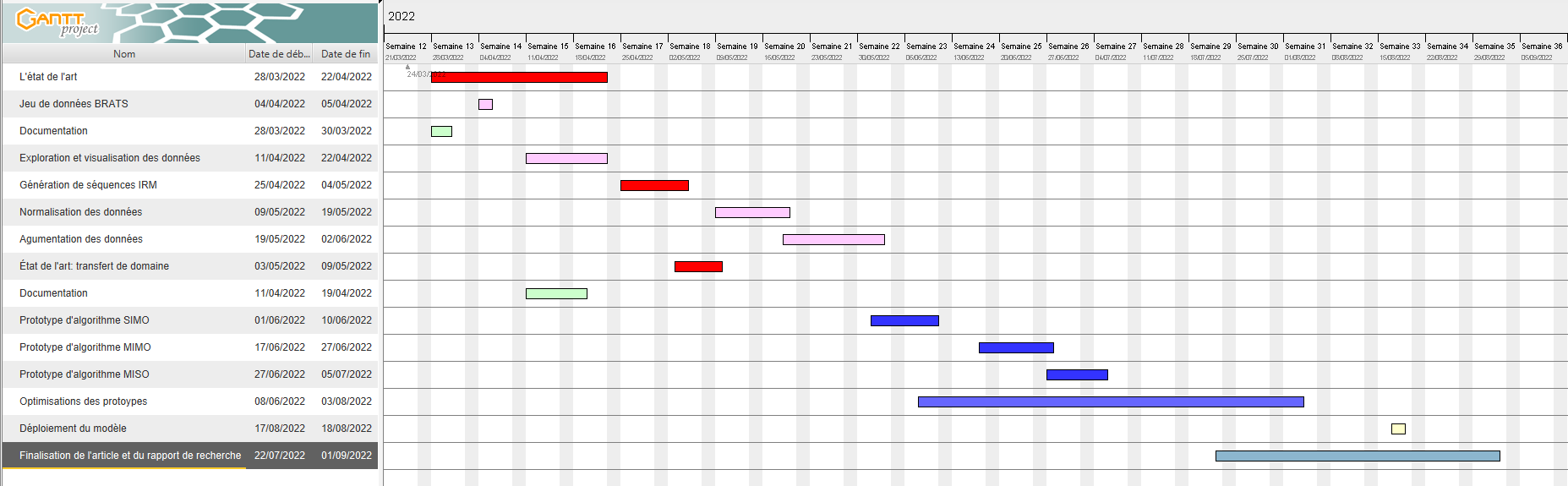
**Linux** et **Windows** systèmes d'exploitation sont des logiciels qui gèrent le matériel informatique, les ressources logicielles et fournissent des services communs aux programmes. Ainsi, il est considéré comme le logiciel fondamental qui pourrait gérer les autres logiciels que nous utilisons, par exemple, anaconda.

### **I.4.2 Programmes de logiciels**

**Service SourceSup**, exploité par Renater, est une plateforme web de gestion pour les organismes d'enseignement supérieur et de recherche français. Chaque membre du réseau peut créer un projet sur la plateforme en permettant la collaboration supplémentaire de personnes extérieures au projet créé.

**GanttProject** décompose le travail, construit un diagramme de Gantt, affecte les ressources et calcule les coûts du projet. Nous l'avons utilisé pour débriefer les tâches du projet pour les semaines en cours à travers les tâches réalisées et celles encore en cours.

**Figure 3**- Diagramme de GANT - gestion du temps des tâches



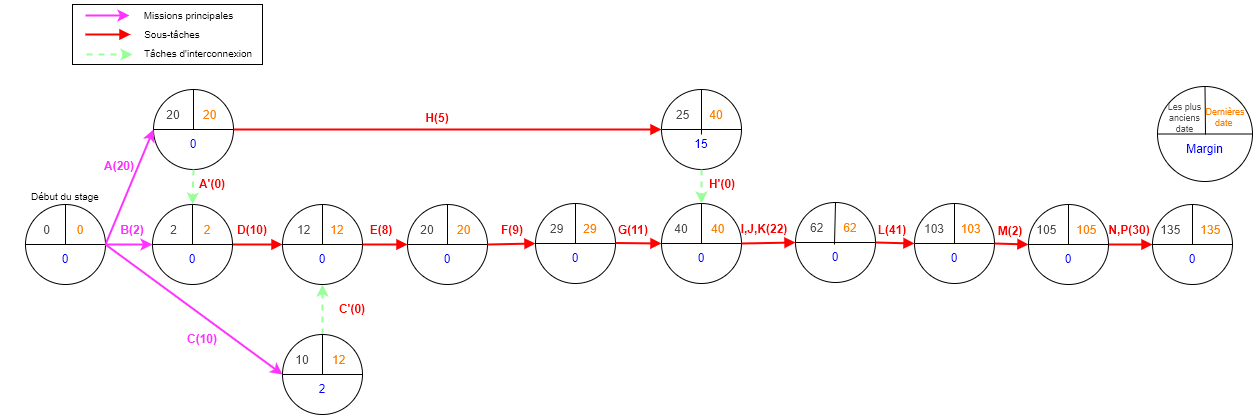
Comme dans la figure () ci-dessus, nous pouvons voir les tâches en cours, terminées et continues sur lesquelles j'ai travaillé. Par exemple, du au , j'ai couvert l'état de l'art concernant les GANs et la synthèse médicale.

**PERT**, un outil de planification, d'organisation et d'ordonnancement des tâches au sein d'un projet, représente le calendrier du projet et catégorise les tâches individuelles. Les diagrammes PERT sont semblables aux diagrammes de Gantt, mais avec une structure différente. Dans le tableau () et la figure () suivants, nous présentons les principales tâches et leurs représentations en fonction de leur priorité et de leur durée au plus tôt et au plus tard.

**Tableau 2**- Diagramme de PERT - gestion des tâches

|  |  |  |  |
| --- | --- | --- | --- |
| **Tâche** | **Nature** | **Durée** | **Antériorité** |
| A | État de l'art |  | - |
| B | Jeu de données BRATS |  | - |
| C | Documentation |  | - |
| D | Visualisation et exploration des données |  | B |
| E | Génération de séquences IRM |  | D, C |
| F | Normalisation des données |  | A, B, C |
| G | Augmentation des données |  | F |
| H | État de l'art : transfert de domaine |  | A |
| I | Prototype SIMO |  | H, G |
| J | Prototype MIMO |  | H, G |
| K | Prototype MISO |  | H, G |
| L | Optimisations des prototypes |  | I, J, K |
| M | Déploiement du modèle |  | I, J, K, L |
| N | Rédaction d'un article de recherche |  | I, J, K |
| P | Rapport final |  | I, J, K |
| **Total** |  |  |

**Figure 4**- Graphique PERT classique - gestion du temps des tâches, représenté en tableau ()



L'élément-clé que nous pouvons remarquer dans le graphique PERT, c'est que nous pouvons estimer la date la plus proche de l'achèvement d'une tâche et l'état de liberté, comme dans la tâche H. Par conséquent, la tâche pourrait être retardée de jours de plus parce que les autres tâches, par exemple, D, E ou F, ne dépendent pas de H.

**draw.io** est un logiciel de diagramme en ligne gratuit. Il est utilisé pour écrire et simplifier des algorithmes, du code et des informations dans un schéma visuel facile à comprendre.

**Webex**, de Cisco, est un logiciel de vidéoconférence, de réunions en ligne, de partage d'écran et de webinaires. Il a été utilisé pour rapporter les résultats, les problèmes et les plaintes au superviseur ainsi que pour planifier les réunions hebdomadaires.

**ENT Zimbra** est un espace collaboratif pour écrire aux collaborateurs de stage, partager des messages, et créer des rendez-vous.

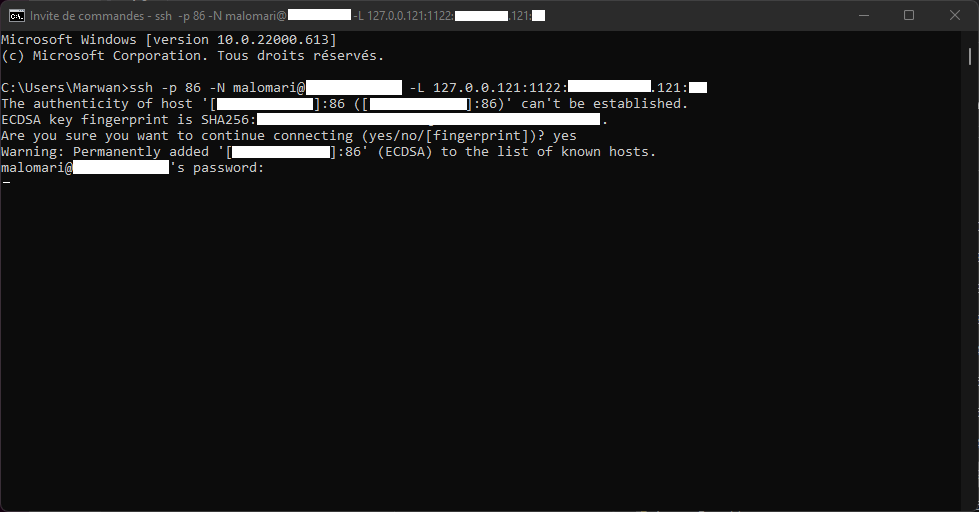
**Microsoft Office** est une application de traitement de texte qui permet de configurer les attributs d'un document, tels que la mise en page, les styles de contenu, et d'ajouter leur contenu de diverses manières et dans divers formats pour produire des documents. Il est utilisé pour rédiger les rapports hebdomadaires ainsi que la version finale du rapport.

**Microsoft PowerPoint** est un programme de présentation, qui utilise des diapositives pour transmettre des informations riches soutenues par différents multimédias.

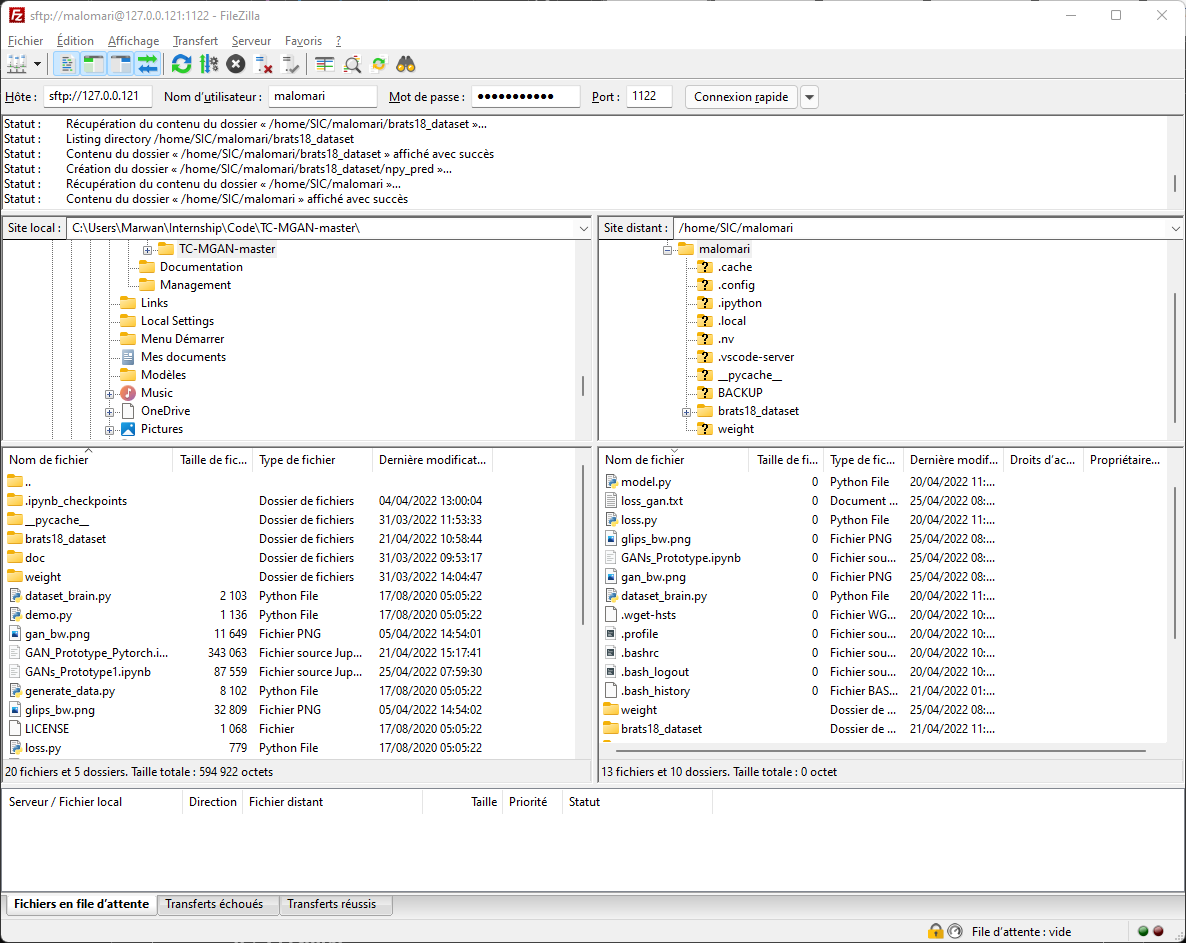
**Zotero[[1]](#footnote-1)** aide à collecter, organiser, citer et partager différentes sources de recherche en ligne.

**FileZilla** est une application FTP gratuite et open-source, multiplate-forme, composée de FileZilla Client et FileZilla Serveur. Nous avons utilisé le client FileZilla, qui prend en charge les connexions aux serveurs FTP, FTPS et SFTP. Un exemple de connexion réseau effectuée au serveur en question est représenté dans les deux figures suivantes () et (). La première figure () établit la connexion avec la machine distante par son IP à l'adresse , port . Dans FileZilla, nous entrons la coordination qui est bien établie dans le terminal pour interroger le serveur en question, comme il est bien précisé dans la figure ().

**Figure 5**- Établissement d'une connexion SSH avec le serveur en question pour la manipulation SFTP



**Figure 6**- Connexion SFTP à l'aide de FileZilla



### **I.4.3 Plateformes de codage**

**Jupyter Notebook** fournit des services de calcul interactifs en utilisant Python. Ainsi, en codant, Jupyter visualisera les résultats dans une même fenêtre de script. Ainsi, il facilite le débogage de l'algorithme ou du code de l'application.

**Anaconda** est une distribution du langage de programmation Python pour le calcul scientifique, notamment la science des données, les applications ML, le traitement des données à grande échelle, l'analyse prédictive, etc. Elle est utilisée dans le but de simplifier la gestion et le déploiement des paquets.

**Microsoft Visual Studio Code**, appelé VS code, est un éditeur de code. Nous l'avons utilisé pour faciliter les connexions SSH et l'intégration du code dans le même espace de travail. En outre, il est plus facile d'utiliser et de manipuler les fichiers du carnet Jupyter sans qu'un serveur local ne s'ouvre en arrière-plan, comme le montre la figure () ci-dessus.

# **1. Introduction**

Le traitement des images fait appel à des méthodologies différentes et multiples, depuis les caractéristiques forgées à la main jusqu'à la DL, pour les objectifs de détection, de classification et de segmentation. En tant que développement conséquent de l'IA, la DL utilise des unités de traitement plus profondes appelées neurones pour apprendre des caractéristiques et des représentations hiérarchiques des données . Dans ce cas, les méthodes de DL surpassent les prédictions humaines, car elles connaissent des progrès et des évolutions impressionnants dans différents domaines scientifiques, par exemple la vision par ordinateur .

Dans le contexte de l'imagerie médicale, la classification et la régénération des images sont généralisées pour distinguer les différentes classes et catégories dans l'expérimentation du réglage des paramètres dans les tâches auto-génératives . Les réseaux génératifs, notamment le transfert de style de collection, la transfiguration d'objets, le transfert de saison, l'amélioration de photos, etc. .

En ce qui concerne l'objectif de ce stage, nous avons donc adopté le modèle GANs pour générer différents attributs à partir d'un seul et de plusieurs attributs en entrée. Les architectures proposées permettent de gagner du temps dans la prédiction des attributs d'imagerie médicale et d'éviter l'injection de certaines substances aux patients. Par exemple, dans le cas du Tc, le patient est injecté avec du gadolinium pour extraire une modalité avec un meilleur contraste. Les données de BRATS, utilisées dans cette étude, sont largement utilisées pour améliorer le diagnostic et la segmentation des lésions cérébrales. En pratique, BRATS comprend des images de fichiers de patients qui contiennent toutes quatre modalités (attributs) et une carte de segmentation (vérité terrain) : T, Tc, T, et FLAIR.

Les travaux de ce projet sont spécifiés en sections. La première section présente déjà le sujet à l'étude. Plus tard, dans la deuxième section, nous découvrirons les travaux connexes qui sont bien fournis sur la synthèse d'images. La troisième section détaille le contexte DL dans le cadre de la synthèse médicale ainsi que le jeu de données BRATS. La quatrième section précise l'approche de recherche en discutant les étapes de prétraitement, les algorithmes et les mesures d'évaluation. Dans la cinquième section, nous présentons les résultats des expériences avec une analyse spécifique. La sixième section conclut l'idée générale du projet de recherche en soulignant les défis à relever et les améliorations à apporter à l'avenir. Enfin, la liste des références est numérotée tout au long de l'article pour faciliter la navigation.

# **2. Travaux relatifs**

Dans cette section, nous présentons les principaux articles de recherche qui ont été expérimentés sur des jeux de données open-source provenant de différents défis, par exemple CelebA, RaFD ou BRATS . L'objectif de la couverture des travaux connexes sur l'utilisation de l'architecture des GANs est de mieux comprendre la littérature et les techniques employées dans les approches des chercheurs par rapport à notre étude de projet. A la fin de cette section, nous fournissons un résumé des études dans le tableau tout au-dessous.

Leur réseau multimodal a été évalué sur quatre contrastes de T, Tc, T et FLAIR en calculant NMAE, PSNR, SSIM, VIF, NIQE . Ils ont adopté la stratégie StarGAN pour cartographier les quatre modalités d'IRM à travers modèles SISO (une modalité à la fois), et un GAN unifié pour les cartographier ensemble. Leur méthode a été évaluée sur BRATS, qui comprend sujets, dont patients atteints de LGG et patients atteints de HGG. La taille originale de chaque image de l'ensemble de données est de . Pour le prétraitement, chaque image est réduite à une taille de patch de . Ensuite, l'estimation finale des régions superposées a été fixée à avec une validation croisée fois. La plupart des paramètres importants de leur modèle sont fixés à , , , et pour équilibrer les poids de la classification, de la cohérence synthétique, de la CC et des pertes adverses, respectivement. En conclusion, par exemple, avec T comme modalité d'entrée, les NMAEs pour les Tc, T, FLAIR générés sont , , et , les PSNRs sont dB, dB, et dB, les SSIM sont de , , et , les VIF sont de , , et , et les NIQE sont de , , respectivement. Enfin, leur étude est limitée aux images multimodales spatialement co-enregistrées avant même qu'elles ne soient utilisées pour la formation et le test sur un petit ensemble de données. Pour les travaux futurs, ils essaieront d'augmenter la quantité d'images d'entraînement et de test par des techniques d'augmentation.

StarGAN , un GAN unifié, gère les traductions multi-domaines d'image à image en utilisant un modèle unique de discriminateur et de générateur un-à-un. Un modèle unique prend en entrée une image et des informations sur le domaine par codage à un coup. Il apprend les correspondances entre tous les domaines disponibles, en utilisant un seul générateur. Le processus d'apprentissage génère une étiquette de domaine cible aléatoire afin d'assurer une traduction parfaite de l'image. En raison des limites des modèles uniques qui ne pouvaient pas être généralisés à plus d'une tâche individuelle, le chercheur a proposé une nouvelle approche qui pourrait gérer plusieurs domaines en utilisant un seul générateur et un discriminateur. Leur approche fonctionne bien sur les tâches de transfert de caractéristiques faciales et de synthèse d'expressions faciales. Leur architecture a été testée sur deux jeux de données. Tout d'abord, CelebA contient étiquettes liées à des caractéristiques faciales telles que la couleur des cheveux, le sexe et l'âge. Deuxièmement, le jeu de données RaFD contient étiquettes pour les expressions faciales telles que la joie, la colère et la tristesse. En outre, pour éviter les valeurs manquantes dans le jeu de données, ils ont appliqué un vecteur de masque de l'étiquette du domaine qui permet d'ignorer les étiquettes inconnues et de se concentrer sur ce qui est déjà disponible. Sur CelebA, StarGAN a obtenu , , , , , , pour la couleur des Cheveux, le Sexe, l'Âge, CS, CA, SA, et CSA, respectivement. Sur RaFD : StarGAN a atteint pertes avec paramètres. Sur CelebA et RaFD, le modèle a correctement appris le rôle prévu d'un vecteur de masque dans les traductions d'image à image lorsque toutes les étiquettes proviennent de plusieurs ensembles de données. En conclusion, StarGAN a généré des images de meilleure qualité visuelle que les méthodes existantes.

Pix2pix , un générateur de U-Net et un discriminateur de CGANs, combine une perte contradictoire avec une perte L pour capturer les basses fréquences. Progressivement, leur modèle sous-échantillonne les images d'entrée à partir d'une grille haute résolution vers une couche de connecteurs de saut (concaténation des canaux des couches précédentes) suivie de U-Net. Leur algorithme s'est avéré efficace, tout comme PatchGAN (qui exécute des patches pouvant capturer les hautes fréquences de la structure de l'image). Pour l'optimisation de l'algorithme, ils ont utilisé des minibatchs SGD et ADAM, avec un TA de , et des paramètres de momentum , . Ils ont également utilisé diverses normalisations de lots, allant de à . Ils ont entraîné leurs CGAN sur différentes tâches et ensembles de données, y compris des Cityscapes, des façades GMP, des photos Google Maps, des photos en couleur, des bords de photos, des esquisses dessinées par l'homme, des images de jour et de nuit, des images thermiques et en couleur, et enfin des photos à pixels manquants et des photos non peintes. Ils ont déclaré que même avec une petite taille d'entraînement, ils ont été en mesure d'obtenir un résultat de descente sur un seul GPU Pascal Titan X. Comme une limitation pour mesurer les pertes de structure en appliquant des mesures traditionnelles comme le MSE par pixel, ils ont appliqué une méthode conjointe. Premièrement, un test de génération de cartes, de photos aériennes et de colorisation d'images a été effectué pour résoudre des problèmes de graphes sur l'AMT. Ensuite, un système métrique a été utilisé pour reconnaître la correspondance d'objets réalistes dans les images. En conclusion, la perte de L1, le CGAN et L1CGAN ont atteint sur les Cityscapes , et d'ACC par pixel, et , et d'ACC par classe, respectivement.

UNIT combines les VAE avec CoCAN, où deux générateurs partagent des poids pour apprendre la distribution des images dans un domaine croisé. Le partage de l'espace latent implique le mappage CC entre les domaines source et cible, de manière interchangeable. Ils ont utilisé la contrainte de partage des poids pour relier les poids des dernières couches des VAE, extraites des hautes fréquences codées, pour le décodage. Ils ont utilisé ADAM avec TA et à momentum. Le batch size d'une image provenant de différents domaines. Leur cadre a différents paramètres de , et . Les codeurs étaient constitués de couches convolutionnelles et de blocs résiduels de base. D'autre part, les générateurs étaient constitués de blocs résiduels de base comme frontal et de couches convolutives transposées comme back-end. D'autre part, les discriminateurs étaient constitués de piles de couches convolutionnelles avec LeakyReLU non linéaire. Ils ont utilisé un ensemble de données cartographiques, qui contenait des paires d'images correspondantes dans deux domaines (images satellites et cartes). Pour itérations, L'expérience a fonctionné dans un cadre non supervisé en utilisant images satellites de l'ensemble de formation comme premier domaine et cartes de l'ensemble de validation comme second domaine. Les cartes de vérité du sol correspondantes ont été traitées pixel par pixel, chaque translation de pixel étant considérée comme correcte uniquement si la différence de couleur était inférieure à des valeurs de couleur de la vérité du sol. Comme mesure de performance, ils ont utilisé la moyenne des pixels ACC. L'algorithme a obtenu des précisions de classification : SVHN→MNIST , MNIST→USPS , USPS→MNIST .

DiscoGAN et CycleGAN sauvegardent les attributs d'information entre les images d'entrée et les images traduites en utilisant une perte CC. Cependant, ils souffrent tous d'une généralisation limitée car il s'agit de modèles SISO.

D'une part, DiscoGAN, un GAN multi-domaine sans aucun ensemble de pré-entraînement explicite, prend une image d'entrée dans un domaine et génère celle qui lui correspond. Ils ont proposé des GAN couplés en forçant les images générées par une perte de reconstruction à être un double de l'image originale, par exemple, une image de sac à main. Dans les expériences, toutes les images d'entrée et traduites sont de taille . Certains des paramètres adoptés sont TA, minibatchs, et l'optimiseur ADAM avec et . En outre, la normalisation par lots a été appliquée à toutes les couches de convolution et de déconvolution, à l'exception de la première et de la dernière couche. De plus, le coefficient de décroissance du poids varie entre . Les images de données varient en rotation azimutale de à . Leurs expériences se résument à des ensembles de données de voiture à voiture et de visage à visage. L'ensemble de données sur les voitures est constitué d'images rendues de modèles de voitures en D avec des angles d'azimut variables à intervalles de . Pour l'évaluation, ils ont traduit les images dans l'ensemble de test et leurs angles d'azimut ont été prédits en utilisant un régresseur. Le modèle proposé montre une forte corrélation entre les angles prédits des images d'entrée et des images traduites, ce qui permet de découvrir la relation d'azimut entre deux domaines. En termes de rotations entre un GAN standard et un GAN à perte reconstructive, les images générées ne varient pas autant que les images d'entrée. Lors d'une expérience sur le jeu de données de voitures, les deux modèles ont souffert d'un effondrement soudain. En outre, ils ont expérimenté la conversion de visage par attributs de visage sur les jeux de données CELEBA, facescrub, où une seule caractéristique, comme le sexe ou la couleur des cheveux, varie entre deux domaines et des domaines partagés. Les résultats traduits ont non seulement des couleurs et des motifs similaires, mais ils ont également un niveau de formalité de mode similaire à celui de l'article de mode d'entrée. Les travaux futurs porteront sur la modification de l'algorithme pour le traitement des modalités mixtes (par exemple, texte et image).

D'autre part, CycleGAN, utilise deux pertes CC qui capturent la translation d'image d'un domaine à l'autre et vice versa. L'architecture contient trois convolutions, des blocs résiduels, deux convolutions à stride fractionné avec stride, et une convolution qui fait correspondre la caractéristique à l'espace RVB. Leur réseau contient blocs pour les images et blocs pour les images . Pour le discriminateur, ils ont utilisé des PatchGAN avec normalisation des instances. Ils ont utilisé différentes techniques de LGAN et d'oscillation pour stabiliser leur modèle. Premièrement, LGAN remplace l'objectif de log like-hood négatif par la perte des moindres carrés. Deuxièmement, une mise à jour de l'oscillation du discriminateur utilise un historique des images générées plutôt que celles produites par les derniers générateurs. Certains paramètres importants sont ADAM TA, un batch size de 1, et une décroissance linéaire vers zéro sur époques. Les expériences ont donné des résultats positifs, mais ont échoué dans certaines tâches de traduction impliquant des changements de couleur, de géométrie et de texture, par exemple, un chat en chien ou en cheval, et un cavalier sur un cheval en zèbre. De plus, une faiblesse sémantique a été repérée dans les tâches d'étiquetage de photos, et l'ambiguïté de la traduction a donc formé des limites importantes. En conclusion, le modèle a obtenu des étiquettes réelles de pour la carte vers la photo, et de pour la photo vers la carte. En outre, il a obtenu de précision par pixel, de précision par classe et de reconnaissance de classe moyenne pour les étiquettes de photo sur le jeu de données Cityscapes.

**Tableau 3**- Résumé des articles en ce qui concerne le numéro de l'étude, le jeu de données, l'approche utilisée, les résultats de l'expérience, les défis et les travaux futurs possibles.

| **Étude** | **Ensembles de données** | **Approche** | **Résultats** | **Problèmes et défis** | **Travaux futurs** |
| --- | --- | --- | --- | --- | --- |
| [3] | **Description :**  BRATS a sujets de modalités : T, Tc, T, et FLAIR. La taille de chaque image : .  **Pré-traitement :**   * images * régions   **Ensemble de validation :**   * -fold CV | **Modèles :**  - 12 modèles SISO et un réseau GAN unifié  **Configurations :**  - Paramètres de perte :  - Mesures d’évaluation :   * NMAE * PSNR * SSIM * VIF * NIQE | T en entrée :  - NMAEs :   * Tc * T * FLAIR   - PSNRs :   * Tc * T * FLAIR   - SSIMs :   * Tc * T * FLAIR   - VIF :   * Tc * T * FLAIR   - NIQE :   * Tc * T * FLAIR | - Images multimodales spatialement co-enregistrées  - Petit jeu de données | - Techniques d'augmentation des données |
| [2] | **Description :**  - CelebA contient étiquettes de caractéristiques faciales, par exemple la couleur des cheveux, etc.  - RaFD contient 8 étiquettes d'expressions faciales, par exemple, heureux, etc.  **Pré-traitement :**  - Vectorisation des étiquettes | **Modèles :**  - Un GAN unifié composé d'un discriminateur et d'un générateur multiples un-à-un  **Configurations :**  Non spécifié | - ACC CelebA :   * Cheveux * Sexe * Âgés * C+S * C+A * S+A * C+S+A   - Loss RaFD : | Non spécifié | Non spécifié |
| [13] | **Description:**  - Cityscapes  - Façades GMP  - Google Maps  - BW à la couleur  - Croquis d'Humandrawn  - Jour à nuit  - Thermique vers couleur  - Pixels manquants  - Top inpainted **Pré-traitement:**  - Génération de mappage  - Colorisation d'images  - Système métrique | **Modèles:**  - U-Net  - CGANs  **Configurations :**  - Pertes adverses & L1  - Déséchantillonnage  - Minibatch  - TA SGD & ADAM  - & Momentum  - batch size | Perte de L, CGAN, & L1+CGAN ACC sur Cityscapes :  - Par-pixel :  - Par-class : | - Petit jeu de données  - Pertes structurelles des mesures traditionnelles en tant que MSE par pixel | Non spécifié |
| [14] | **Description :**  Le jeu de données cartographiques contient des paires d'images dans deux domaines : les images satellites et les cartes.  **Ensemble d'entraînement :**  images satellites comme premier domaine  **Ensemble de validation :**  cartes en tant que second domaine | **Modèles :**   * VAEs * CoCAN   **Configurations :**  - Partage du poids  - ADAM TA  - momentums  - LeakyReLU  - batch size  - Traduction des pixels  - itérations  - Moyenne de pixels ACC  - Perte de CC  - Paramètres de perte: | Classification ACC :   * SVHN→MNIST: * MNIST→USPS: * USPS→MNIST: | Non spécifié | Non spécifié |
| [15] | **Description :**  - Données sur les voitures  - CELEBA  - Facescrub  **Pré-traitement :**  **-**  images  - Rotation en azimut de à | **Modèles :**   * GANs couplés   **Configurations :**  - TA  - Perte de reconstruction  - minibatch  - Normalisation de batch  - dégradation du poids  -ADAM optimiseur: | - Découverte de la relation azimutale entre domaines | - Limitation de généralisation du modèle SISO  - Effondrement soudain du modèle | - Algorithme modification dans le traitement des modalités mixtes (par exemple, texte et image) |
| [16] | **Description :**  - Cityscapes  - Carte vers Photo  **Pré-traitement :**  - images  - images  - images | **Modèles :**  - PatchGANs  - convolutions  - Blocs résiduels   * blocs * blocs   - convolutions fractionnelles  - stride  - Convolution RVB  **Configurations:**  - Instance  normalization  - Pert de CC  - Perte de least-squares  - ADAM TA  -  - batch size  - Dégradation linéaire sur époques | - Carte vers Photo :  - Photo vers Carte :  - Cityscapes :   * par-pixel ACC * par-class ACC * classe moyenne IOU | - Limitation de généralisation du modèle SISO  - Faiblesse sémantique entraînant une ambiguïté de traduction  - Échec des traductions sur les changements de couleur, de géométrie et de texture | Non spécifié |

En conclusion, la littérature nous a permis de constater que la recherche se concentre sur la méthode SIMO, comme nous l'avons vu dans les articles suivants . De là, nous avons consacré notre travail de recherche à l'introduction de l'architecture MISO, inspirée de . Elle tire profit des performances de pointe, discutées dans la littérature, pour prédire un seul attribut à partir de plusieurs entrées. L'objectif principal de la recherche est de prédire le Tc à partir de plusieurs entrées afin de réduire le temps d'examen IRM et d'éviter l'injection de gadolinium.

# **3. Background**

Dans cette section, nous discuterons du DL dans le contexte du diagnostic médical, en mettant en évidence les architectures GANs ainsi qu'une description détaillée du jeu de données BRATS.

## **3.1 L'apprentissage profond dans le diagnostic médical**

L'IA est prometteuse dans le domaine du diagnostic médical, car elle permet d'éviter les erreurs potentielles et les erreurs médicales préjudiciables. La DL, basée sur les ANN avec apprentissage de représentation par renforcement, méthodes supervisées, semi-supervisées et non supervisées , améliore le diagnostic des symptômes, qui sont difficiles à repérer même par les meilleurs experts. Faisant partie de la grande famille des méthodes ML, les architectures DL telles que les DNN, RNN et CNN ont été appliquées à divers domaines, notamment la vision par ordinateur, la TAL, la MT et l'analyse d'images médicales, où elles ont produit des résultats compétitifs qui surpassent les performances humaines .

En général, les ANN ont été inspirés par le traitement de l'information et les nœuds de communication distribués dans les systèmes biologiques. Les ANN se distinguent des cerveaux biologiques par leur tendance à être statiques et symboliques, alors que le cerveau biologique tend à être dynamique et analogique. Cependant, le noyau principal des ANN fait référence à l'utilisation de couches multiples pour extraire progressivement des caractéristiques de plus haut niveau d'une entrée brute. Par exemple, dans le traitement des images, les couches inférieures peuvent identifier les bords, tandis que les couches supérieures peuvent identifier les concepts pertinents pour les humains, tels que les chiffres, les lettres ou les visages.

De nos jours, la DL reste la méthodologie ML la plus prometteuse et là plus utilisée pour la radiologie et la détection des maladies en général. Ce n'est pas une surprise, car l'imagerie diagnostique prévaut dans le diagnostic clinique et la reconnaissance d'images. En 2016, Geoffrey Hinton, un informaticien et chercheur notable, a prédit que les radiologues et les spécialistes qui diagnostiquent les maladies à partir de l'imagerie médicale comme les rayons X, les tomodensitogrammes et les IRM ; ils allaient bientôt perdre leur emploi. "Les gens devraient arrêter de former des radiologues dès maintenant", va-t-il annoncé, "Il est évident que d'ici cinq ans, l'apprentissage profond va faire mieux que les humains" [8].

Les progrès de la DL la rendent très utile, mais pas assez fiable pour que les machines puissent remplacer les experts humains, comme l'a indiqué Hinton. Cependant, la DL est toujours utilisée pour aider les médecins à présélectionner et à classer les cas par ordre de priorité, mais pas en tant qu'outil principal pour diagnostiquer les patients. A partir de là, il existe diverses utilisations de DL en radiologie et dans d'autres pratiques de diagnostic :

* Détection des anomalies neurologiques.
* Dépistage des cancers courants.
* Identification des infections des reins et du foie.
* Détection des tumeurs cérébrales à haut débit.
* Analyse de l'imagerie dentaire.
* Détection des fractures osseuses et des lésions musculo-squelettiques.

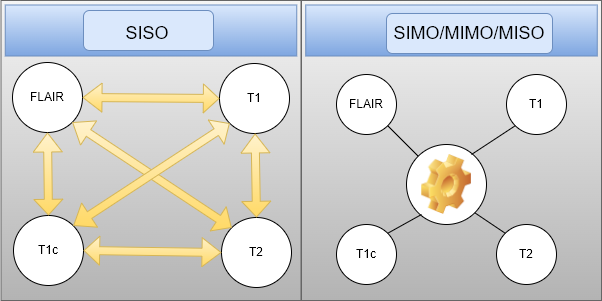
La synthèse d'images médicales, qui fait partie des pratiques DL, est une alternative aux séquences d'impulsions multiples pour l'acquisition d'IRM à contrats multiples [3]. L'IRM est un processus non standardisé dans différentes institutions, la synthèse d'image cross-modale a été proposée pour relever ce défi en fournissant les modalités manquantes. Elle est utilisée dans la pratique clinique en raison de sa capacité à fournir des informations utiles, par exemple les quatre contrats :

1. T1 distingue les matières blanches et grises.
2. T1c évalue le changement de forme de la tumeur avec une démarcation améliorée autour de la tumeur.
3. T2 montre la présence de liquide dans le tissu cortical.
4. FLAIR montre les contours de la lésion.

Dans la recherche et les pratiques médicales, différentes architectures peuvent être envisagées lors de l'application de la DL. Elles sont résumées dans la liste suivante et la figure (7) :

1. **SISO**: une image cible générée à partir d'une image source donnée.
2. **MISO**: surmonter les limites de SISO lorsque les images source et cible sont faiblement corrélées grâce à l'apprentissage de représentations latentes partagées.
3. **MIMO**: synthétiser une ou plusieurs modalités à partir d'une entrée de modalités d'IRM.
4. **SIMO**: où une seule modalité est disponible en entrée, mais où plusieurs contrastes sont nécessaires en sortie.

**Figure 7**- Méthodes SISO, SIMO, MISO et MIMO dans le traitement des modalités IRM

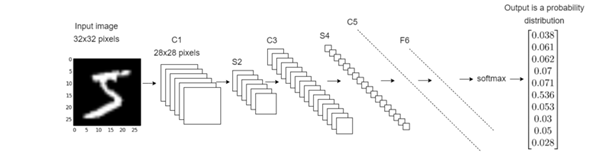


En observation, l'IRM de différentes modalités pourrait fournir des informations complémentaires pour le diagnostic médical, mais il est également difficile et très coûteux d'accéder à toutes les modalités. De nombreuses méthodes se concentrent sur une seule modalité pour en faire une synthèse, ce qui limite considérablement la généralisation des résultats à d'autres modalités. Par conséquent, pour chacune des deux modalités, nous devons développer un modèle distinct afin de pouvoir les mettre en correspondance. Pour résoudre ce problème difficile, nous proposons un GAN à modalités multiples pour synthétiser trois modalités RM (FLAIR, T1 et T1c) à partir d'une modalité RM T2. Un autre défi consiste à prédire FLAIR ou T1c, en évitant d'injecter une substance (par exemple du gadolinium) dans le cerveau du patient, à partir de différentes entrées : T1 et T2 ou T1, T2, FLAIR. Les architectures proposées seront détaillées ultérieurement dans [**4.2 Algorithm**](#_4.2_Algorithm).

## **3.2 CNNs**

Comme les architectures U-Net et GAN sont toutes deux basées sur des CNN, l'architecture est détaillée dans la présente section en prenant comme exemple l'architecture LeNet-5, publiée par Yann LeCun en 1998 [22]. Elle est bien présentée dans la figure (8) suivante :

**Figure 8**- LeNet-5, un exemple d'architecture CNN, extrait de [21]



Les CNN ou ConvNets sont couramment utilisés dans les tâches de classification, de détection et de régression de la vision par ordinateur. En général, les CNNs prennent une entrée sous la forme d'un tableau ou d'une matrice, par exemple une image. Dans ce cas, il s'agirait d'un tableau de pixels et cela dépendrait de sa résolution. L'entrée proposée se présente sous la forme d'une hauteur (h) × largeur (l) × dimension (d). Le d pourrait se référer à RGB par l'indicateur 3 et 1 pour les images en niveaux de gris. Pour procéder à la formation et au test du modèle CNN, chaque image d'entrée passe par une série de couches de convolution avec des filtres (noyaux) spécifiques, des fonctions de mise en commun, de FCL et de perte pour obtenir finalement une classification probabiliste. Pour une explication claire des CNNs, LeNet-5 sera utilisé pour démontrer les différentes couches de CNNs ainsi que les architectures U-Net et GAN.

* Convolution (C1, C3, and C5)
  1. Up-Convolution
  2. Convolution transposée
* Couche de Pooling (S2 and S4)
* FCL (F6)
* Fonctions d'activation

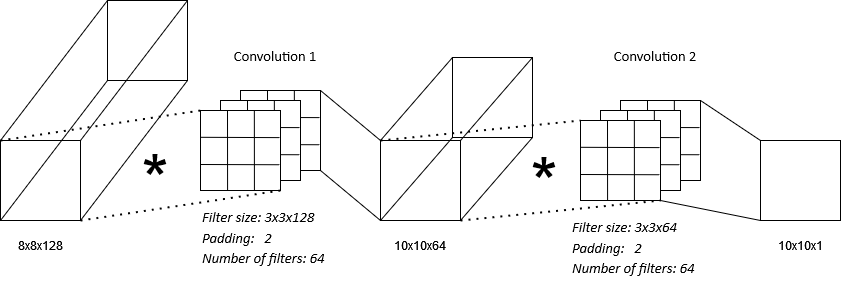
### 

### **3.2.1 Convolution (C1, C3, and C5)**

La première couche qui extrait les caractéristiques de l'image d'entrée en préservant la relation entre les pixels en apprenant les caractéristiques de l'image à l'aide de petits carrés de données. Elle utilise une opération mathématique de deux entrées comme une matrice d'image et un noyau. De tels filtres, par exemple, la détection des bords, le flou et la netteté. Le filtre utilise le stride, qui est le nombre de pixels déplacés sur les données d'entrée. Ainsi, si le stride est fixé à 4, le filtre déplacera 4 pixels à la fois et ainsi de suite.

* Par exemple, la convolution pour augmenter la largeur et la hauteur et réduire la profondeur de l'entrée, par exemple l'image, appliquerait deux couches convolutionnelles consécutives, comme le montre bien la figure (9) :
  1. Première convolution : 3×3×128 filtre, 2 padding, 64 nombre de filtres. Résultats : 10×10×64.
  2. Deuxième convolution : 3×3×64 filtre, 2 padding, 64 nombre de filtres. Résultats : 10×10×1.

**Figure 9**- Une déconstruction de l'opération convolutive dans les CNNs



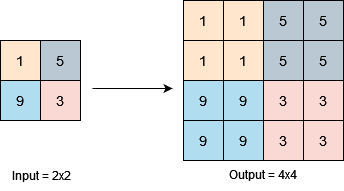
#### **3.2.1.1 Convolution transposée**

D'autre part, la convolution transposée, connue sous le nom de déconvolution, est l'opération inverse de la convolution. Elle permet de compresser la dimension des caractéristiques et d'agrandir leur taille.

#### **3.2.1.2 Up-Convolution**

L'up-convolution ou sur-échantillonnage augmente la largeur et la hauteur de l'image d'entrée en dupliquant les valeurs des pixels pour chaque couche sans poids supplémentaires ni autres opérations complexes. Un exemple d'up-convolution est représenté dans la figure suivante (10), où il y a une entrée plus petite, et la sortie est sur-échantillonnée en effectuant des duplications.

**Figure 10**- Exemple de suréchantillonnage

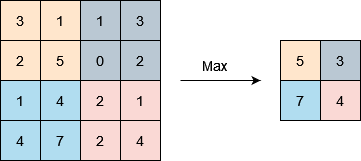


### **3.3.2** **Couche de Pooling (S2 and S4)**

Lorsque l'entrée est importante, le sous-échantillonnage ou le sous-échantillonnage permet de réduire le nombre de paramètres. Il existe différents types de regroupement, comme indiqué ci-après :

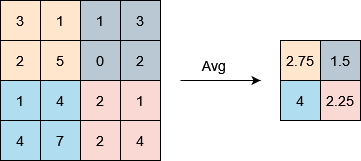
1. **Maximum Pooling** prend le plus grand élément de la carte des caractéristiques.

**Figure 11**- Exemple de max-pooling avec fliter 2×2 et stride 2×2



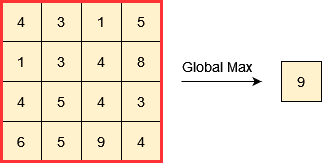
1. **Pooling** **moyenne** prend l'élément moyen de la carte des caractéristiques.

**Figure 12**- Exemple de pooling moyen avec filtre 2×2 et stride 2×2



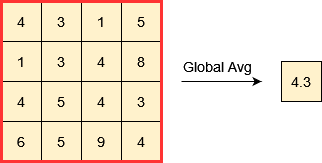
1. **Pooling global** a deux types, qui sont détaillés dans la sous-section suivante :
   1. **Global Max Pooling** prend le plus grand nombre dans une fenêtre d'entrée globale.

**Figure 13**- Exemple de Max-pooling global avec une taille de pool égale à la taille d'entrée



* 1. Global Average Pooling prend la moyenne totale d'une fenêtre d'entrée globale.

**Figure 14**- Exemple de pooling moyen global avec une taille de pool égale à la taille d'entrée



**3.3.3 FCL (F6)**

La couche FCL flatte l'entrée de la couche précédente en un vecteur. La couche FCL alimente le vecteur par les ANN, où plus tard une fonction d'activation, comme par exemple la fonction SoftMax, classera les sorties de l'entrée alimentée, par exemple l'image.

### **3.3.4 Fonctions d'activation**

Elle est appelée fonction de transfert si la plage de sortie est infinie, tandis qu'elle est appelée fonction d'écrasement dans le cas où la plage de sortie de la fonction d'activation est limitée, par exemple, la fonction sigmoïde qui transpose les nombres réels entiers en [0,1]. Désormais, la fonction d'activation calcule la somme pondérée de la couche en la cartographiant dans un vecteur spatial à l'aide de l'une des fonctions : ReLU, LeakyReLU, Sigmoïde, etc.

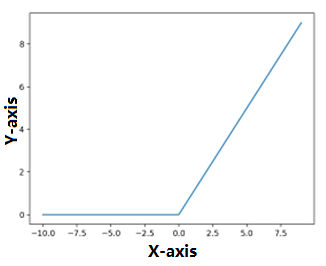
#### **3.3.4.1 ReLU**

La fonction ReLU est calculée comme suit :

(1)

, où X est l'entrée, et l'opération Max prend le maximum de (0,0, entrée X). Par conséquent, si la valeur d'entrée X est négative, la fonction renvoie 0,0. Dans le cas contraire, la fonction renvoie une réponse très positive. Dans la figure suivante (15), la fonction ReLU est bien représentée.

**Figure 15**- Fonction ReLU en représentation graphique



#### **3.3.4.2 LeakyReLU**

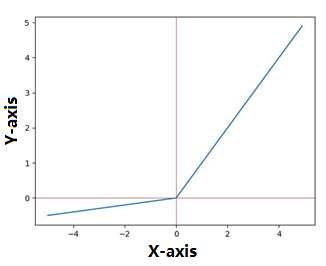
D'autre part, LeakyReLU effectue une valeur seuil chaque fois que l'entrée X est inférieure à 0. Par conséquent, l'entrée est multipliée par alpha. L'opération est résumée comme suit :

, (2)

, (3)

Le LeakyReLU est notamment utilisé pour résoudre le problème des gradients évanescents, car dans le ReLU, l'entrée négative X est toujours fixée à 0. Dans ce cas, un phénomène de ReLU mort est signalé car aucun apprentissage n'a lieu lorsque le nouveau poids reste comme l'ancien. Par conséquent, la dérivée est 0 dans la rétropropagation.

**Figure 16**- Fonction LeakyReLU en représentation graphique



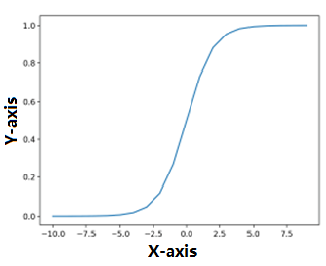
#### **3.3.4.3 Sigmoïde**

L'activation sigmoïde, aussi appelée fonction logistique ou fonction en forme de S, se calcule comme suit :

, (4)

Par conséquent, la fonction prend une entrée de valeur réelle X∈R et sort une valeur dans la gamme de 0 à 1. Avec une entrée plus positive ; plus la sortie sera proche de 1,0, tandis qu'avec une valeur plus négative, plus la sortie sera proche de 0,0.

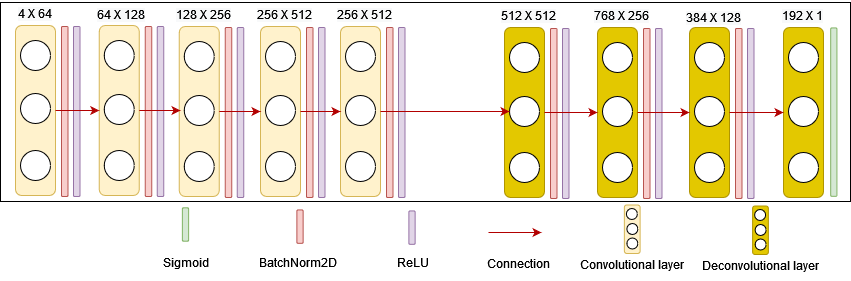
**Figure 17**- Fonction sigmoïde en représentation graphique



## **3.3 U-Net**

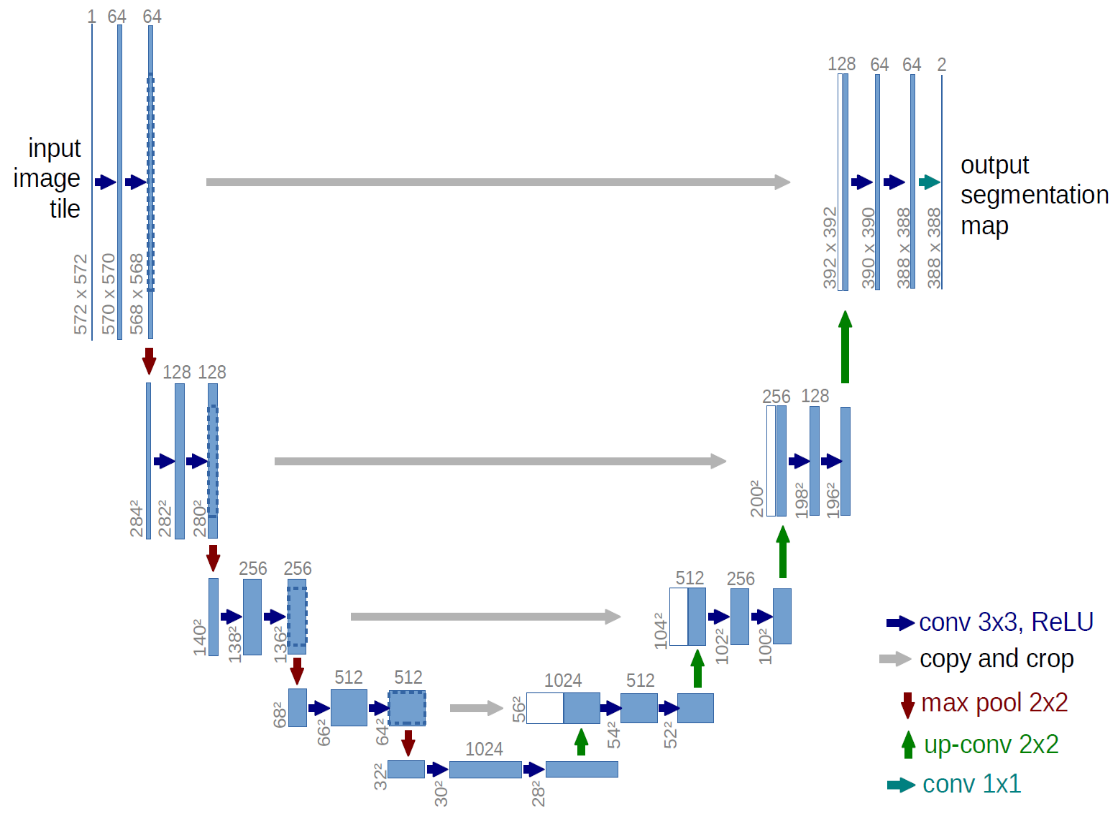
U-Net, une segmentation d'image, permet de réduire le volume de données. L'U-Net est composé de deux canaux. Le premier canal ressemble à l'encodeur car il capture le contexte de l'image. D'autre part, le décodeur est une convolution transposée qui travaille pour construire l'image originale sur la base des segments d'image extraits au cours du processus. Fondamentalement, le réseau assemble des couches de convolution suivies d'une mise en commun maximale qui réduit la densité de l'image (réduit les paramètres d'apprentissage du réseau).

**Figure 18**- Architecture du réseau U-Net



Il a été présenté pour la première fois par [19] en 2015 et, comme son nom l'indique, c'est un réseau en forme de U composé de chemins contractés et étendus. Le réseau a une approche de segmentation d'image qui va au-delà de ses concurrents. L'intuition derrière le U-Net, sur une descente (chemin contracté), il apprend la classification des objets. En montée (chemin étendu), il apprend la position de l'objet. En d'autres termes, la couche correspondante sur le chemin raccourci transmet certaines informations au chemin étendu. De cette façon, le contexte de classification est transféré au module de localisation, ce qui rend l'ensemble du réseau beaucoup plus performant.

**Figure 19**- L'architecture U-Net, qui est extraite de [19][20]



En profondeur, l'architecture se compose de quatre blocs de cryptage et d'un autre de décryptage reliés par un pont de connexion. Le réseau de codeurs (chemin contracté) divise par deux la taille de l'espace et double le nombre de filtres (canaux fonctionnels) à chaque bloc de codage. De même, le réseau de décodeurs double la taille de l'espace et divise par deux le nombre de canaux fonctionnels.

Tout d'abord, le réseau d'encodage agit comme un extracteur de caractéristiques et apprend la représentation abstraite de l'image d'entrée par une séquence de blocs d'encodage, qui consistent tous en deux convolutions 3×3. Ensuite, la couche de convolution est suivie d'une fonction ReLU, qui aide à mieux généraliser les données d'apprentissage et introduit une non-linéarité dans le réseau. La sortie de la ReLU sert de connexion de pontage pour le bloc décodeur correspondant.

Après l'encodeur, un max-pooling jusqu'à 2×2 réduit les dimensions spatiales (hauteur et largeur) de la carte de caractéristiques. De plus, la normalisation par lots réduit les changements de covariance interne et rend le réseau plus stable pendant l'apprentissage.

Les connexions de saut fournissent des informations supplémentaires au décodeur pour générer de meilleures caractéristiques sémantiques. Elles agissent également comme une connexion de dérivation, permettant un flux de gradation indirect vers les couches précédentes sans dégradation. En outre, elles rationalisent le flux de gradients pendant la rétropropagation, ce qui permet d'améliorer le réseau dans le processus d'apprentissage et de représentation.

Le réseau d'encodeurs et de décodeurs complète le flux d'informations. Il consiste en deux convolutions 3×3, où chaque convolution est suivie d'une fonction d'activation ReLU. Enfin, le réseau de décodeurs est utilisé pour dériver la représentation abstraite et générer le masque de segmentation sémantique. Le bloc décodeur commence par une convolution de transposition 2×2. Ensuite, il est concaténé avec la carte caractéristique de connexion correspondante du bloc de données. Ces connexions sautantes fournissent des fonctionnalités des couches précédentes qui sont parfois perdues en raison de la profondeur du réseau. Ensuite, deux convolutions 3×3 sont utilisées, où chaque convolution est suivie d'une fonction d'activation ReLU.

La sortie finale du décodeur subit une convolution 1×1 avec activation sigmoïde. La fonction d'activation sigmoïde fournit un masque de segmentation qui représente la classification pixel par pixel.

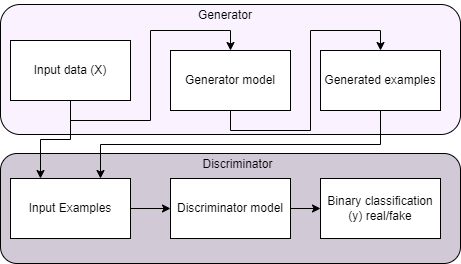
## **3.4 GANs**

En spécifiant un objectif de haut niveau, en produisant des résultats indiscernables de la réalité, puis en apprenant automatiquement une fonction de perte appropriée pour satisfaire cet objectif, c'est exactement ce que font les GANs. Les GANs à apprentissage par perte tentent de classifier si l'image de sortie est réelle ou fausse en formant un modèle généralisé pour minimiser cette perte. Par conséquent, les images floues ne seront pas acceptées car elles auront manifestement l'air fausses. Comme les GAN apprennent une perte adaptative aux données, ils peuvent être appliqués à une multitude de tâches qui nécessitent traditionnellement différents types de fonctions de perte. La perte contradictoire fait le succès clé des GANs, qui forcent essentiellement les images générées à être indiscernables des vraies.

Par définition, les GANs sont une approche de modélisation générative utilisant des méthodes DL comme dans les CNNs. En pratique, les GAN forment un modèle génératif à partir de deux sous-modèles : le modèle générateur, d'une part, apprend à générer de nouveaux exemples en mettant en correspondance les modalités des entrées des réseaux et, d'autre part, alimente le modèle discriminateur pour classer les exemples comme réels ou faux.

Comme nous pouvons le voir dans la figure suivante (20), le modèle de générateur de GAN est considéré comme un problème d'apprentissage non supervisé car il génère un échantillon par lot. En retour, ces entités avec les exemples réels alimentent le modèle GANs. Ensuite, le modèle discriminant, considéré comme un problème d'apprentissage supervisé, est mis à jour en fonction de sa performance de classification de l'échantillon comme réel ou faux. L'ensemble du concept des GANs est représenté dans la figure suivante (20) :

**Figure 20**- Le concept des GANs



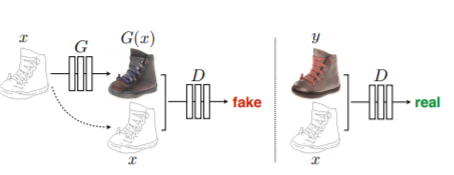
D'autre part, le GAN unifié, composé d'un générateur et d'un discriminateur uniques, tente de mettre en correspondance des images provenant de quatre modalités. En bref, le générateur prend une image avec sa modalité en entrée pour la synthétiser vers une modalité cible. Le discriminateur est capable de différencier les images réelles et synthétisées selon leurs modalités correspondantes.

En termes mathématiques, G:X→Y tel que la distribution des images de G(X) est indiscernable de la distribution Y en utilisant une cartographie de perte adversariale. Le GAN apprend une cartographie à partir de l'image observée X et du bruit aléatoire Z, vers l'étiquette y à travers G : {X,Z}→Y [13]. Le générateur G est entraîné à produire des sorties qui sont indiscernables des "vraies" images par un discriminateur D hautement entraîné. G est entraîné à générer au mieux des images générées par des faux et D fait de son mieux pour distinguer les vraies des fausses jusqu'à ce qu'elles deviennent semblables. Le processus d'apprentissage est fortement sous-contraint ; nous le couplons avec un mapping inverse F∶ Y → X et introduisons une perte CC pour imposer F(G(X))≈X, et vice versa. Pour ne pas perdre de vue qu'il existe deux calculs de pertes CC différents :

1. Perte de CC à avance X → G(X) → F(G(X)) ≈ X
2. Perte de CC à reculons : Y → F(Y) → G(F(Y)) ≈ Y

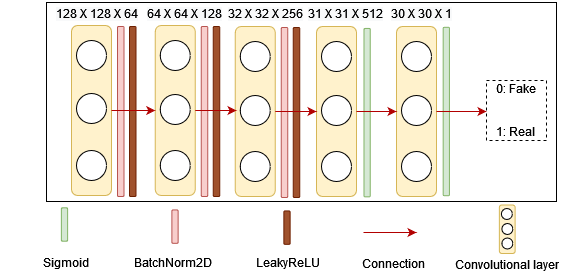
Ainsi, cette procédure de formation est mathématisée dans la figure (21).

**Figure 21**- Démonstration de la cartographie générateur-discriminateur, extraite de [13].



La figure suivante (22) détaille l'architecture GAN en suivant la sortie de l'architecture U-Net, représentée dans la figure (18), sera l'entrée du présent réseau.

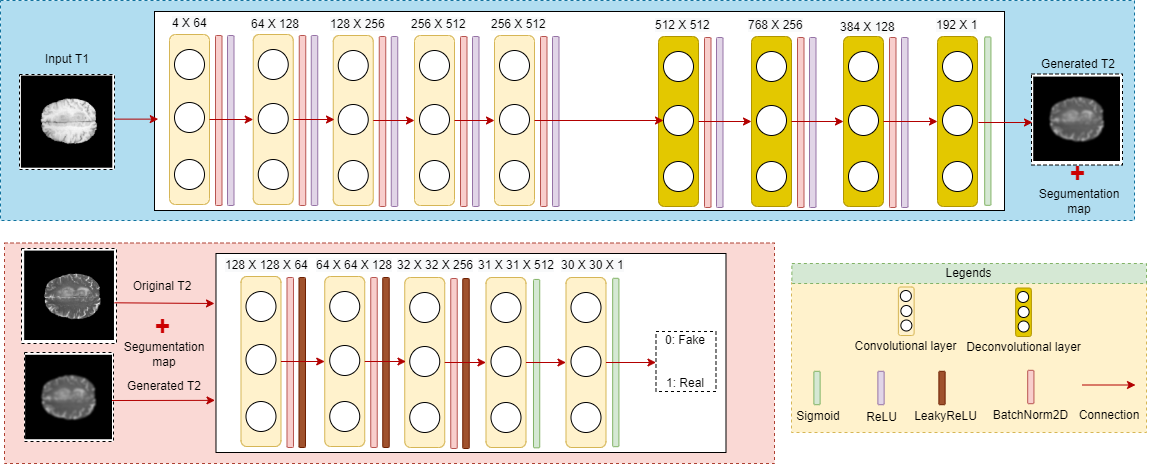
**Figure 22**- L'architecture du GAN



## **3.5 U-Net et GAN**

Après la représentation séparée des deux architectures du générateur U-Net et du discriminateur GAN, la figure suivante (23) représente le réseau DL complet en fournissant un exemple fictif. Le générateur prend en entrée T1 pour générer un faux T2, qui sera l'entrée du GAN avec le vrai T2. À la fin, le GAN s'efforcera de différencier le vrai du faux.

**Figure 23**- L'architecture d'assemblage de U-Net et des GANs



## **3.5 Jeu de données**

BRATS2018 contient des scans IRM 3-T multimodaux acquis en clinique, disponibles sous forme de fichiers NIfTI (.nii.gz), sont des volumes T1, T1c, T2 et FLAIR natifs. Ils ont été obtenus à l'aide de différents protocoles cliniques et de différents scanners provenant de plusieurs institutions [5][6].

Toutes les images ont été segmentées manuellement par 1 à 4 évaluateurs utilisant le même protocole d'annotation. Ces annotations ont été finalement approuvées par un neuroradiologue expérimenté. Les données ont été distribuées après enregistrement simultané en utilisant le même modèle anatomique, interpolation à la même résolution 1mm^3, et prétraitement du décapage du crâne.

La figure suivante (24) représente quatre modalités différentes avec leur carte de segmentation.

**Figure 24** – Echantillon de données illustrant les quatre modalités avec carte de segmentation (vérité terrain)

Une image contenant texte, invertébré, mollusque, matériel

Description générée automatiquement

L'ensemble de données contient 1425 échantillons de modalités, répartis sur 285 patients chacun. Les patients ont été divisés en sous-catégories : HGG et LGG.

BRATS2018

Le tableau (3) suivant résume la distribution de l'ensemble des données.

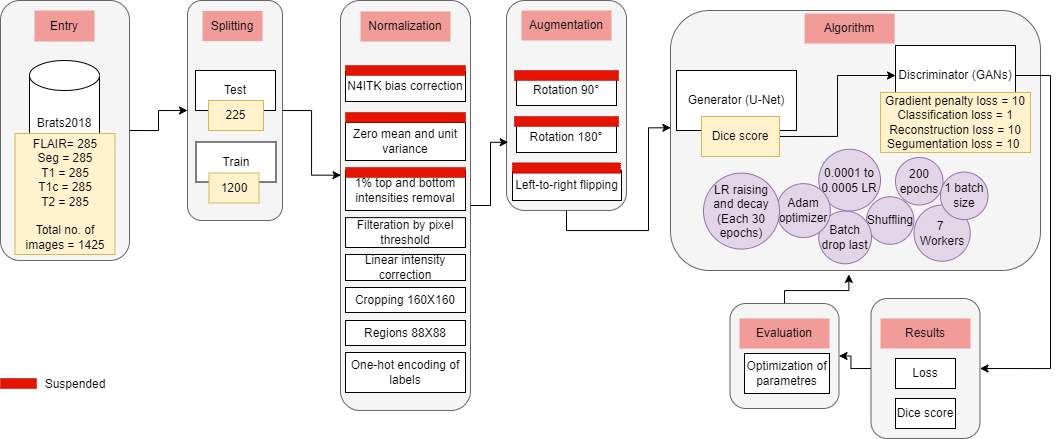
**Tableau 4**- BRATS2018 modality distribution over 285 patients

|  |  |
| --- | --- |
| **Modalité** | **Distribution** |
| T1 | 285 |
| T2 | 285 |
| FLAIR | 285 |
| T1c | 285 |
| Seg | 285 |
| **Total** | 1425 |

# **4. Approache**

Cette section précise les différentes étapes de l'architecture en mettant en lumière l'entrée des données, la division, la normalisation ainsi que les techniques d'augmentation, l'algorithme générateur-discriminateur avec les résultats générés, et enfin l'évaluation et l'optimisation des paramètres. L'ensemble du schéma est présenté dans la figure (25).

**Figure 25**- Architecture du programme, spécifiant les différentes étapes de la saisie des données, du prétraitement, de l'augmentation, de l'algorithme, des résultats et des mesures d'évaluation. Note : le processus marqué en rouge a été suspendu sur les architectures finales



Pour la démonstration, les données sont divisées en 80% de formation et 20% de test. Ensuite, les images sont normalisées en appliquant 5 techniques différentes : correction de biais, normalisation, suppression des intensités, recadrage et enfin codage à un coup. Plus tard dans le processus, les images normalisées sont augmentées par l'application de 3 techniques supplémentaires : retournement de gauche à droite, rotations à 90° et 180°.

L'algorithme consiste donc en un générateur qui prend l'entrée pour générer des modalités fausses que le discriminateur doit classer en deux classes : fausse ou réelle. Plus l'algorithme est incapable de distinguer la différence entre la réalité et l'illusion, plus l'algorithme tend à atteindre la simple perfection.

Le générateur et le discriminateur ont certains paramètres en commun, comme détaillé dans ce qui suit :

1. Optimiseur ADAM avec décroissance de 0,0001-0,0005 TA et relèvement à chaque 25 époques.
2. 7 travailleurs de calcul.
3. L'option "Batch drop last" est activée pour ignorer le dernier lot lorsque le nombre d'exemples n'est pas divisible par le batch size.
4. Mélange des données.
5. 200 époques.

D'autre part, le générateur et le discriminateur ont des paramètres différents. Tout d'abord, le générateur possède les paramètres uniques suivants :

* batch size.
* score de Dice

Le discriminateur distingue certains paramètres, différents du générateur U-Net, détaillés dans ce qui suit :

* batch size.
* Pénalité de perte de gradient.
* Perte de classification.
* Perte de reconstruction.
* Perte de segmentation.

Après le processus de formation, les résultats sont évalués en considérant la perte ainsi que le score de dés. A la fin, nous avons adopté une optimisation des paramètres basée sur la performance de chaque processus de formation. Ainsi, une amélioration globale est manifestement dégagée dans le processus d'optimisation.

## **4.1 Pré-traitement**

Cette section comprend différentes étapes, notamment l'encodage des étiquettes en un coup, la suppression de 1 % des intensités supérieure et inférieure, la moyenne nulle et la variance unitaire, la correction du biais N4ITK et le recadrage 128×128×155 ainsi que 80×80×155. Un résultat du prétraitement des données est représenté dans la figure suivante (26).

**Figure 26** – Le résultat de l'étape de prétraitement, y compris la correction des biais, la normalisation, le filtrage et le recadrage.

Une image contenant invertébré, mollusque

Description générée automatiquement

Malheureusement, la normalisation et la correction des biais sont suspendues dans les prototypes finaux car elles ont généré des échecs dans le processus d'apprentissage. Cette partie est suspecte pour une investigation future. En outre, l'augmentation des données est également suspendue dans le prototype final car le modèle de base a pris la plupart du temps dans l'expérimentation, et il serait intéressant de comparer les résultats de base aux résultats du modèle augmenté à l'avenir.

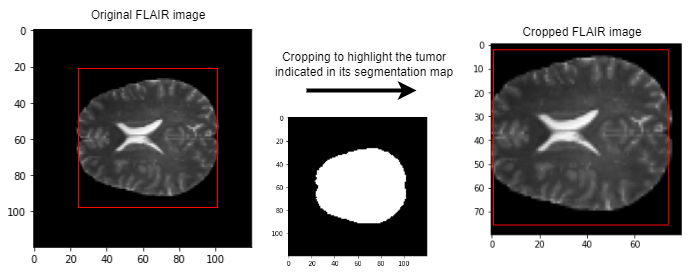
### **4.1.1 Recadrage des images**

L'image d'entrée est de 240×240×155 recadrée à 128×128 pour les 3 prototypes, comme indiqué ci-dessous :

1. SIMO (T1 → T2, FLAIR et T1c).
2. MISO (T1 et T2 → FLAIR).
3. MISO (T1, T2, et FLAIR → T1c).

En outre, un recadrage de 80×80 est adopté pour mettre en évidence la zone tumorale dans les modalités à l'aide de la carte de segmentation. Le recadrage est appliqué au troisième prototype de MISO (T1, T2, et FLAIR → T1c), car le but du projet de recherche est d'éviter l'insertion de gadolinium.

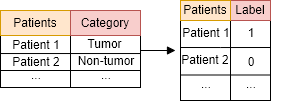
**Figure 27**- Recadrage de l'image de 128×128 à 80×80, mettant en évidence la zone de la tumeur.



### **4.1.2 One-hot encoding**

Les étiquettes des images sont transférées à l'encodage à un coup, qui consiste essentiellement à mettre 1 chaque fois qu'il y a une tumeur, sinon, elle est notée avec la valeur zéro. La représentation de l'encodage est représentée dans le Tableau (4) suivant :

**Tableau 5**- One-hot encoding des étiquettes de tumeur et de non-tumeur.



### **4.1.3 Normalization**

Dans l'étape de normalisation, l'algorithme supprime les 1 % d'intensités supérieures et inférieures, puis normalise l'image avec une moyenne nulle et une variance unitaire. Par conséquent, la normalisation des caractéristiques garantit que la moyenne des valeurs de chaque caractéristique dans les données est égale à zéro (si la moyenne est soustraite du numérateur) et a une variance unitaire [18]. Cette méthode est souvent utilisée pour déterminer la moyenne de distribution et l'écart-type de chaque caractéristique. Ensuite, soustrayez la moyenne de chaque caractéristique. Divisez ensuite la valeur de chaque caractéristique (la moyenne a déjà été soustraite) par son écart-type.

, (5)

où x est le vecteur de caractéristiques original, x ̅=moyenne(x) est la moyenne du vecteur de caractéristiques, et σ est son écart-type.

En outre, les régions hors arrière-plan doivent être normalisées puisque les intensités vont de 0 à 5000. Le minimum dans la tranche normalisée correspond à l'intensité 0 dans la tranche non normalisée, qui a été remplacée par -9 pour garder la trace des intensités 0.

### **4.1.4 Filtration**

Les tranches d'images 2D sont filtrées en fixant un seuil où le nombre de pixels dans la zone du cerveau est inférieur à 2000.

### **4.1.5 Correction de l'intensité linéaire**

Les tranches d'images ont été mises à l'échelle de façon linéaire de leur intensité originale à [-1,+1].

### **4.1.6 Correction du biais**

L'algorithme de correction de champ biaisé N4 corrige les inhomogénéités d'intensité à basse fréquence présentes dans les données d'image IRM, connues sous le nom de champs biaisés ou renforcés. Cette méthode a été rapportée comme ayant été appliquée avec succès comme correction de champ plat pour les données de microscopie. En outre, elle part d'un modèle paramétrique simple et ne nécessite pas de classification des tissus [17].

### **4.1.7 Augmentation des données**

L'augmentation des données est adoptée car l'ensemble de données est considéré comme petit pour la mise en œuvre de DL et en particulier de réseaux génératifs. Par conséquent, nous avons adopté différentes techniques d'augmentation en utilisant des fonctions définies et l'aide de TensorFlow en appliquant des rotations de 90° et 180° et un retournement de gauche à droite de l'image d'entrée originale. Par exemple, la figure (11) suivante représente un exemple d'augmentation de données appliquée à la tranche "Brats18\_TCIA01\_390\_1\_78".

**Figure 28**- Augmentation des données, y compris : des rotations à 90° et 180° et des retournements de gauche à droite.

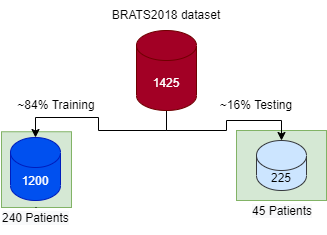
Une image contenant texte, mollusque, chaîne, matériel

Description générée automatiquement

### **4.1.8 Répartition des données**

Les données d'entraînement sont divisées en deux ensembles, l'un d'entraînement et l'autre de test. Pour les 3 prototypes indiqués précédemment dans la section [**4.1.1 Image Cropping**](#_4.1.1_Image_Cropping), Tout d'abord, l'ensemble d'entraînement représente ~84% de l'ensemble de données d'entraînement et l'ensemble de test représente ~16%. Par conséquent, l'ensemble d'apprentissage contient 1200 images pour chaque générateur et discriminateur et l'ensemble de test comprend le reste des images, soit 225 images au total. Le processus de fractionnement est détaillé et bien documenté dans la figure suivante (12) :

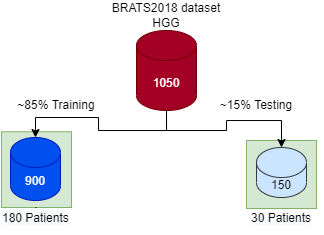
**Figure 29**- Données divisées en ensembles de test et de formation pour les prototypes SIMO (T1 → T2, FLAIR et T1c), MISO (T1 et T2 → FLAIR) et MISO (T1, T2 et FLAIR → T1c) avec une taille d'entrée d'image de 128×128.



Comme nous avons pu le voir sur la Figure (29), l'ensemble de test contient 255 échantillons divisés par 5 modalités : T1, T2, T1c, FLAIR et Seg. D'autre part, 1200 échantillons d'entraînement sont dédiés à l'entraînement de U-Net et de GANs.

D'autre part, pour le quatrième prototype de recadrage des régions 80×80, le jeu de données est divisé en 150 ensembles de test et 900 ensembles d'entraînement sur la seule sous-catégorie HGG BRATS2018. La catégorie HGG a été sélectionnée car elle ne contient que des tumeurs de type 4.

**Figure 30**- Données divisées en ensembles de test et de formation pour le prototype MISO (T1, T2 et FLAIR → T1c) avec un recadrage de 80×80 à l'entrée de l'image.



## **4.2 Algorithme**

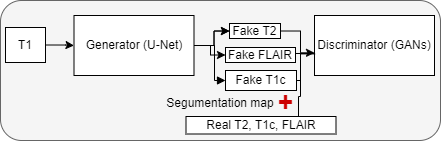
Avec l'existence de nombreux modèles CNN profonds qui obtiennent des résultats compétitifs en termes d'état de l'art, j'ai choisi l'U-Net et le GANs pour affiner le jeu de données que nous avons entre les mains pour son efficacité et sa simplicité. L'U-Net, comme décrit précédemment, gère la cartographie de la segmentation tandis que l'algorithme GAN joue le rôle de discriminer les fausses images générées plus tard dans le processus de formation.

Dans la section suivante, nous présenterons les différentes méthodes algorithmiques que nous avons développées pour combler les lacunes de la recherche dans la littérature par 3 méthodes différentes, mais nous limiterons l'expérimentation à la fin aux méthodes SIMO et MISO car nous visons à prédire FLAIR et T1c.

### **4.2.1 SIMO**

L'architecture SIMO prend une seule entrée, par exemple, dans notre cas T2 pour prédire le reste des modalités en passant par le générateur U-Net et le discriminateur GAN. L'architecture est bien présentée dans la figure suivante (13) :

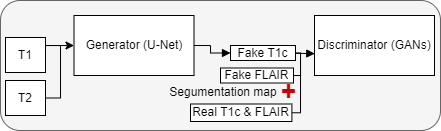
**Figure 31**- L'architecture SIMO prend T1 en entrée, et synthétise T2, FLAIR, T1c, et la carte de segmentation pour la discrimination



### **4.2.2 MIMO**

L'architecture MIMO prend plusieurs entrées, par exemple, dans notre cas, T1 et T2 pour prédire le reste des modalités en passant par le générateur U-Net et le discriminateur GAN. L'architecture est bien présentée dans la figure suivante (14) :

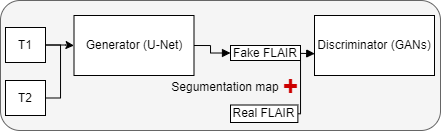
**Figure 32**- L'architecture MIMO prend T1 et T2 en entrée, et synthétise T1c et FLAIR pour la discrimination



### **4.2.3 MISO**

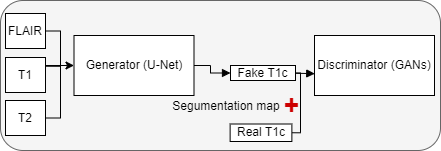
L'architecture MISO prend plusieurs entrées, par exemple, dans notre cas T1 et T2 pour prédire FLAIR ou T1c en passant par le générateur U-Net et le discriminateur GAN. La première architecture de T1 et T2 → FLAIR est bien présentée dans la figure suivante (31) :

**Figure 33**- MISO prend T1 et T2 en entrée, et synthétise FLAIR pour la discrimination



D'autre part, la seconde est l'architecture de T1, T2, et FLAIR → T1c est bien présentée dans la figure suivante (32) :

**Figure 34**- MISO prend T1 et T2, et FLAIR en entrée, et synthétise T1c pour la discrimination



## **4.3 Evaluation Metrics**

Le générateur U-Net, prédisant la carte de segmentation et le sous-échantillonnage, évalué par le score du dé. Le discriminateur GAN est évalué par différents poids de perte, détaillés dans la liste suivante :

* Pénalité de gradient
* Perte de classification
* Perte de reconstruction
* Perte de segmentation

# **5. Expériences et résultats**

L'expérience finale a été lancée sur les spécifications informatiques suivantes du 23 juin 2022 au 31 août 2022 :

**Tableau 6**- Spécifications de l'ordinateur de laboratoire

|  |  |
| --- | --- |
| **Mark** | DELL R |
| **CPUs** | Intel Xeon Silver GHz ,M Cache |
| **GPUs** | Nvidia Tesla VS G |
| **RAM** | GB |
| **SWAP** | GB |
| **Stockage** | TB |
| **Carte de réseau** | GB |

Dans la première phase, le générateur assure de meilleurs résultats car il génère à la fois l'image fausse et sa carte de segmentation. Plus tard, au cours de l'entraînement de l'architecture GAN, les résultats remarqueront une amélioration significative grâce à la perte contradictoire.

**Tableau 7**- Score de dés U-Net des 4 architectures, SIMO a atteint 0,8708 ACC, MISO (sortie FLAIR) a atteint 0,4140, MISO (sortie T1c 128×128) a atteint 0,4147, et MISO (sortie T1c 80×80) a atteint ‼‼

|  |  |
| --- | --- |
| SIMO |  |
| MISO (FLAIR) |  |
| MISO (T1c ) |  |
| MISO (T1c |  |

**Figure 35**- Score de dés U-Net, une comparaison entre SIMO, MISO (sortie FLAIR), MISO (sortie T1c 128×128), et MISO (sortie T1c 80×80),

|  |  |
| --- | --- |
|  |  |
|  |  |

SIMO

**Tableau 2**- Score de dés GANs, une comparaison entre les pertes de discrimination, génération, segmentation

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Tableau 3**- Score de dés GANs, une comparaison entre les pertes de générateur

|  |  |
| --- | --- |
|  |  |
|  |  |

**Tableau 4**- Score de dés GANs, une comparaison entre les pertes de discriminateur

|  |  |
| --- | --- |
|  |  |

**Tableau 5**- Score de dés GANs, une comparaison entre les pertes de discriminateur et de générateur

|  |  |
| --- | --- |
|  | |
|  |  |

MISO F

MISO T 128

MISO T 80

# **6. Conclusion**

Cette section est consacrée à la conclusion de la recherche. Elle comprend les deux sous-sections des conclusions professionnelles et personnelles.

## **6.1 Professionnel**

Grâce à l'art DL, j'ai pu expérimenter différents paramètres d'architecture sur le jeu de données. Grâce à ces expériences, j'ai également constaté que pour ce jeu de données, l'ajustement fin de l'ensemble du modèle donne non seulement de meilleurs résultats mais aide également le modèle à converger beaucoup plus rapidement que l'ajustement fin du reste des couches.

## **6.2 Personnel**

L'opportunité de stage que j'ai eue avec le laboratoire XLIM a été une grande chance d'approfondir mon apprentissage et ma carrière professionnelle dans le traitement de l'image (précisément, sur l'imagerie médicale et la bio-ingénierie), ML, et DL. Ainsi, je me considère comme une personne chanceuse d'avoir le bon poste dans l'ingénierie des données et du ML. Je suis très reconnaissant d'avoir la chance de mettre en pratique de nombreuses ressources de connaissances et de compétences d'indépendance, de recherche et de développement personnel. C'est avec une grande confiance que je peux dire que j'ai la chance de rencontrer différents spécialistes dans le secteur de la santé ainsi qu'en informatique, notamment en DL et ML. En général, ils m'ont conduit bien sûr dans la bonne direction et dans la direction souhaitée pendant la période de stage du 28 mars au 1er septembre de l'année 2022.

## **6.3 Travaux futurs**

Nous pourrions également envisager d'expérimenter différentes fonctions de perte, l'optimisation des hyper-paramètres, des réseaux génératifs plus profonds et l'augmentation des données.

## **6.4 Challenges**

J'ai été confronté à de nombreuses difficultés en termes de configuration de l'environnement Anaconda, qui n'était absolument pas adapté au démarrage de l'algorithme CNNs proposé. De plus, les ressources de calcul limitées sont le défi le plus difficile à relever car je n'ai pas pu mettre à niveau mon ordinateur portable personnel pour améliorer l'ensemble des résultats du projet de recherche. Le tableau suivant (7) spécifie les spécifications de l'ordinateur portable sur lequel nous avons testé les architectures.

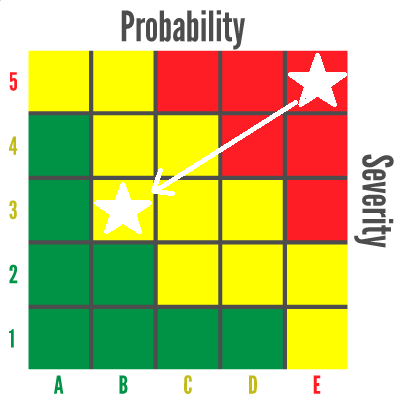
**6.5 Analyse des risques**

Dans cette section, je présenterai les risques rencontrés pendant toute la période du stage.

**6.5.1 L'indisponibilité de l'ensemble des données**

En raison de l'indisponibilité du jeu de données du CHU de Poitiers, nous avons décidé d'expérimenter les architectures de réseaux génératifs sur un jeu de données open-source, appelé BRATS2018. Après avoir pris en compte le plan proposé, nous avons réussi à réduire la gravité du risque d'une gravité et d'une probabilité élevées à une probabilité plus faible avec une gravité moyenne faible.

**Figure 36**- Matrice de probabilité/gravité du risque de non-disponibilité de l'ensemble de données



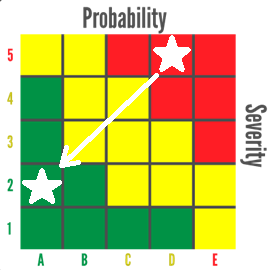
### **6.5.2 Utilisateur administratif non subventionné**

Dans le laboratoire XLIM, le droit d'administration n'est pas donné à un stagiaire, pour une période temporaire. Le problème est que je n'ai pas le droit d'installer les applications par moi-même et que la gestion de l'environnement de programmation me semble impossible. A partir de là, j'ai pris les solutions suivantes pour limiter le risque :

1. Utiliser mon ordinateur personnel pour effectuer le test du prototype.
2. Installer des applications avec des extensions \*.tar pour éviter les droits d'administration.
3. Acquérir deux machines différentes à distance pour finaliser les prototypes finaux.
4. Travailler avec l'environnement virtuel Python car anaconda demande beaucoup de droits d'administration.

Après avoir pris en compte les solutions ci-dessus, le risque a donc considérablement diminué, comme nous pouvons le visualiser dans la figure suivante (21) :

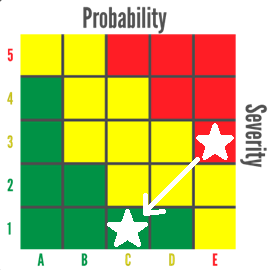
**Figure 37**- Matrice de probabilité/gravité du risque de non-subventionnement de l'utilisateur de l'administration



### **6.5.3 L'ambiguïté du cahier des charges**

Au début du stage, le cahier des charges demandé pour l'ensemble de la période de stage était un peu trop large et ne se limitait pas à un domaine de recherche précis. Plus tard, après une réunion avec le tuteur pédagogique, il m'a demandé d'organiser un planning complet pour l'ensemble du stage. Pour cela, j'ai fait du GanttProject ainsi que du PERT pour occulter les tâches à venir et les tâches en cours. Dès lors, la gravité du problème s'est réduite avec une probabilité moindre.

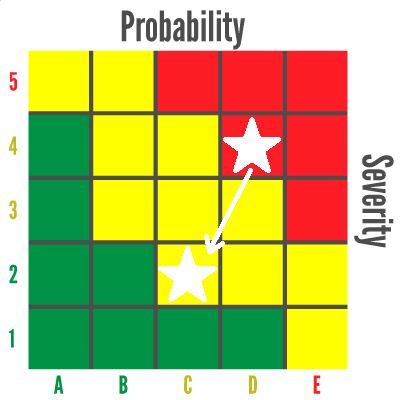
**Figure 38**- Matrice de probabilité/gravité du risque d'ambiguïté des spécifications



### **6.5.3 Entraînement prolongé des architectures DL**

En raison du délai prolongé et du délai d'apprentissage des différentes architectures DL, nous avons proposé de recadrer les tranches d'IRM. Un recadrage de 80×80 est mis en œuvre pour réduire la gravité du problème sur l'architecture MISO, qui prend en entrée : T1, T2, FLAIR, et des sorties : T1c. Il n'est implémenté que sur cette architecture car l'objectif de la recherche est d'éviter l'injection de gadolinium. Le risque est bien diminué comme le montre la figure suivante (38).

**Figure 39**- Matrice de probabilité/sévérité du risque de formation prolongée des modèles



# **7. Références**

1. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. <https://doi.org/10.1007/978-3-319-24574-4_28>
2. Choi, Y., Choi, M., Kim, M.S., Ha, J., Kim, S., & Choo, J. (2018). StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8789-8797.
3. Dai, X., Lei, Y., Fu, Y., Curran, W.J., Liu, T., Mao, H., & Yang, X. (2020). Multimodal MRI Synthesis Using Unified Generative Adversarial Networks. Medical physics. <https://doi.org/10.1002/mp.14539>
4. Menze et al. (2015). The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE transactions on medical imaging, 34(10), 1993–2024. <https://doi.org/10.1109/TMI.2014.2377694>
5. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., Freymann, J.B., Farahani, K., & Davatzikos, C. (2017). Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features. Scientific Data, 4.
6. Bakas et al. (2018). Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge. ArXiv, abs/1811.02629.
7. Xlim Présentation. (n.d.). Retrieved May 10, 2021, from <https://www.xlim.fr/laboratoire/presentation>
8. Standford University. (2017). RSNA 2017: Rads who use AI will replace rads who don't. Retrieved May 3, 2021, from <https://aimi.stanford.edu/news/rsna-2017-rads-who-use-ai-will-replace-rads-who-don-t>
9. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proc. IEEE, 86, 2278-2324.
10. Meskó, B., & Görög, M. (2020). A short guide for medical professionals in the era of artificial intelligence. npj Digit. Med. 3, 126. <https://doi.org/10.1038/s41746-020-00333-z>
11. Krizhevsky, A., Sutskever, I., & G. E. Hinton. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 1097–1105.
12. K. He, X. Zhang, S. Ren, & J. Sun. (2015). Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385.
13. Isola, P., Zhu, J., Zhou, T., & Efros, A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5967-5976.
14. Liu, M.-Y., Breuel, T., & Kautz, J. (2017). Unsupervised image-to-image translation networks. arXiv preprint arXiv:1703.00848.
15. Kim, T., Cha, M., Kim, H., Lee, J.K., & Kim, J. (2017). Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. ICML. https://doi.org/10.48550/arXiv.1703.05192
16. Zhu, J., Park, T., Isola, P., & Efros, A.A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. 2017 IEEE International Conference on Computer Vision (ICCV), 2242-2251.
17. Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., & Gee, J. C. (2010). N4ITK: improved N3 bias correction. IEEE transactions on medical imaging, 29(6), 1310–1320. <https://doi.org/10.1109/TMI.2010.2046908>
18. Grus, Joel (2015). Data Science from Scratch. Sebastopol, CA: O'Reilly. pp. 99, 100. ISBN 978-1-491-90142-7.
19. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. ArXiv, abs/1505.04597.
20. Ronneberger, O. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. University of Freiburg. Retrieved July 20, 2022, from <https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net>
21. Wood, T. (2019). Convolutional Neural Network. Retrieved July 22, 2022, from <https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network>
22. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition.
23. Zwanenburg, J.J.M., Hendrikse, J., Visser, F. et al. (2010). Fluid attenuated inversion recovery (FLAIR) MRI at 7.0 Tesla: comparison with 1.5 and 3.0 Tesla. Eur Radiol 20, 915–922. <https://doi.org/10.1007/s00330-009-1620-2>
24. Laboratoire I3M. I3M. (n.d.). Retrieved July 29, 2022, from <https://i3m.labo.univ-poitiers.fr>
25. Plateforme IRM 7 TESLA. Plateforme IRM 7 tesla. (n.d.). Retrieved July 29, 2022, from <https://www.chu-poitiers.fr/specialites/irm7tesla>

# **8. Annexe**

## **8.1 Ensembles de données :**

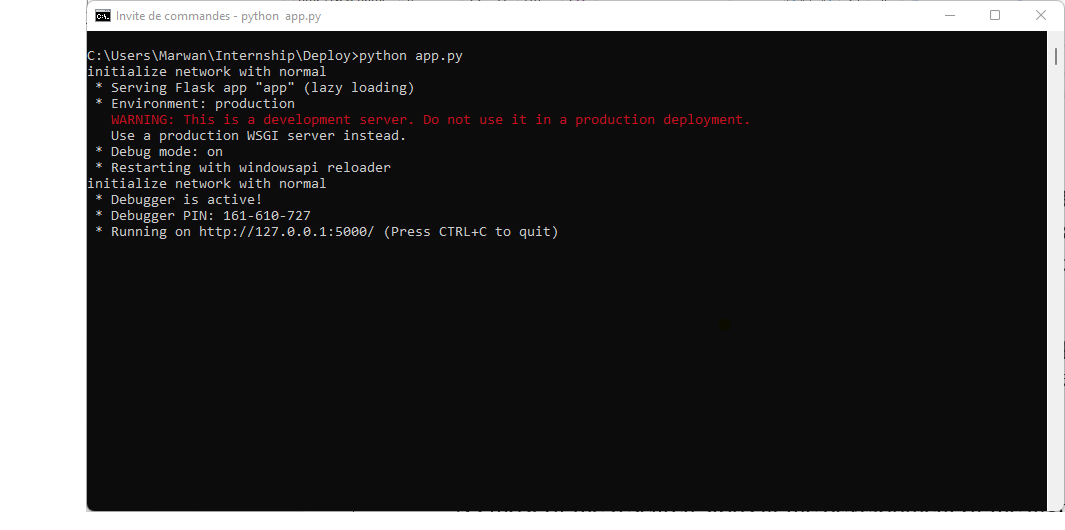
1. Le jeu de données CelebA contient 202 599 images de visages de célébrités, chacune annotée de 40 attributs binaires. Ils recadrent les images, puis les redimensionnent en 128×128. Ils ont sélectionné aléatoirement 2 000 images comme ensemble de test, et ils ont utilisé toutes les images restantes comme données d'entraînement. Ils ont construit sept domaines en utilisant les attributs suivants : couleur des cheveux (noir, blond, brun), sexe (homme/femme) et âge (jeune/vieux).
2. Le jeu de données RaFD se compose de 4 824 images recueillies auprès de 67 participants, qui ont exprimé huit expressions faciales dans trois directions et angles de regard différents. Les images sont recadrées à 256×256, les visages centralisés, et enfin redimensionnées à 128×128.

## **8.2 DLOps**

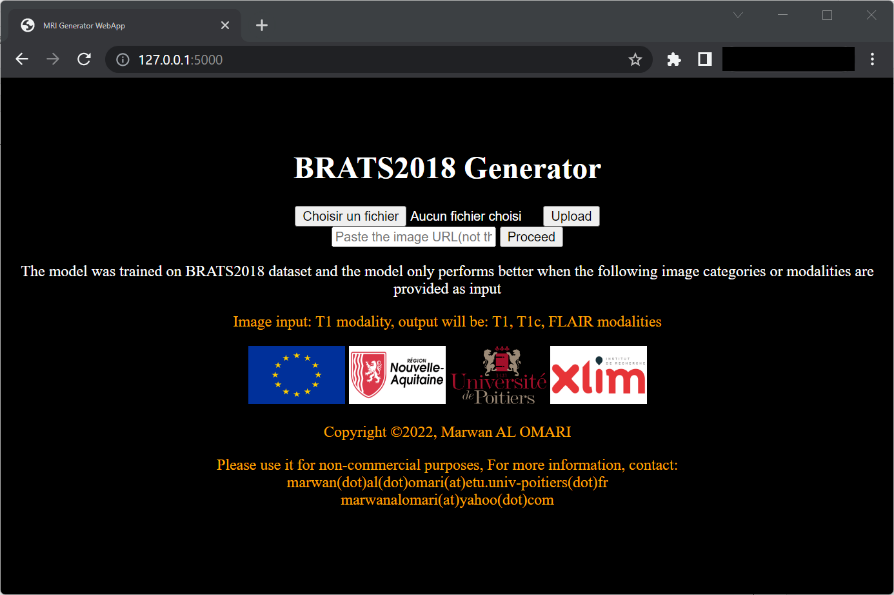
### **8.2.1 Modèle de SIMO**

Comme la plupart des recherches s'arrêtent au développement de l'algorithme, un prototype du modèle SIMO est déployé en tant que microservice, en utilisant la technologie de flask et Python.

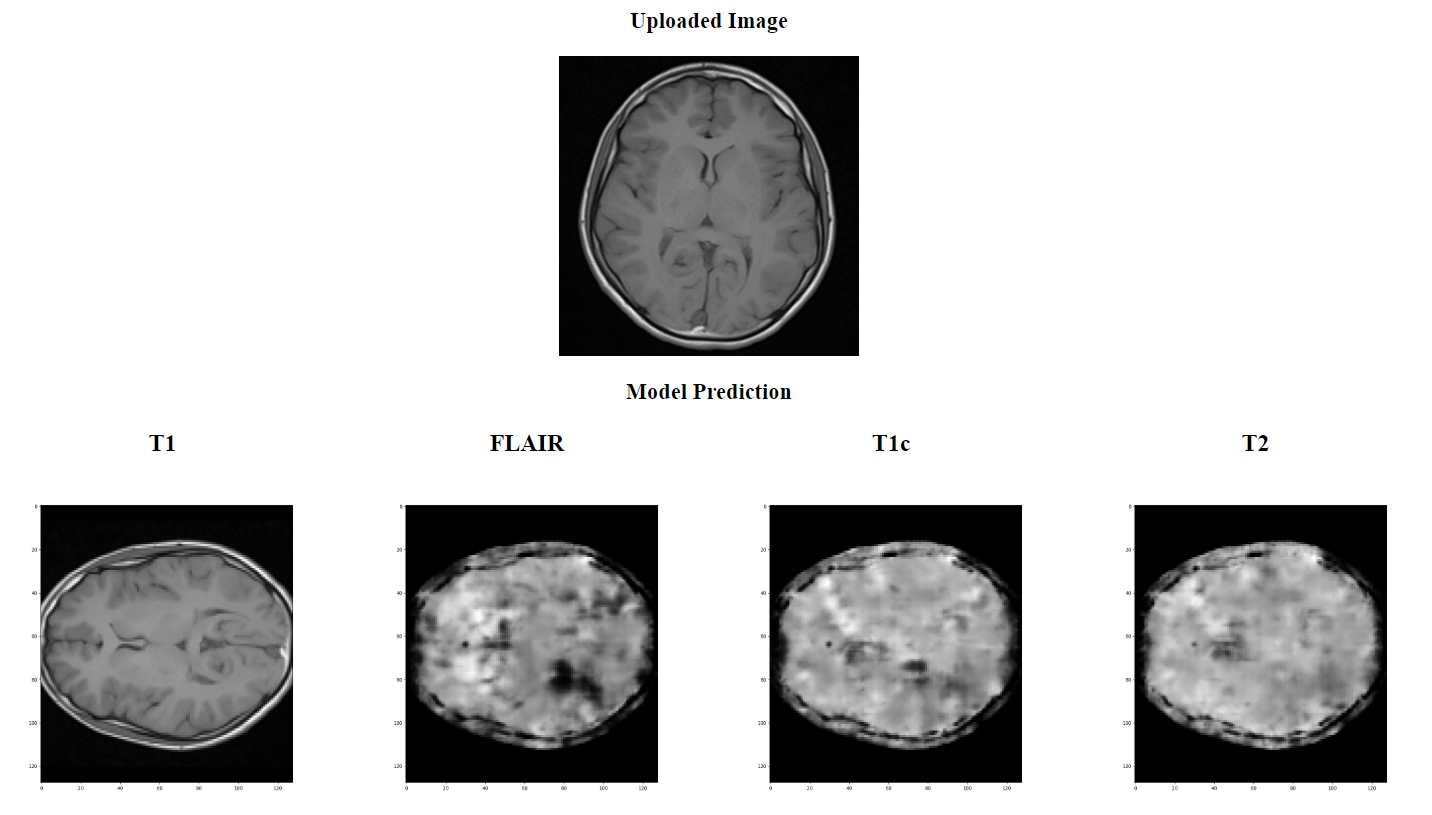
**Figure 40**- Le modèle SIMO, en tant que microservice, accessible sur le service local pour le débogage 127.0.0.1:5000



**Figure 41**- Le modèle SIMO, en tant que microservice, accessible sur le service local pour le débogage 127.0.0.1:5000



**Figure 42**- Le modèle SIMO, en tant que microservice, accessible sur le service local pour le débogage 127.0.0.1:5000



1. [www.zotero.org](http://www.zotero.org) [↑](#footnote-ref-1)