

**Running Head:** **AI for enhanced non-invasive medical imaging**

**University of Poitiers Academic year**

**SP2MI, UFR Fundamental and Applied Sciences**

**Master Connected Objects**

**Internship report**

**Addressed for the validation of Master 2: Engineering of intelligent objects**

**Artificial intelligence for enhanced non-invasive medical imaging**

Internship Period: Mars, - September

Submitted by

Marwan AL OMARI **Internship tutor**

BOURDON Pascal

XLIM laboratory CNRS

Bldg H1 - SP2MI

Referring teacher: Bd Marie and Pierre Curie

Noel Richard Poitiers Cedex

**Table of Contents**

[**List of Figures** III](#_Toc110590148)

[**List of Tables** V](#_Toc110590149)

[**Abstract** VII](#_Toc110590150)

[**Abbreviations and Acronyms** VIII](#_Toc110590151)

[**I. Laboratories and Internship Presentations** X](#_Toc110590152)

[**I.1 XLIM Laboratory** X](#_Toc110590153)

[**I.2 I3M Laboratory** X](#_Toc110590154)

[**I.3 Internship Presentations** XI](#_Toc110590155)

[**I.3.1 Specifications** XII](#_Toc110590156)

[**I.3.2 Description** XII](#_Toc110590157)

[**I.4 Project Management** XIII](#_Toc110590158)

[**I.4.1 Operating systems** XIII](#_Toc110590159)

[**I.4.2 Software programs** XIII](#_Toc110590160)

[**I.4.3 Coding Plateforms** XVI](#_Toc110590161)

[**1. Introduction** 1](#_Toc110590162)

[**2. Related Work** 1](#_Toc110590163)

[**3. Background** 6](#_Toc110590164)

[**3.1 Deep Learning in Medical Diagnosis** 6](#_Toc110590165)

[**3.2 CNNs** 2](#_Toc110590166)

[**3.2.1 Convolution (C1, C3, and C5)** 1](#_Toc110590167)

[**3.3.2 Pooling Layer (S2 and S4)** 2](#_Toc110590168)

[**3.3.3 FCL (F6)** 3](#_Toc110590169)

[**3.3.4 Activation functions** 3](#_Toc110590170)

[**3.3 U-Net** 2](#_Toc110590171)

[**3.4 GANs** 4](#_Toc110590172)

[**3.5 U-Net and GAN** 5](#_Toc110590173)

[**3.5 Dataset** 6](#_Toc110590174)

[**4. Approach** 7](#_Toc110590175)

[**4.1 Pre-processing** 8](#_Toc110590176)

[**4.1.1 Image Cropping** 8](#_Toc110590177)

[**4.1.2 One-hot encoding** 2](#_Toc110590178)

[**4.1.3 Normalization** 2](#_Toc110590179)

[**4.1.4 Filtration** 3](#_Toc110590180)

[**4.1.5 Linear intensity correction** 3](#_Toc110590181)

[**4.1.6 Bias correction** 3](#_Toc110590182)

[**4.1.7 Data augmentation** 3](#_Toc110590183)

[**4.1.8 Data split** 3](#_Toc110590184)

[**4.2 Algorithm** 4](#_Toc110590185)

[**4.2.1 SIMO** 5](#_Toc110590186)

[**4.2.2 MIMO** 5](#_Toc110590187)

[**4.2.3 MISO** 5](#_Toc110590188)

[**4.3 Evaluation Metrics** 6](#_Toc110590189)

[**5. Experiments and Results** 6](#_Toc110590190)

[**6. Conclusion** 19](#_Toc110590191)

[**6.1 Professional** 19](#_Toc110590192)

[**6.2 Personal** 19](#_Toc110590193)

[**6.3 Future Works** 19](#_Toc110590194)

[**6.4 Challenges** 19](#_Toc110590195)

[**6.5.2 Administration user non-grant** 20](#_Toc110590196)

[**6.5.3 The ambiguity of specification** 20](#_Toc110590197)

[**6.5.3 Prolonged training of DL architectures** 21](#_Toc110590198)

[**7. References** 21](#_Toc110590199)

[**Annex** 24](#_Toc110590200)

# **List of Figures**

[**Figure 1**- Representation of XLIM laboratory, which consists of scientific poles and scientific axes X](#_Toc110590201)

[**Figure 2**- Representative FLAIR images of a single subject at field strengths: -T, -T and -T, extracted from [23] XI](#_Toc110590202)

[**Figure 3**- GANT diagram - time management of tasks XIII](#_Toc110590203)

[**Figure 4**- Classical PERT chart - time management of tasks XIV](#_Toc110590204)

[**Figure 5**- Establishing SSH connection to the server in question for SFTP manipulation XV](#_Toc110590205)

[**Figure 6**- SFTP connection using FileZilla XV](#_Toc110590206)

[**Figure 7**- SISO, SIMO, MISO, and MIMO methods in handling MRI modalities 2](#_Toc110590207)

[**Figure 8**- LeNet-, an example of CNNs architecture, which extracted from [21] 2](#_Toc110590208)

[**Figure 9**- A deconstruction of convolutional operation in CNNs 1](#_Toc110590209)

[**Figure 10**- Upsampling example 2](#_Toc110590210)

[**Figure 11**- Max-pooling example with fliter and stride 2](#_Toc110590211)

[**Figure 12**- average-pooling example with fliter and stride 2](#_Toc110590212)

[**Figure 13**- Global Max-pooling example with pool size equal to input size 3](#_Toc110590213)

[**Figure 14**- Global average-pooling example with pool size equal to input size 3](#_Toc110590214)

[**Figure 15**- ReLU function in graphical representation 4](#_Toc110590215)

[**Figure 16**- LeakyReLU function in graphical representation 4](#_Toc110590216)

[**Figure 17**- Sigmoid function in graphical representation 2](#_Toc110590217)

[**Figure 18**- U-Net architecture 2](#_Toc110590218)

[**Figure 19**- The architecture U-Net, which extracted from [19][20] 3](#_Toc110590219)

[**Figure 20**- GANs concept 4](#_Toc110590220)

[**Figure 21**- Demonstration of generator-discriminator mapping, which extracted from [13] 5](#_Toc110590221)

[**Figure 22**- GAN architecture 5](#_Toc110590222)

[**Figure 23**- The assembly architecture of U-Net and GANs 6](#_Toc110590223)

[**Figure 24** – Data sample demonstrating the four modalities with segmentation map (Ground Truth) 6](#_Toc110590224)

[**Figure 25**- Program architecture, specifying the different steps of data entry, pre-processing and augmentation, algorithm, results and evaluation metrices 7](#_Toc110590225)

[**Figure 26** – The outcome of the pre-processing step, including bias-correction, normalization, filtration, and size cropping 8](#_Toc110590226)

[**Figure 27**- Image cropping from to , highlighting the zone of tumor 2](#_Toc110590227)

[**Figure 28**- Data augmentation, including: and rotations and left-to-right flipping 3](#_Toc110590228)

[**Figure 29**- Data split into testing and training sets for SIMO (T1 → T2, FLAIR and T1c), MISO (T1 and T2 → FLAIR), and MISO (T1, T2, and FLAIR → T1c) prototypes with image entry size of 4](#_Toc110590229)

[Figure 30- Data split into testing and training sets for MISO (T1, T2, and FLAIR → T1c) prototype with image entry cropping of 4](#_Toc110590230)

[**Figure 31**- SIMO architecture takes T1 as input, & synthesis T2, FLAIR, T1c, and segmentation map for discrimination 5](#_Toc110590231)

[**Figure 32**- MIMO architecture takes T1 and T2 as input, and synthesis T1c and FLAIR for discrimination 5](#_Toc110590232)

[**Figure 33**- MISO takes T1 and T2 as input, and synthesis FLAIR for discrimination 6](#_Toc110590233)

[**Figure 34**- MISO takes T1 and T2, and FLAIR as input, and synthesis T1c for discrimination 6](#_Toc110590234)

[**Figure 35**- U-Net dice score, a comparison between SIMO, MISO (output FLAIR), MISO (output T1c ), and MISO (output T1c ), 18](#_Toc110590235)

[**Figure 36**- Risk probability/severity matrix for the un-availability of dataset 20](#_Toc110590236)

[**Figure 37**- Risk probability/severity matrix for risk of administration user non-grant 20](#_Toc110590237)

[**Figure 38**- Risk probability/severity matrix for risk of specification ambiguity 21](#_Toc110590238)

[**Figure 39**- Risk probability/severity matrix for risk of prolonged models’ trainings 21](#_Toc110590239)

[**Figure 40**- The model SIMO, as a microservice, accessible on local service for debugging 127.0.0.1:5000 24](#_Toc110590240)

[**Figure 41**- The model SIMO, as a microservice, accessible on local service for debugging 127.0.0.1:5000 24](#_Toc110590241)

[**Figure 42**- The model SIMO, as a microservice, accessible on local service for debugging 127.0.0.1:5000 25](#_Toc110590242)

# **List of Tables**

[**Table 1**- PERT chart – tasks management XIV](#_Toc110590243)

[**Table 2**- Articles Summarization in respect to study number, dataset, used approach, experiments results, challenges and possible future works 4](#_Toc110590244)

[**Table 3**- BRATS2018 modality distribution over 285 patients 6](#_Toc110590245)

[**Table 4**- One-hot encoding of tumor and non-tumor labels 2](#_Toc110590246)

[**Table 5**- Laboratory computer specifications 6](#_Toc110590247)

[**Table 6**- U-Net dice score of the 4 architectures, SIMO achieved ACC, MISO (output FLAIR) achieved , MISO (output T1c ) achieved , and MISO (output T1c ) achieved 17](#_Toc110590248)

**Acknowledgments**

I devote my sincere thanks first of all to my main supervisor Pascal Bourdon, and my co-supervisor Christine Fernandez for the constant follow-up and directive instructions all along the project. Besides, I would like to thank my tutor Noel Richard for his insights on different aspects of project management. Also, I must thank the head of the master’s program Clency Perrine for his guidance and firm organization. Moreover, I would like to thank the XLIM staff, as I had the opportunity to work there and have the chance to establish new connections.

Moreover, I would like to devote my sincere thanks to Campus France for financing my master studies and a whole year of French language in France. I really appreciate the life-changing opportunity that has been given me, in Poitiers, France. Besides, I would not forget my family, they have stayed beside me supporting me on the difficult times during my master studies as well as through the internship. I must mention that I had not much knowledge in French language at the start of the master’s program. I could still remember the days I spent just being blocked, without any advancement, I was trying to figure out some words in language. But as everything else in this world, it gets better and better as the time rolls on.

# **Abstract**

Artificial intelligence has advanced in later years, showing enhanced capabilities in handling data for various applications of computer vision, including classification, segmentation, detection, etc. Generative neural network has proved its effectivity in capturing deeper features during the process of image synthesis. Based on the state-of-the-art, generation networks are competitive in comparison to traditional machine learning algorithms, in learning similar structures between input images. Hence, this research project develops improved single and multiple input and single output architectures for the goal of synthesizing medical imaging. These architectures, including SIMO (output T2, FLAIR, T1c), MISO (output FLAIR), MISO (output T1c 128×128), and lastly MISO (output T1c 80×80); are important as it saves much time on tumor diagnosis and it would be the cure to avoid injecting an additional substance, for example, gadolinium, to the brain while extracting T1c modality. The models are trained and experimented using hyper-optimization techniques to fine-tune the parameters on the BRATS2018 dataset. Next, the architectures have achieved competitive results in highlighting the state-of-the-art-performance.

**Keywords**: Artificial intelligence, generative neural networks, deep learning, medical synthesis

# **Abbreviations and Acronyms**

In this section, abbreviations and acronyms, which used in this report paper, are clarified to avoid the repetitiveness of the same terms. They are as follows:

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

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

**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)

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

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

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

**T1**-weighted (T1)

**T1** and **C**ontrast-enhance (T1c)

**T2**-weighted (T2)

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

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

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

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

**C**onvolutional **N**eural **N**etworks (CNNs)

**M**agnetic **R**esonance **I**maging (MRI)

**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*/**S**cientific **R**esearch **N**ational **C**enter (CNRS)

**P**rogram **E**valuation and **R**eview **T**echnique (PERT)

**O**perating **S**ystems (OS)

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

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

**C**onvolutional Patched**GANS** (CGANs)

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

**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)

**N**atural **L**anguage **P**rocessing (NLP)

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

**A**rtificial **I**ntelligence (AI)

**CO**rona**VI**rus **D**isease **19** (Covid-19)

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

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

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

**M**agnetic **R**esonance **I**maging (MRI)

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

**ACC**uracy (ACC)

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

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

**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)

**BRA**in **T**umor **S**egmentation Challenge **2015** (BRATS2015)

**BRA**in **T**umor **S**egmentation Challenge **2018** (BRATS2018)

**L**earning **R**ate (LR)

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

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

**C**entre **H**ospitalier **U**niversitaire/**U**niversity **H**ospital **C**enter(CHU)

**I**magerie **M**étabolique **M**ulti-noyaux **M**ulti-organes (I3M)

# **I. Laboratories and Internship Presentations**

In this section, a brief introduction is represented about XLIM and I3M laboratories as well as their different domain specialties. Then, an overall presentation of the internship position will be discussed along with its desired objectives and required missions, and most importantly the project management.

## **I.1 XLIM Laboratory**

XLIM numbered as , as a one of the French research centers, is known under the name of CNRS [7]. The laboratory is greatly centered on electronics, optics, photonics, mathematics, computer science, image processing, telecommunication, network security, bioengineering, and energy. A representation of XLIM divisions is presented below in Figure (1).

**Figure 1**- Representation of XLIM laboratory, which consists of scientific poles and scientific axes

Une image contenant texte, signe

Description générée automatiquement

XLIM is a multidisciplinary research institute, located on several geographical sites, in Limoges on the sites of the Faculty of Science and Technology, the ENSIL, Ester-Technopole, on the University Campus of Brive; and on the site of the Futuroscope Technopole in Poitiers. It brings together more than teacher-researchers, CNRS researchers, engineers, administrative staff technicians, doctoral and post-doctoral students.

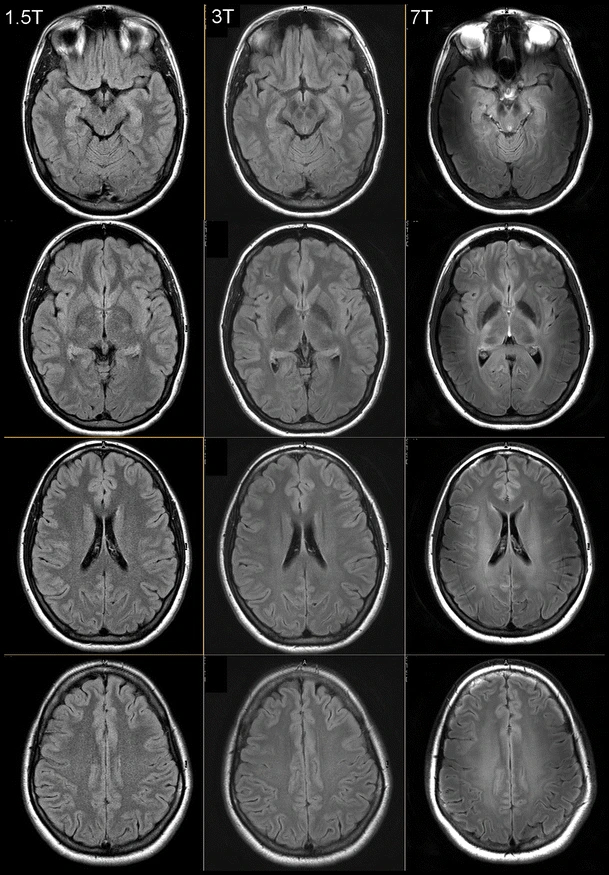
The main research of XLIM is mainly relied on two platforms. On one hand, a platform gives researchers access to technological equipment for the creation of optical structures. For instance, a large pool of lasers and instruments are available for experimental characterization of electronic, optical, electromagnetic, and radiating devices.On the other hand, the laboratory engaged in modeling and simulation activities as it is equipped with a software forge, which is available to its members and their external collaborators. For that, it offers multiscale simulation of complex systems based on interface models between physics, mathematics, and computer science.

## **I.2 I3M Laboratory**

I3M was founded in February , a collaboration between CNRS, SIEMENS HEALTHINNERS, CHU Poitiers, and the Poitiers University [24]. It is a distinctive transdisciplinary research center based on metabolic imaging of organs in France and even in Europe. The laboratory uses automatic processing and multi-modality analysis with the help of AI for the objective of multivariate MRI images as well as helping with diagnostic and therapeutic follow-up in brain, heart, and renal disorders. Besides, I3M uses ultra-high field -Tesla MRI, which is considered the rd in the entire world [25].

-Tesla MRI enables access to extremely high-resolution imaging of molecules and metabolism in addition to anatomy and organ function. It makes it possible to measure organ function and structure in an unmatched way. For example, Figure (2) represents a FLAIR image comparison among -T, -T and -T of a single subject. Therefore, the FLAIR images, captured by 7-T, have a higher SNR and much better contrast in comparison to others generations of Tesla MRI.

**Figure 2**- Representative FLAIR images of a single subject at field strengths: -T, -T and -T, extracted from [23]



## **I.3 Internship Presentations**

During my internship, I have been a research engineer, working on the development and experimentation of DL architectures for solving the problem of generating different MRI sequences, mainly on the generation of fake modalities on BRATS. I have acquired many skills, which will be my next source for knowledge-making and strategy development in the future. Some of the skills I have learned summarized in the following:

1. Network management and programming of a distance computer using SSH and SFTP
2. Prototyping an algorithm for research enhancement
3. Project planification and self-management

The workplace of the internship takes place in two different laboratories and on different time intervals:

* XLIM: presence From March to July and from August to September.
* I3M: presence From July to August.

### **I.3.1 Specifications**

The main specifications of the internships, as follows:

1. Developing a solution for constructing MRI sequences using artificial intelligence algorithms.
2. Producing a prototype in Python language.

In the following section, I will describe and give extra details on the missions during the whole period of internship.

### **I.3.2 Description**

To begin with, I was responsible in discovering BRATS2018 dataset in a medical context aside from its structural and representational complexity. Then, I analyzed the dataset in conducting different manipulations using python programming language. I discovered the data further with representational graphs using Matplotlib visualization.

In addition, I developed SIMO and MISO DL architectures, using GANs along with U-Net algorithms, to predict certain modalities from one or multiple input(s). Therefore, I identified the state-of-the-art solutions and algorithms for enhancing the results, including pre-processing methods and algorithm structures. I experimented different methods for fine-tuning the GANs algorithm, including pipeline and sample testing.

During fact-to-face as well as web meetings, I had to present the results that I came up in each step; so, through presentations, my French language was improving steadily as I was writing weekly and monthly reports and emails to my supervisor as well as my pedagogical supervisor.

In addition to the technical tasks, my skills, as a part of XLIM’s internship program in medical imaging, developed massively in communication, organization, research independence, analytical vision, curiosity, and creativity. I have learnt to present ideas with good communication skills in both verbal and written forms. The writing is not free of limitations though I have tried my best to keep the writings free from errors.

In daily bases, I was learning new things that had shaped my mind in data science and AI. I believe I have proved myself as an essential member in providing revolutionary ideas and innovative solutions to improve image synthesis. However, I have faced many problems in setting up the environment for training the algorithms as I have between laboratories and in each time I have different equipment to use, without the right of admin. I could say that everything depends on a hierarchy so if I wanted something, as to install a program, I must wait for it as I have no authority on the computer. Thus for the first phase, I found out using my personal computer would save much time on developing the different prototypes of GANs architectures.

Nevertheless, I have been a bit limited in my personal computer, which it was crashing in training with huge amount of data. Later, I had the accessibility to a server at a distance with Quadro RTX, which I ran all the experimentations on. At the end, I have run the final experimentations on another server at a distance that is equipped with Nvidia Tesla.

As for importance, I have found organization as one of the essential part of time management and task handling of different tasks in each stage of advancement. I have remarked that data science and DL require a dynamic and analytical mind in visualizing fitted solutions while covering a wide literature background in related work in medical imaging. Practically, the literature review has widened my horizons to pass certain challenges and lessons that other researchers have been through.

In sum, I have really obtained great insights into the key performance and potential capacity of image synthesis through data analysis and algorithmic development. I believe they are rarely manifested in French research laboratories, especially after the difficult time of Covid-19 pandemic.

### **I.4 Project Management**

In this section, I am going to present the software I have used in the internship, starting from operating systems, programs, and coding platforms.

### **I.4.1 Operating systems**

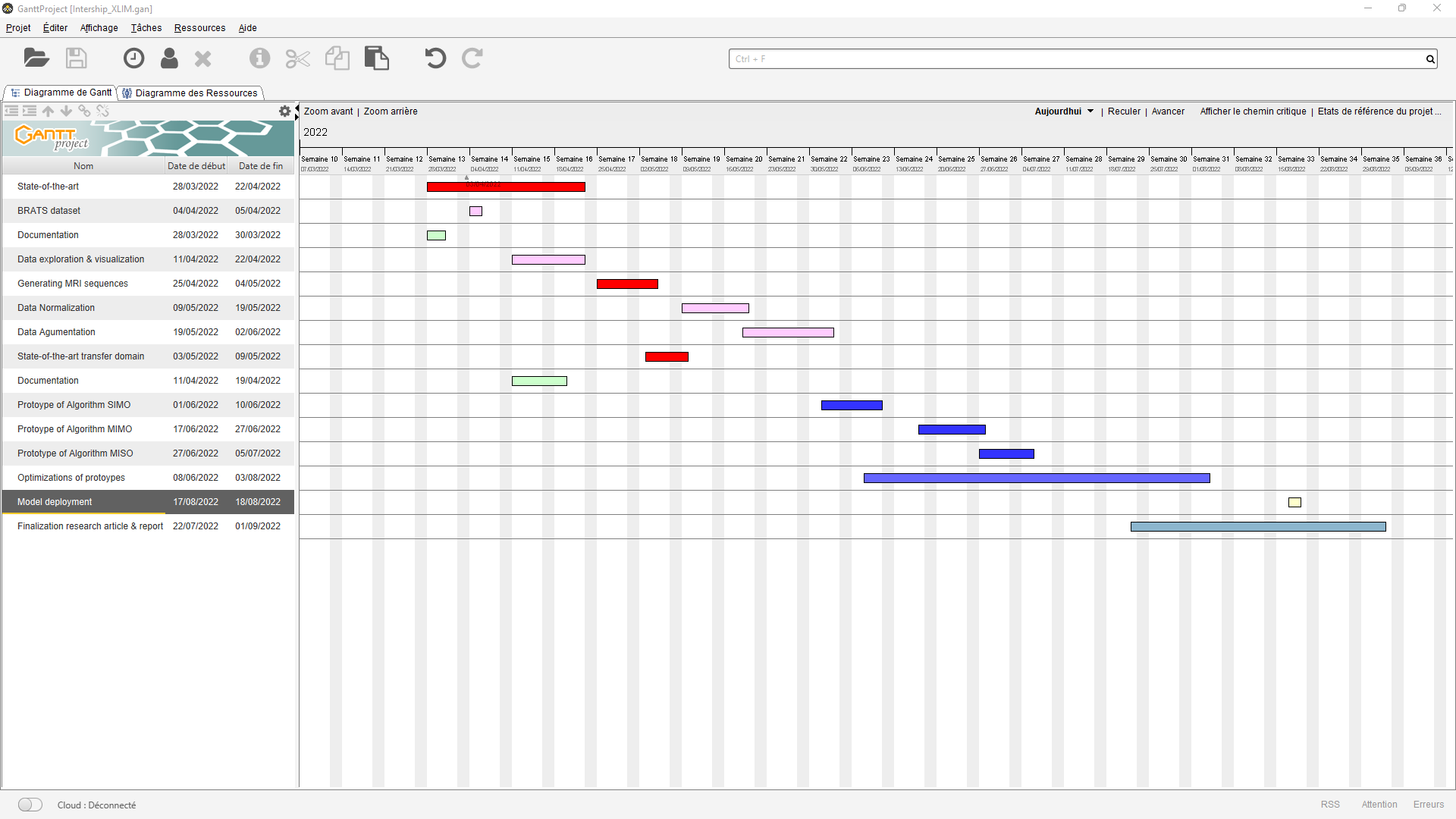
**Linux** and **Windows** OS are system software that manages computer hardware, software resources, and provides common services for computer programs. So, it is considered the fundamental software that could manage the other software that we use, for example, anaconda.

### **I.4.2 Software programs**

**SourceSup service**, which operated through RENATER, is a management web platform for Higher Education and French Research organizations. Every member of the network can create a project on the platform in permitting additional collaboration of outside people to the created project.

**GanttProject** breaks down the work, build a Gantt chart, assign resources, and calculate project costs. We used it to debrief the project tasks for the ongoing weeks through performed and still in progress tasks.

**Figure 3**- GANT diagram - time management of tasks



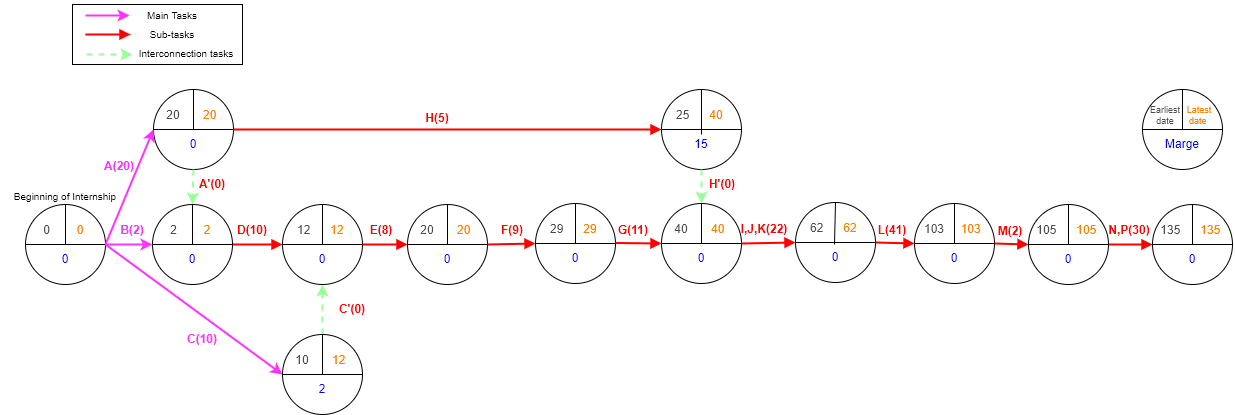
As in the above Figure (3), we could see the current, finished, and continuous tasks that I have been working on. For example, from the period between to , I have covered the state-of-the-art concerning GANs and medical synthesis.

**PERT**,a tool for planning, organizing, and scheduling tasks within a project, represents the project schedule and categorize individual tasks. PERT charts are like Gantt charts, but with a different structure. In the following Table (1) and Figure (4), we will present the main tasks and its representations in respect to priority and its earliest and latest duration.

**Table 1**- PERT chart – tasks management

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Nature** | **Duration** | **Anteriority** |
| A | State-of-the-art | 20 | - |
| B | BRATS dataset | 2 | - |
| C | Documentation | 10 | - |
| D | Data visualization and exploration | 10 | B |
| E | Generating MRI sequences | 8 | D, C |
| F | Data normalization | 9 | A, B, C |
| G | Data augmentation | 11 | F |
| H | Enrichment state-of-the-art | 5 | A |
| I | Prototype SIMO | 8 | H, G |
| J | Prototype MIMO | 7 | H, G |
| K | Prototype MISO | 7 | H, G |
| L | Optimizations of prototypes | 41 | I, J, K |
| M | Model deployment | 2 | I, J, K, L |
| N | Writing research article | 15 | I, J, K |
| P | Final Report | 15 | I, J, K |
| **Total** | 15 | 24 weeks 1190 |

**Figure 4**- Classical PERT chart - time management of tasks



The key element that we could notice in PERT chart, we could estimate the earliest date of completing a task and the state of freedom, as in task H. Therefore, the task could be delayed days more because the other tasks, for example, D, E, or F, they are not dependent on H.

**draw.io** is a free online diagram software. It is used to write and simplify algorithm, code, and information into a visual schema that is easy to be understood.

**Webex**, by Cisco, is a software for video conferencing, online meetings, screen sharing, and webinars. It was used to report results, problems, and complaints to the supervisor as well as for planning the weekly meetings.

**ENT Zimbra** is a collaborative space to write collaborators of internship, share messages, and create appointments.

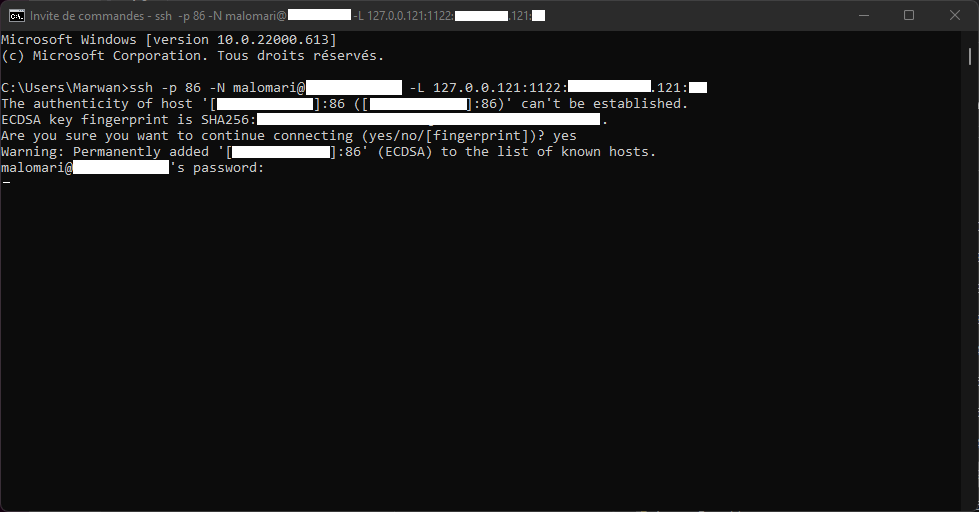
**Microsoft Office** is a text processing application that allows the configuring attributes of a document, such as layout, the styles of content, and to add their content in a variety of ways and formats to produce documents. It is used to write the weekly reports as well as the final draft of the report.

**Microsoft PowerPoint** is a presentation program, which uses slides to convey rich information supported by different multimedia.

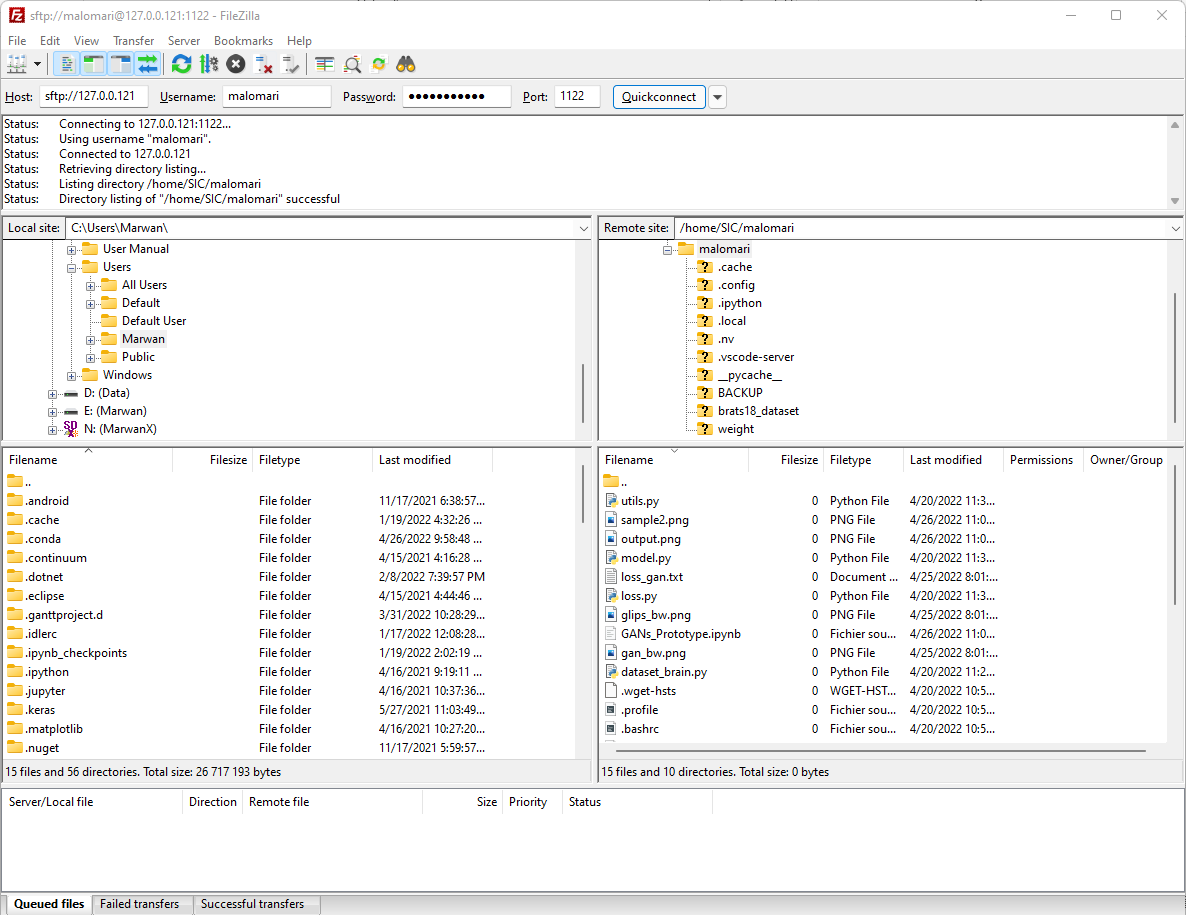
**Zotero[[1]](#footnote-1)** helps to collect, organize, cite, and share different research sources online.

**FileZilla** is a free and open-source, cross-platform FTP application, consisting of FileZilla Client and FileZilla Server. We used FileZilla client, which supports FTP, FTPS and SFTP server connections. An example of network connection made to the server in question represented in the following two Figures (5) and (6). The first Figure (5) establishes connection to the distant machine by its IP at address , port . In FileZilla, we input the coordination that is well established in the terminal to interrogate the server in question, as it is well specified in Figure (6).

**Figure 5**- Establishing SSH connection to the server in question for SFTP manipulation



**Figure 6**- SFTP connection using FileZilla



### **I.4.3 Coding Plateforms**

**Juypter Notebook** provides services for interactive computing using python. Therefore, as coding in python, Juypter will visualize the results on a same script window. Thus, it facilities the debugging of the algorithm or the application code.

**Anaconda** is a distribution of Python programming language for scientific computing, including data science, ML applications, large-scale data processing, predictive analytics, etc. It is used for the aim of simplifying package management and deployment.

**Microsoft Visual Studio Code**, referred to as VS code, is a code editor. We used it to facilitate the SSH connections and code integration in same workspace. Besides, it is easier to use and handle Jupyter notebook files without a local server opening in the background, that is represented above in Figure (5).

# **1. Introduction**

Image processing employs different and multiple methodologies from hand-forged features to ML for the objectives of detection, classification, and segmentation. As consective development in AI, DL employs deeper processing units called neurons to learn hierarchical features and representations of data [8]. In this case, DL methods outperform humans’ predictions as it is going through impressive advancement and evolvement in different fields of science, for example, computer vision [9].

In the context of medical imaging, image classification and regeneration is generalized to distinguish the different classes and categories in experimentation with parameter fine-tuning in auto-generative tasks [13]. Generative networks, including collection style transfer, object transfiguration, season transfer, photo enhancement, etc. [15][16].

As for the aim of this internship, we have, therefore, adopted GANs model to generate different attributes from a single and multiple attributes as an input. The proposed architectures save time in predicting medical imaging attributes and avoiding injecting certain substance to the patients. For instance, in the case of T1c, the patient is injected with gadolinium to extract a modality with better contrast. The data of BRATS, being used in this study, are widely used for the improvements of brain lesions diagnosis and segmentation. Practically, BRATS2018 includes patient file images all of which contains four modalities (attributes) and a segmentation map (ground truth): T1, T1c, T2, and FLAIR.

The works of this project are specified in sections. The first section is already introduced the subject under study. Later in the second section, we would discover the related works that are well provided on image synthesis. Section details the DL background in the context of medical synthesis along with the BRATS2018 dataset. The fourth section would specify the research approach in discussing the pre-processing steps, algorithms, and the evaluation metrics. The fifth section, we would present the results of the experiments with dedicated analysis. The sixth section concludes the whole idea of the research project in highlighting the challenges along with further improvements in the future. Lastly, there would be the reference list that is numbered all through the paper for navigation simplicity.

# **2. Related Work**

In this section, we would present the main research articles that experimented on open-source datasets from different challenges, for example, CelebA, RaFD or BRATS [4]. The aim of covering the related work of using the architecture of GANs is to understand better the literature and the employed techniques in the researchers’ approaches with respect to our project study. At the end of this section, we would provide a briefing of the studies in Table (2).

[3] Their multimodal network assessed on four contrasts of T1, T1c, T2, and FLAIR by calculating NMAE, PSNR, SSIM, VIF, NIQE. They adopted StarGAN strategy [2] in mapping the four MRI modalities through SISO models (one-to-one modality), and a unified GAN to map them together. Their method was evaluated on BRATS2015, which consists of subjects in which, patients with LGG, and patients with HGG. The original size of each image in the dataset is . For preprocessing, each image is narrowed to patch size. Then, the final estimation of overlapped regions was set to with -fold cross validation. Most of the important parameters in their model are set to , , , , and for balancing the weights of classification, synthetic consistency, CC, and adversarial losses, respectively. As a conclusion, for example, with T1 as input modality, the NMAEs for the generated T1c, T2, FLAIR are , , and , the PSNRs are dB, dB, and dB, the SSIMs are , , and , the VIF are , , and , and NIQE are , , and , respectively. Finally, their study is limited to spatially co-registered multimodal images before even they were used for training and testing on a small dataset. For future work, they will try to increase the amount of training and testing images through augmentation techniques.

[2] StarGAN, a unified GAN, handles multi-domain image-to-image translations using a single model of one-to-one discriminator and generator. A single model takes in an input of an image and domain information by one-hot encoding. It learns the mappings between all available domains, using only one generator. Training process generates random target domain label as to perfect image translation. Due to the limitations single models that could not be generalized to more than an individual task, the researcher proposed a novel approach that could handle multiple domains using a single generator and a discriminator. Their approach works fine on facial feature transfer and facial expression synthesis tasks. Their architecture was tested on two datasets. Firstly, CelebA dataset contains labels related to facial features such as hair color, gender, and age. Secondly, RaFD dataset contains labels for facial expressions as like happy, angry, and sad. Besides, to avoid missing values in the dataset, they applied a mask vector of domain label that works on ignoring unknown labels and focus on what is already available. On CelebA, StarGAN achieved , , , , , , for Hair color, Gender, Aged, H+G, H+A, G+A, and H+G+A, respectively. On RaFD: StarGAN achieved losses with parameters. On both CelebA+RaFD, their model properly learned the intended role of a mask vector in image-to-image translations when all the labels included altogether from multiple datasets. In conclusion, StarGAN generated images of higher visual quality compared to existing methods.

[13] Pix2pix, U-Net generator and CGANs discriminator, combines an adversarial loss with a L1 loss for capturing the low frequencies. Progressively, their model downsamples the input images from a high-resolution grid to a bottle neck layer of skip connectors (concatenates the channels from previous layers) followed by U-Net. Their algorithm proved to be effective as well as PatchGAN (runs patches that could capture high frequencies on image structure). For algorithm’s optimization, they had used minibatch SGD and ADAM, with a LR, and β1 = , β2 = momentum parameters. Also, various batch normalization had been used, ranging from to. They trained their CGANs on different tasks and datasets, including Cityscapes, GMP Facades, Google Maps photos, BW to color photos, edges of photos, human-drawn sketches, day to night images, thermal to color, and finally missing pixel photos top inpainted photo. They declared even though with a small training size; they were able to achieve a descent result on a single Pascal Titan X GPU. As a limitation to measure structure losses in applying traditional metrics as per-pixel MSE, they applied joint method. First, a test of map generation, aerial photo generation, and image colorization conducted for solving graph problems on AMT. Secondly, a metric system was used for recognizing the realistic objects match in images. As a conclusion, L1 loss, CGAN, and L1+CGAN achieved on cityscapes , , and per-pixel ACC, and , , and per-class ACC, respectively.

[14] UNIT combines VAEs with CoCAN, where two generators share weights to learn the distribution of images in cross domain. Shared-latent space implies CC mapping between source and target domains, interchangeably. They employed weight-sharing constraint to relate the weights of the last layers of VAEs, extracts encoded high-frequencies, for decoding. They used ADAM with LR and to momentums. The batch size of one image from different domains. Their framework has different parameters of , and . The encoders consisted of convolutional layers and basic residual blocks. On the other hand, the generators consisted of basic residual blocks as the front-end and transposed convolutional layers as the back-end. On the other hand, the discriminators consisted of stacks of convolutional layers with nonlinear LeakyReLU. They used map dataset, which contained corresponding pairs of images in two domains (satellite images and maps). for iterations, The experiment operated in an unsupervised setting using satellite images from training set as a first domain and maps from the validation set as a second domain. The corresponding ground truth maps pixel-wisely where each pixel translation was counted correct only if the color difference was within of the ground truth color values. As a performance metric, they used average pixel ACC. The algorithm achieved classification accuracies: SVHN→MNIST , MNIST→USPS , USPS→MNIST .

[15] DiscoGAN and [16] CycleGAN save information attributes among input and translated images by using a CC loss. However, they all suffer from limited generalization as they are SISO models.

On one hand, DiscoGAN, a cross-domain GANs without any explicit pre-training sets, takes an input image in one domain and generates its correspondence one. They proposed coupled GANs in forcing generated images through a reconstruction loss to be a duplicate of the original one, for example, a handbag image. In experiments, all input and translated images are size. Some of the adopted parameters are LR, minibatch size, and ADAM optimizer with and . Besides, batch normalization was applied to all convolution and deconvolution layers except the first and the last layers. Also, the weight decay coefficient varies between . Data images vary azimuth rotation from to . Their experiments summarized on car to car and face to face datasets. Car dataset consists of rendered images of 3D car models with varying azimuth angles at intervals. For evaluation, they translated images in the test set and their azimuth angles were predicted using a regressor. The proposed model shows strong correlation between predicted angles of both input and translated images, which successfully discovers azimuth relation between two domains. In terms of rotations between a standard GAN and a reconstructive loss GAN, the generated images do not vary as much as the input images. In car experiment, the two models suffered from sudden collapse. In addition, they experimented face conversion by face attributes on CELEBA, facescrub datasets, where only one feature, such as gender or hair color, varies between two domains and shared ones. The translated results have not only similar colors and patterns, but they also have a similar level of fashion formality as the input fashion item. Future works will be directed to algorithm modification in handling mixed modalities (e.g., text and image).

On the other hand, CycleGAN, using two CC losses that capture image translation from one domain to another and vice versa. The architecture contains three convolutions, residual blocks, two fractionally-strided convolutions with stride, and a convolution that maps feature to RGB space. Their network contains blocks for images and blocks for ones. For the discriminator, they used PatchGANs with instance normalization. They used different LGAN and oscillation techniques to stabilize their model. Firstly, LGAN replaces negative log like-hood objective through least-squares loss. Secondly, a discriminator oscillation updates uses a history of generated images rather than the ones produced by the latest generators. Some important parameters are , ADAM LR, batch size of , and a linear decay to zero over epochs. The experiments achieved positive results, but it failed on some translation tasks involving color, geometry, and texture changes, for example, cat to dog or horse, and a rider on a horse to zebra. Also, semantic weakness was spotted in photo labelling tasks, therefore, translation ambiguity formed substantial boundaries. To conclude, the model achieved real labels on Map to Photo , and on Photo to Map. Besides, it achieved per-pixel ACC, per-class ACC, and mean class IOU for labels to photo on Cityscapes dataset.

**Table 2**- Articles summarization in respect to study number, dataset, used approach, experiment’s results, challenges and possible future works

| **Study** | **Datasets** | **Approach** | **Results** | **Problems and Challenges** | **Future work** |
| --- | --- | --- | --- | --- | --- |
| [3] | **Description:**  BRATS2015 has subjects of modalities: T1, T1c, T2, and FLAIR. The size of each image: .  **Pre-processing:**   * images * regions   **Validation set:**   * -fold CV | **Models:**  - SISO models and a unified GAN  **Configurations:**  - Loss parameters:  - Evaluation metrics:   * NMAE * PSNR * SSIM * VIF * NIQE | T1 as input:  - NMAEs:   * T1c * T2 * FLAIR   - PSNRs:   * T1c * T2 * FLAIR   - SSIMs:   * T1c * T2 * FLAIR   - VIF:   * T1c * T2 * FLAIR   - NIQE:   * T1c * T2 * FLAIR | - Spatially co-registered multimodal images  - Small dataset | - Data augmentation techniques |
| [2] | **Description:**  - CelebA dataset contains labels of facial features, e.g. hair color, etc.  - RaFD dataset contains labels of facial expressions, e.g. happy, etc.  **Pre-processing:**  - Label vectorising | **Models:**  - A unified GAN of multiple one-to-one discriminator and generator  **Configurations:**  Not specified | - ACC CelebA:   * Hair * Gender * Aged * H+G * H+A * G+A * H+G+A   - Loss RaFD: | Not specified | Not specified |
| [13] | **Description:**  - Cityscapes  - GMP Facades  - Google Maps  - BW to color  - Humandrawn sketches  - Day-to-night  - Thermal to color  - Missing pixels  - Top inpainted  **Pre-processing:**  - Map generation  - Image colorization  - Metric system | **Models:**  - U-Net  - CGANs  **Configurations:**  - Adversarial & L1 losses  - Downsampling  - Minibatch  - LR SGD & ADAM  - & Momentum  - batch size | L1 loss, CGAN, & L1+CGAN ACC on cityscapes:  - Per-pixel:   * 86% * 76% * 83%   - Per-class: | - Small dataset  - Structural losses of traditional metrics as per-pixel MSE | Not specified |
| [14] | **Description:**  Map dataset contains pairs of images in two domains: satellite images and maps.  **Training set:**  satellite images as first domain  **Validation set:**  maps as second domain | **Models:**   * VAEs * CoCAN * CC loss   **Configurations:**  - Weight-sharing  - ADAM LR  - momentums  - LeakyReLU  - batch size  - Pixel translation  - iterations  - Averaged pixel ACC  - Loss parameters: | Classification ACC:   * SVHN→MNIST: * MNIST→USPS: * USPS→MNIST: | Not specified | Not specified |
| [15] | **Description:**  - Car dataset  - CELEBA  - Facescrub  **Pre-processing:**  **-**  images  - Azimuth rotation from to | **Models:**   * Coupled GANs   **Configurations:**  - LR  - Reconstruction loss.  - minibatch  - Batch normalization  - weight decay  -ADAM optimizer: | - Discovering azimuth relation between domains | - Limited  generalization of SISO model  - Sudden model collapse | - Algorithm  modification in handling mixed modalities (e.g., text and image) |
| [16] | **Description:**  - Cityscapes  - Map to Photo  **Pre-processing:**  - images  - images  - images | **Models:**  - PatchGANs  - convolutions  - Residual blocks   * blocks * blocks   - fractional convolutions  - stride  - RGB convolution  **Configurations:**  - Instance  normalization  - CC loss  - Least-squares loss  - ADAM LR  -  - batch size  - Linear decay over epochs | - Map to Photo:  - Photo to Map:  - Cityscapes:   * per-pixel ACC * per-class ACC * mean class IOU | - Limited  generalization of SISO model  - Semantic weakness resulting in translation ambiguity  - Translations Failure on  color, geometric, and texture changes | Not specified |

In conclusion, from the literature, we could notice a research focus on SIMO method, as we have discussed from the following articles [2][3][15][16]. From there, we devoted our research work to introduce MISO architecture, inspired from [2][3]. It takes the advantages of the leading performance, discussed in literature, to predict a sole attribute from multiple input. The main research focus is to predict T1c from multiple inputs as to reduce MRI examination time as well as avoiding injecting gadolinium.

# **3. Background**

In this section, we would discuss the DL in the context of medical diagnosis, highlighting the GANs architectures as well as a detailed description of the BRATS2018 dataset.

## **3.1 Deep Learning in Medical Diagnosis**

The AI is promising in the medical diagnosis situation through avoiding susceptible mistakes and harmful medical errors. DL, based on ANNs with representation learning in reinforcement, supervised, semi-supervised, and unsupervised methods [10], enhance the diagnosis of symptoms, which are tricky to spot even by the best experts. As a part of the broader family of ML methods, DL architectures such as DNNs, RNNs and CNNs have been applied to diverse fields, including computer vision, NLP, MT, and medical image analysis, where they have produced competitive results that would surpass human performance [8].

In general, ANNs were inspired by information processing and distributed communication nodes in biological systems. ANNs have differences from biological brains in the tendency to be static and symbolic, while the biological brain tends to be dynamic and analogue. However, the main core of DL refers to the use of multiple layers to progressively extract higher-level features from a raw input. For example, in image processing, lower layers may identify edges, while the higher layers may identify the concepts relevant to humans such as digits or letters or faces.

Nowadays, DL remains the most promising and widely used ML methodology for radiology and disease detection in general. It comes as no surprise as diagnostic imaging prevails in clinical diagnosis and image recognition. In 2016, Geoffrey Hinton, a notable computer scientist and researcher, predicted that radiologists and specialists who diagnose diseases from medical imaging like X-rays, CT scans, and MRI; they would soon lose their jobs. “People should stop training radiologists right now,” he announced, “It’s obvious that within five years deep learning is going to do better than humans,” [8].

The advancement of DL makes it much useful but not much reliable to the level that Hinton indicted by which machines would replace human experts. However, DL is still used to support the doctors in pre-selection and prioritize cases, but not as a main tool to diagnose patients.

From here, there are diverse use of DL in radiology and other diagnostic practices:

* Detecting the Neurological Abnormalities.
* Screening of the Common Cancers.
* Identifying the Infections in Kidney & Liver.
* Brain Tumor Detection with High ACC.
* Dental Imaging Analysis.
* Detecting the Bone Fractures and Musculoskeletal Injuries.

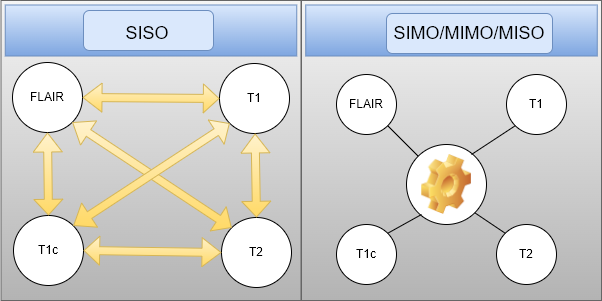
As one of the DL practices, medical image synthesis is an alternative to multiple pulse sequences for acquiring multiple contracts MRI [3]. MRI scan is non-standardized process across different institutions, cross-modal image synthesis proposed to tackle this challenge in providing missing modalities. It is used in clinical practice due to its capacity in providing useful information, for example, the four contracts:

* T1 distinguish white and grey matters.
* T1c assess the change of tumor shape with enhanced demarcation around tumor.
* T2 shows fluid obviously from cortical tissue.
* FLAIR shows contours of lesion.

In research and medical practices, there are different architectures could be considered whenever applying DL. They are summarized in the following list and Figure (7):

1. **SISO**: a target image generated from a given source image.
2. **MISO**: overcome limitations of SISO when source and target images are weakly correlated through learning shared latent representations.
3. **MIMO**: synthesizing one or more modalities from an input of MRI modalities.
4. **SIMO**: where only single modality is available as input, but multiple contrasts are necessary in output.

**Figure 7**- SISO, SIMO, MISO, and MIMO methods in handling MRI modalities

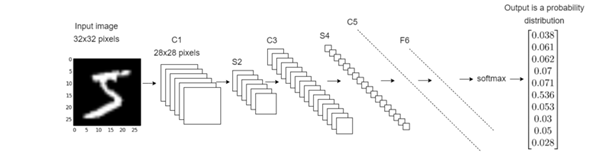


In observation, MRI of different modalities could provide complementary information for medical diagnosis, but also it is challenging and much expensive to access all the modalities. Many methods focus on a single modality to a modality synthesize, which marks a grand limitation in generalizing the outcomes to other modalities. Therefore, for each two modalities, we should develop a separate model so we could be able to map them together. To address this challenging problem, we propose a multiple modality GANs to synthesize three MR modalities (FLAIR, T1 and T1c) from one MR modality T2. Another challenging area is predicting FLAIR or T1c, as avoiding injecting substance (e.g. gadolinium) to the patient’s brain, from different input: T1 and T2 or T1, T2, FLAIR. The proposed architectures will be detailed later in [**4.2 Algorithm**](#_4.2_Algorithm).

## **3.2 CNNs**

As both U-Net and GAN architectures are basically based on CNNs, the architecture is detailed in the present section by taking the LeNet- architecture, published by Yann LeCun in 1998 [22], as an example. It is well presented in the following Figure (8):

**Figure 8**- LeNet-, an example of CNNs architecture, which extracted from [21]



CNNs or ConvNets is commonly used in computer vision tasks of classification, detection, and regression. In general, the CNNs take an input in shape of an array or matrix, for example, an image. In this case, it would be based on an array of pixels and would depend on its resolution. The proposed input would be in shape of height (h) width (w) dimension (d). The d could refer to RGB by indicator and for grayscale images. To proceed into training and testing the CNNs model, each input image passes through a series of convolution layers with a specific filters (kernels), pooling, FCL, and loss functions for finally having a probabilistic classification. As for a clear explanation of CNNs, LeNet-, would be used to demonstrate the different layers of CNNs as well as for U-Net and GAN architectures.

* Convolution (C1, C3, and C5)
  1. Up-Convolution
  2. Transposed convolution
* Pooling Layer (S2 and S4)
* FCL (F6)
* Activation functions

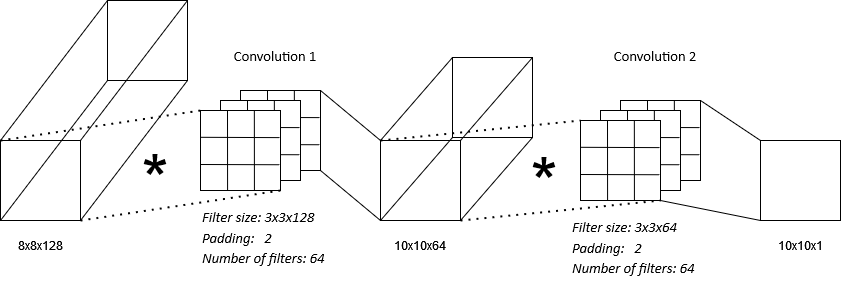
### 

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

The first layer that would extract features from the input image in preserving the relationship between pixels by learning image features using small squares of data. It uses a mathematical operation of two inputs as like image matrix and a kernel. Such filters, for example, edge detection, blur, and sharpen. The filter makes use of stride, which is the number of pixels shifts over the input data. So, if the stride is set to , the filter would move pixels at a time and so forth.

* For example, the convolution to increase the width and height and reduce the depth of the input, e.g. image, would apply two consecutive convolutional layers, as it is well visualized in Figure (9):
  1. First convolution: filter, padding, number of filters. Results: .
  2. Second convolution: filter, padding, number of filters. Results: .

**Figure 9**- A deconstruction of convolutional operation in CNNs



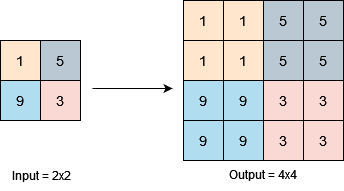
#### **3.2.1.1 Transposed convolution**

On the other hand, transposed convolution, known as deconvolution, is the inverse operation of convolution. It works on compressing the dimension of features and enlarging their size.

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

Up-Convolution or Upsampling increases the width and height of input image by duplicating pixels’ values for each layer without additional weights and any other complex operations. An example of up-convolution is represented in the following Figure (10), where there is a smaller input, and the output is up-sampled by performing duplications.

**Figure 10**- Upsampling example

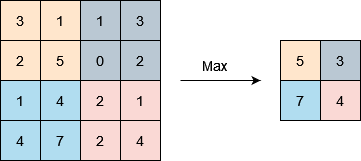


### **3.3.2** **Pooling Layer (S2 and S4)**

Whenever the input is large, downsampling or subsampling works in reducing the number of parameters. There are different types of pooling, as specified in the following:

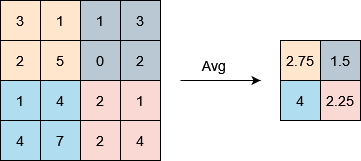
1. **Max Pooling** takes the largest element from the feature map.

**Figure 11**- Max-pooling example with fliter and stride



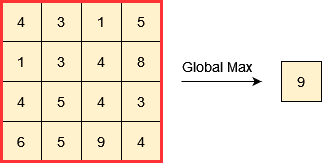
1. **Average Pooling** takes the average element from the feature map.

**Figure 12**- average-pooling example with filter and stride



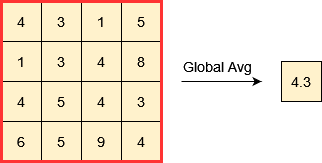
1. **Global Pooling** has two types, which detailed in the following sub-section:
   1. **Global Max Pooling** takes the largest number from a global input window.

**Figure 13**- Global Max-pooling example with pool size equal to input size



* 1. Global Average Pooling takes the total average from a global input window.

**Figure 14**- Global average-pooling example with pool size equal to input size



**3.3.3 FCL (F6)**

The FCL layer flattens the input from the previous layer into a vector. The FCL fed the vector by ANNs, where later an activation function, as for instance, SoftMax function would classify the outputs of fed input, e.g. image.

### **3.3.4 Activation functions**

It is referred to as transfer function if the output range is infinite, whereas it is called as a squashing function in case the output range of activation function is limited, for example, the sigmoid function that maps the whole real numbers into . Henceforth, the activation function calculates the weighted sum of the layer in mapping it in a space vector using one of the functions: ReLU, LeakyReLU, Sigmoid, etc.

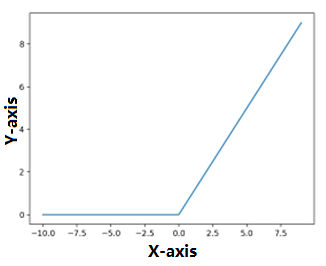
#### **3.3.4.1 ReLU**

The ReLU function computed as follows:

(1)

, where is the input, and Max operation takes the maximum out of (, input ). Therefore, if the input value is negative, the function would return . Otherwise, the function would return much positive answer. In the following Figure (15), the ReLU function is well represented.

**Figure 15**- ReLU function in graphical representation



#### **3.3.4.2 LeakyReLU**

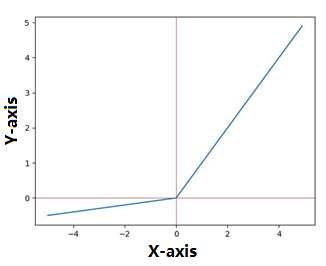
On the other hand, LeakyReLU performs a threshold value whenever the input is less than . Hence, the input is multiplied by alpha. The operation is summarized as follows:

, (2)

, (3)

The LeakyReLU noticeably used for the solving the problem of vanishing gradients as in ReLU, the negative input is set to all the time. In this case, a dead ReLU phenomena is reported since there is no learning occurred when the new weight remains as the old one. Therefore, the derivate is in backpropagation.

**Figure 16**- LeakyReLU function in graphical representation



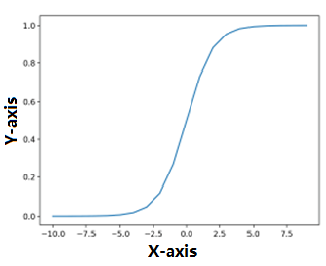
#### **3.3.4.3 Sigmoid**

The sigmoid activation, as called logistic function or S-shaped function, calculated as follows:

, (4)

Therefore, the function takes a real value input and outputs a value in range from to . With more positive input; the closer the output will be to , while with more negative value, the closer the output to .

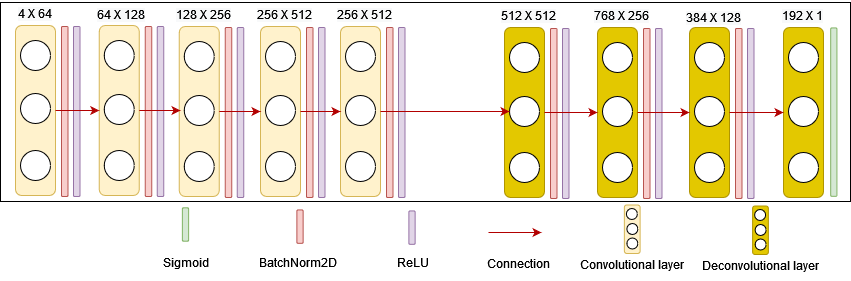
**Figure 17**- Sigmoid function in graphical representation



## **3.3 U-Net**

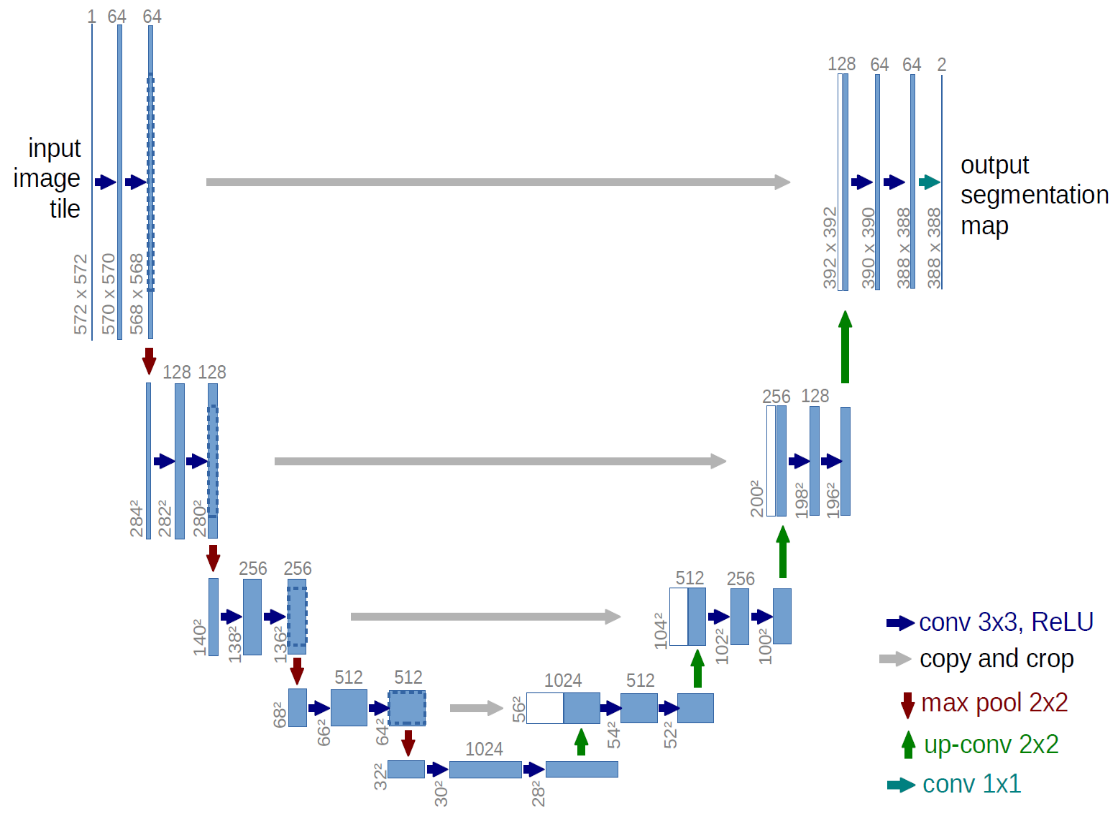
U-NET, an image segmentation, helps to reduce the volume of data. The U-Net is composed of two channels. The first channel resembles to the encoder as it captures the context of the image. On the other hand, the decoder is a transposed convolution that works to construct the original image based on image segments extracted in the process. Basically, the network assembles layers of convolution followed by max pooling later that reduce image density (reduces training parameters of network).

**Figure 18**- U-Net architecture



It was first introduced by [19] in 2015 and, as its name implies, is a U-shaped network consisting of both contracted and extended paths. The network has an image segmentation approach that goes beyond its competitors. The intuition behind the U-Net, on a downhill (contracted path), it learns the classification of objects. On an uphill (extended path), it learns the position of the object. In other words, the corresponding layer on the shortened path passes some information to the extended path. In this way, the classification context is transferred to the localization module, which makes the whole network much performant.

**Figure 19**- The architecture U-Net, which extracted from [19][20]



In depth, the architecture consists of four encryptions and another decryption blocks connected via a connection bridge. The encoder array (contracted path) halves the space size and doubles the number of filters (functional channels) at each encoder block. Likewise, the decoder array doubles the spatial size and halves the number of feature channels.

First, the encoder network acts as a feature extractor and learns the abstract representation of input image through a sequence of encoding blocks, all of which consists of two convolutions. Next, convolution layer is followed by a ReLU function, which helps in better generalizing the training data, and introduces non-linearity in the network. The output of the ReLU acts as a jumper connection for the corresponding decoder block.

Following the encoder, a max-pooling up to reduces the spatial dimensions (height and width) of the feature map. Also, batch normalization reduces the internal covariance changes and makes the network more stable during training.

Skip connections provide additional information for the decoder to generate better semantic features. They also act as a bypass connection, enable indirect gradation flow to the previous layers without degradation. Besides, it streams the flow of gradients during backpropagation, thus it helps better the network in the learning and representation process.

The network of encoders and decoders complete the information flow. It consists of two convolutions, where each convolution is followed by a ReLU activation function. Finally, the decoder network is used to derive the abstract representation and generate the semantic segmentation mask. The decoder block starts with a transpose convolution. Then, it is concatenated with the corresponding connection characteristic map of the data block. These leaping connections provide functionality from previous layers that are sometimes lost due to network depth. Then, two convolutions are used, where each convolution is followed by a ReLU activation function.

The final decoder output undergoes convolution with sigmoid activation. The sigmoid activation function provides a segmentation mask that represents the pixel-by-pixel classification.

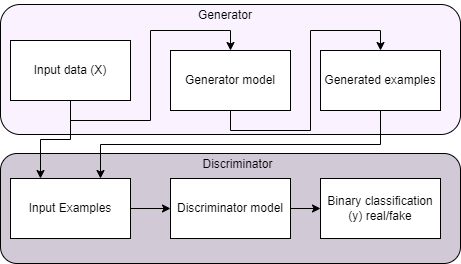
## **3.4 GANs**

By specifying a high-level target, in producing indistinguishable results from reality, and then automatically learning an appropriate loss function to satisfy that goal, is exactly what GANs do. The loss learning GANs tries to classify whether the output image is real or fake by training a generalized model to minimize this loss. Therefore, blurred images will not be accepted as they will obviously look fake. Because GANs learn an adaptive loss to data, they can be applied to a multitude of tasks that traditionally require different types of loss functions. The adversarial loss makes the key success of GANs, which basically force generated images indistinguishable from real ones.

As a definition, GANs are generative modeling approach using DL methods as in CNNs. In practice, GANs train a generative model by two sub-models: the generator model at one hand, learn to generate new examples by mapping the modalities of the networks’ inputs as on the other hand to feed it to discriminator model in classifying the examples as either real or fake.

As we could see in the following Figure (20), GAN generator model is considered as unsupervised learning problem as it generates a batch sample. In return, these entities with the real examples fed into the GANs model. Then, the discriminator model, as considered a supervised learning problem, got updated based on its performance of classification of the sample to either real or fake. The whole concept of the GANs is represented in the following Figure (20):

**Figure 20**- GANs concept



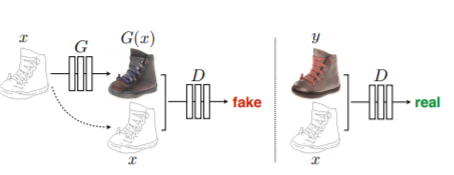
On the other hand, the unified GAN of single generator and discriminator tries to map images from four modalities. Briefly, the generator takes an image with its modality as input to synthesize it to a target modality. The discriminator is capable to differentiate between real and synthesized images on their corresponding modalities.

In mathematical terms, such that the distribution of images from is indistinguishable from the distribution using an adversarial loss mapping. GAN learns a mapping from the observed image and the random noise , to label through [13]. Generator is trained to produce outputs that are indistinguishable from “real” images by a highly trained discriminator . is trained to best generate fake-generated images and tries the best to distinguish the real from fake till they become alike. The learning process is highly under-constrained; we couple it with an inverse mapping and introduce a CC loss to enforce , and vice versa. To keep in mind, there are two different CC losses computations:

1. Forward CC loss:
2. Backward CC loss:

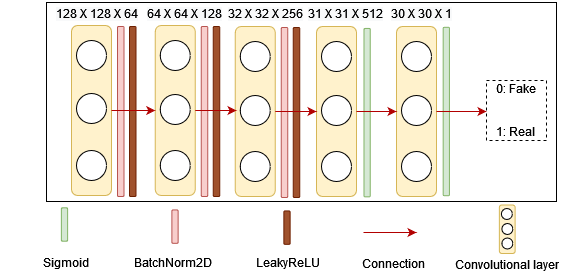
Thus, this training procedure is mathematized in Figure (21).

**Figure 21**- Demonstration of generator-discriminator mapping, which extracted from [13]



The following Figure (22) details the GAN architecture in follow up with the output of U-Net architecture, represented in Figure (18), will be the input of the present network.

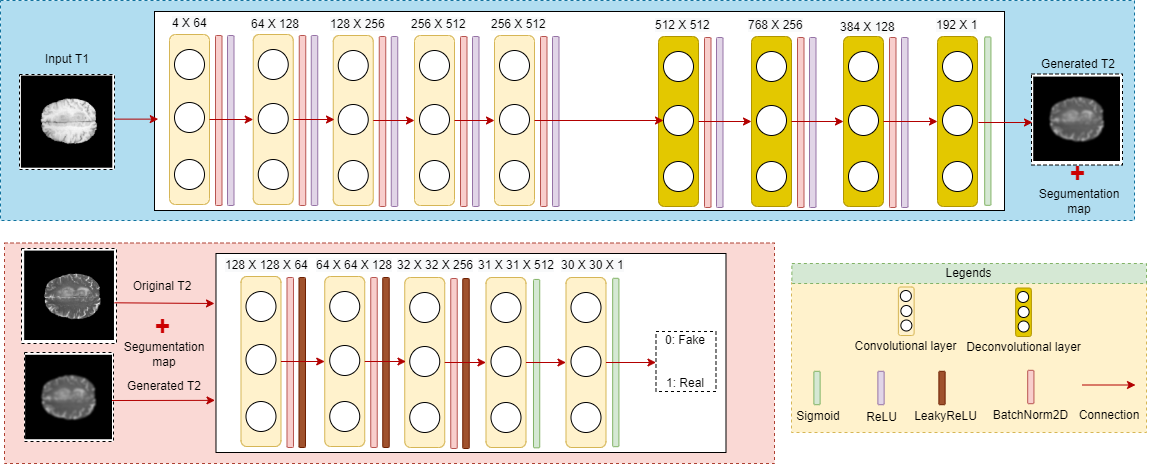
**Figure 22**- GAN architecture



## **3.5 U-Net and GAN**

After the separate representation of both architectures of generator U-Net and the discriminator GAN, the following Figure (23) represents the complete DL network in providing a mockery example. The generator takes as an input T1 to generator fake T2, which will be the input of GAN along with the real T2. At the end, the GAN would work to differentiate all of which is true or fake.

**Figure 23**- The assembly architecture of U-Net and GANs



## **3.5 Dataset**

BRATS2018 contains multimodal clinically-acquired -T MRI scans, available as NIfTI files (.nii.gz), are native T1, T1c, T2, and FLAIR volumes. They were obtained using different clinical protocols and different scanners from multiple institutions [5][6].

All images were manually segmented by to evaluators using the same annotation protocol. These annotations were finally approved by an experienced neuroradiologist. The data distributed after simultaneous registration using the same anatomical template, interpolation to the same resolution , and pretreatment of skull stripping.

The following Figure (24) represents four different modalities along with its segmentation map.

**Figure 24** – Data sample demonstrating the four modalities with segmentation map (Ground Truth)

Une image contenant texte, invertébré, mollusque, matériel

Description générée automatiquement

The dataset contains samples of modalities, distributed over patients each. The patients, reportedly, divided into sub-categories: HGG and LGG.

BRATS2018

The following Table (3) summarizes the dataset distribution.

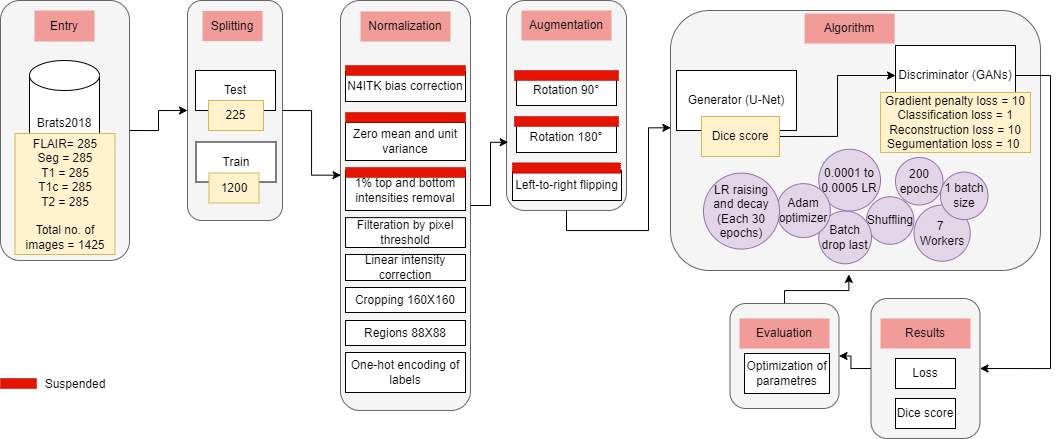
**Table 3**- BRATS2018 modality distribution over 285 patients

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

# **4. Approach**

This section specifies the different steps of the architecture by shading light on data input, split, normalization as well as augmentation techniques, generator-discriminator algorithm with the generated results, and finally evaluation and parameters optimization. The whole schema is presented in Figure (25).

**Figure 25**- Program architecture, specifying the different steps of data entry, pre-processing, augmentation, algorithm, results and evaluation metrics. Note: the process marked with red was suspended on the final architectures



For demonstration, the data is split into training and testing. Then the images are normalized by applying different techniques: bias correction, normalization, intensities removal, cropping and finally one-hot encoding. Later in the process, the normalized images are augmented by applying supplementary techniques: left-to-right flipping, and rotations.

The algorithm, therefore, consists of a generator that takes the input to generate fake modalities for the discriminator to classify into two classes either fake or real. The more the algorithm is unable to distinguish the difference between the reality and the illusion, the algorithm tends to achieve merely perfection.

The generator and the discriminator share certain parameters in common, as detailed in the following:

* ADAM optimizer with LR decay and raising at each epochs.
* computational workers.
* Batch drop last is enabled to ignore the last batch whenever the number of examples is not divisible by the batch size.
* Data shuffling.
* Epochs.

On the other hand, the generator and discriminator have different parameters. First, the generator has the following unique parameters:

* batch size.
* Dice score.

The discriminator distinguishes certain parameters, different from U-Net generator, detailed in the following:

* batch size.
* Gradient penalty loss.
* Classification loss.
* Reconstruction loss.
* Segmentation loss.

After the training process, the results are assessed in considering the loss as well as the dice score. At the end, we adopted parameter optimization based on the performance of each training process. Hence, an overall enhancement is evidently released in optimization process.

## **4.1 Pre-processing**

This section consists of different steps, including one-hot encoding of labels, top and bottom intensities removal, zero mean and unit variance, N4ITK bias correction, and as well as cropping. A result of the data pre-processing is represented in the following Figure (26).

**Figure 26** – The outcome of the pre-processing step, including bias-correction, normalization, filtration, and size cropping

Une image contenant invertébré, mollusque

Description générée automatiquement

Unfortunately, normalization and bias correction are suspended in the final prototypes as they have generated failure in the learning process. This part is suspect for future investigation. Besides, data augmentation is as well suspended from the final prototype as the base model has taken most of the time in experimentation, and it would be interesting to compare the base results to the results of augmented model in the future.

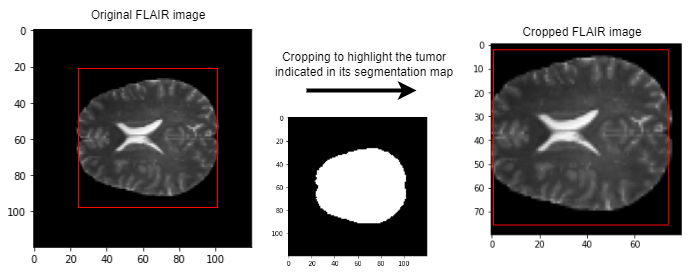
### **4.1.1 Image Cropping**

The entry image is cropped to for the prototypes, as indicted in the following:

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

Moreover, a cropping of  is adopted to highlight the tumor area in the modalities by the aid of segmentation map. The cropping is applied to the third prototype of MISO (T1, T2, and FLAIR → T1c), as it the aim of the research project to avoid the insertion of gadolinium.

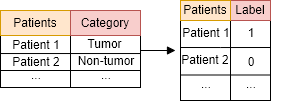
**Figure 27**- Image cropping from to , highlighting the zone of tumor



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

The labels of the images are transferred to one-hot encoding, which is basically putting whenever there is a tumor, otherwise, it is noted with zero value. The representation of encoding is represented in the following Table (4):

**Table 4**- One-hot encoding of tumor and non-tumor labels



### **4.1.3 Normalization**

In the normalization step, the algorithm removes the top and bottom of intensities and then normalize image with zero mean and unit variance. Therefore, feature standardization ensures that the mean of the values ​​of each feature in the data is zero (if the mean is subtracted by the numerator) and has a unit variance [18]. This method is often used to determine the distribution mean and standard deviation of each characteristic. Then subtract the average from each feature. Then divide the value of each feature (the mean has already been subtracted) by its standard deviation.

, (5)

where is the original feature vector, is the mean of feature vector, and is its standard deviation.

Furthermore, the non-background regions should be normalized since intensities range from to . The minimum in the normalized slice corresponds to intensity in unnormalized slice, which replaced with to keep track of intensities.

### **4.1.4 Filtration**

The image slices filtered out in fixing a threshold where the pixel number in brain area is lesser than .

### **4.1.5 Linear intensity correction**

The images slices were linearly scaled their original intensity to .

### **4.1.6 Bias correction**

The N4 bias field correction algorithm corrects the low frequency intensity inhomogeneities present in the MRI image data known as biased or enhanced fields. This method has been reported to have been successfully applied as a flat field correction for microscopy data. Furthermore, it starts with a simple parametric model and does not require tissue classification [17].

### **4.1.7 Data augmentation**

Data augmentation is adopted as the dataset is considered a small one for implementing DL and especially generative networks. Therefore, we adopted different augmentation techniques making use of defined functions and the help of TensorFlow through applying and rotations and left to right flipping of the original input image. For example, the following Figure (11) represents a sample of data augmentation that is been applied to “Brats18\_TCIA01\_390\_1\_78” slice.

**Figure 28**- Data augmentation, including: and rotations and left-to-right flipping

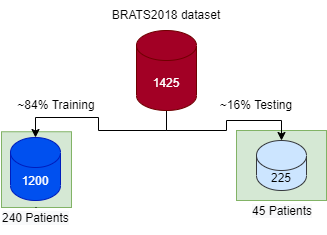
Une image contenant texte, mollusque, chaîne, matériel

Description générée automatiquement

### **4.1.8 Data split**

The training data are split into two sets of training and testing sets. For 3 prototypes indicated before in section [**4.1.1 Image Cropping**](#_4.1.1_Image_Cropping), firstly, the training set is of the training dataset and the testing set is . Therefore, the training set contains images for each generator and discriminator and the testing set includes the rest of images that are pictures in total. The splitting process is detailed and well documented in the following Figure (12):

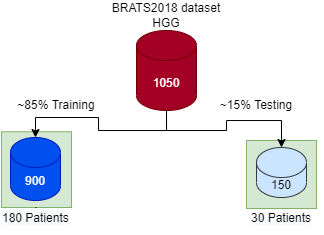
**Figure 29**- Data split into testing and training sets for SIMO (T1 → T2, FLAIR and T1c), MISO (T1 and T2 → FLAIR), and MISO (T1, T2, and FLAIR → T1c) prototypes with image entry size of



As we could see from Figure (29), the testing set contains samples divided by modalities: T, T, T1, FLAIR, and Seg. On the other hand, training samples are dedicated for training U-Net and GANs.

On the other hand, for the fourth prototype of regions cropping, the dataset is split for test set and training set on only HGG BRATS2018 subcatgory. The HGG category was selected as it contains only tumors of type .

Figure 30- Data split into testing and training sets for MISO (T1, T2, and FLAIR → T1c) prototype with image entry cropping of



## **4.2 Algorithm**

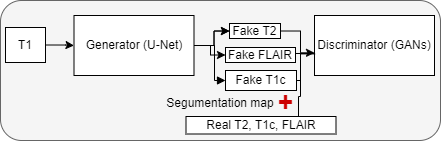
With the existence of many deep CNNs models achieving competitive results in terms of the state-of-the art, I chose U-Net and GANs to fine-tune the dataset in our hands for its efficiency and simplicity. The U-Net, as well described before, handles the segmentation mapping while the GANs algorithm plays the role to discriminate the generated fake images later in the training process.

In the following section, we would present the different algorithm methods that we develop to address the research gap in the literature by different methods, but we would narrow down the experimentation at the end on SIMO and MISO methods as we are aiming at predicting FLAIR and T1c.

### **4.2.1 SIMO**

SIMO architecture takes a single input, for example, in our case T2 to predict the rest of the modalities by passing by U-Net generator and GAN discriminator. The architecture is well presented in the following Figure (13):

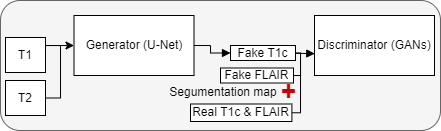
**Figure 31**- SIMO architecture takes T1 as input, & synthesis T2, FLAIR, T1c, and segmentation map for discrimination



### **4.2.2 MIMO**

MIMO architecture takes multiple input, for example, in our case T1 and T2 to predict the rest of the modalities by passing by U-Net generator and GAN discriminator. The architecture is well presented in the following Figure (14):

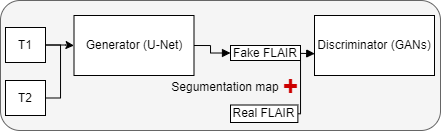
**Figure 32**- MIMO architecture takes T1 and T2 as input, and synthesis T1c and FLAIR for discrimination



### **4.2.3 MISO**

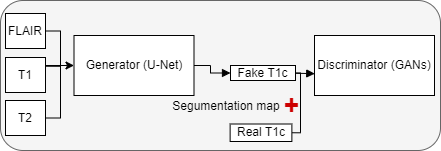
MISO architecture takes multiple input, for example, in our case T1 and T2 to predict FLAIR or T1c by passing by U-Net generator and GAN discriminator. The first architecture of T1 and T2 → FLAIR is well presented in the following Figure (31):

**Figure 33**- MISO takes T1 and T2 as input, and synthesis FLAIR for discrimination



On the other hand, the second is architecture of T1, T2, and FLAIR → T1c is well presented in the following Figure (32):

**Figure 34**- MISO takes T1 and T2, and FLAIR as input, and synthesis T1c for discrimination



## **4.3 Evaluation Metrics**

The generator U-Net, predicting segmentation map and down-sampling, evaluated by dice score. The GAN discriminator is evaluated by different loss weights, detailed in the following list:

* Gradient penalty
* Classification loss
* Reconstruction loss
* Segmentation loss

# **5. Experiments and Results**

The final experiment has been launched on the following computer specifications from the June for August :

**Table 5**- Laboratory computer specifications

|  |  |
| --- | --- |
| **Mark** | DELL R |
| **CPUs** | Intel Xeon Silver GHz ,M Cache |
| **GPUs** | Nvidia Tesla VS G |
| **RAM** | GB |
| **SWAP** | GB |
| **Storage** | TB |
| **Network card** | GB |

In the first phase, the generator assures better results as it generates both the fake image as well as its segmentation map. Later in the training of the GAN architecture, the results would remark significant improvement through adversarial loss.

**Table 6**- U-Net dice score of the 4 architectures, SIMO achieved ACC, MISO (output FLAIR) achieved , MISO (output T1c ) achieved , and MISO (output T1c ) achieved

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

**Figure 35**- U-Net dice score, a comparison between SIMO, MISO (output FLAIR), MISO (output T1c ), and MISO (output T1c ),

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

GANs

# **6. Conclusion**

This section is devoted for the research conclusion. It includes the two subsections of professional and personal conclusions.

## **6.1 Professional**

By the art DL, I was able to experiment different settings of architecture on dataset. Through experiments, I have also found that for this dataset, fine-tuning the whole model not only gives better result but also helps the model converges much faster than fine-tuning the rest of the layers.

## **6.2 Personal**

The internship opportunity I had with XLIM laboratory was a great chance to further my learning and professional careers in image processing (precisely, on medical imaging and bioengineering), ML, and DL. Thus, I do consider myself as lucky individual for the having the right position in data and ML engineering. I am so grateful for having the chance to practice many grateful resources of knowledge and skills of independency, research, and personal development. I am with great confidence I could say I am lucky to meeting different specialists in health sector as well as in computer science, including DL and ML. In general, they have led me of course to the right and desired direction through the internship period from March to September in the year .

## **6.3 Future Works**

We could consider experimenting different loss functions, hyper-parameter optimization, deeper generative networks, and data augmentation, too.

## **6.4 Challenges**

I have faced many difficulties in terms of setting up the Anaconda environment, which was totally not suitable for starting up the proposed CNNs algorithm. Also, the limited computational resources are the most difficult challenge because I could not upgrade my personal laptop to improve the whole results of the research project. The following table (7) specifies the specifications of the laptop that we have tested the architectures on.

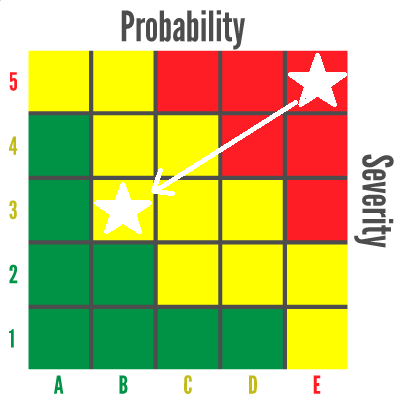
**6.5 Risk Analysis**

In this section, I will present the encountered risks during the whole period of the internship.

**6.5.1 The un-availability of dataset**

Due to the unavailability of the dataset of the hospital CHU Poitiers, we have decided to experiment the architectures of generative networks on an open-source dataset, which called BRATS2018. After taking into consideration the proposed plan, we have successfully reduced the gravity of the risk from high severity and probability to lower probability with low medium gravity.

**Figure 36**- Risk probability/severity matrix for the un-availability of dataset



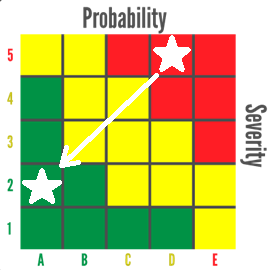
### **6.5.2 Administration user non-grant**

In XLIM laboratory, the administrative right is not given to an intern, for a temporary period. The problem is having no rights to install the applications by myself as well as managing programming environment seemed impossible. From that, I have taken the following solutions to demine the risk:

1. Using my personal computer to conduct the prototype test.
2. Installing applications with \*[[2]](#footnote-2).tar extensions to avoid the administration rights.
3. Acquiring two different machines at distance to finalize the final prototypes.
4. Working with python virtual environment as anaconda asks a lot for admin privileges.

After taking in account the above solutions, therefore, the risk diminished substantially as we could visualize in the following Figure (21):

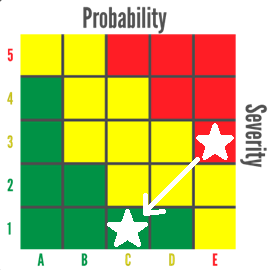
**Figure 37**- Risk probability/severity matrix for risk of administration user non-grant



### **6.5.3 The ambiguity of specification**

At the beginning of the internship, the specification demanded during the whole period of stage was a bit too broad and not narrow to a focus research area. Later, after a meeting with the educational tutor, he set me up to organize a whole planning map for the whole internship. For that, I have done GanttProject as well as PERT to overshadow the forthcoming tasks and the current tasks at hand. Henceforth, the gravity of the problem reduced with a lower probability.

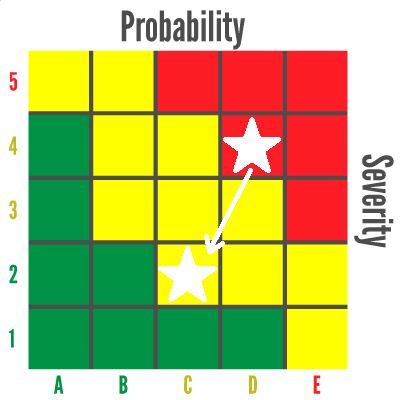
**Figure 38**- Risk probability/severity matrix for risk of specification ambiguity



### **6.5.3 Prolonged training of DL architectures**

For the reason of prolonged and training delay of the different DL architectures, we have proposed cropping the MRI slices. An cropping is implemented to reduce the gravity of the problem on MISO architecture, which takes as input: T1, T2, FLAIR, and outputs: T1c. It is only implemented on this architecture because it is the focus of the research to avoid inject gadolinium. The risk is well diminished as visualized in the following Figure (38).

**Figure 39**- Risk probability/severity matrix for risk of prolonged models’ trainings



# **7. References**

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.
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>

# **Annex**

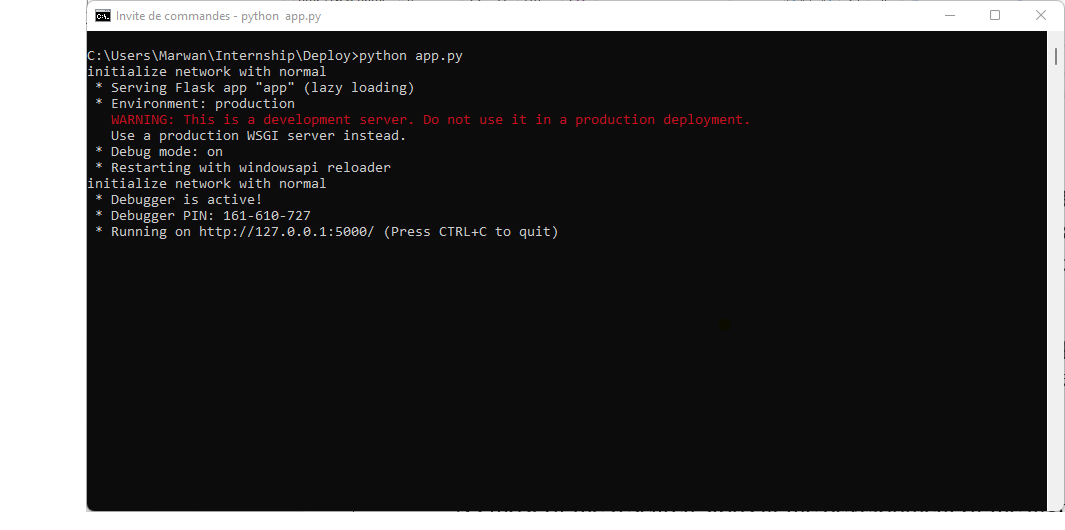
CelebA dataset contains face images of celebrities, each annotated with binary attributes. They crop the images, then resize them as . Randomly, they selected images as test set, and they used all remaining images for training data. They constructed seven domains using the following attributes: hair color (black, blond, brown), gender (male/female), and age (young/old).

RaFD dataset consists of images collected from participants, who made eight facial expressions in three different gaze directions and angles. The images are cropped to , centralized faces, and finally resized to .

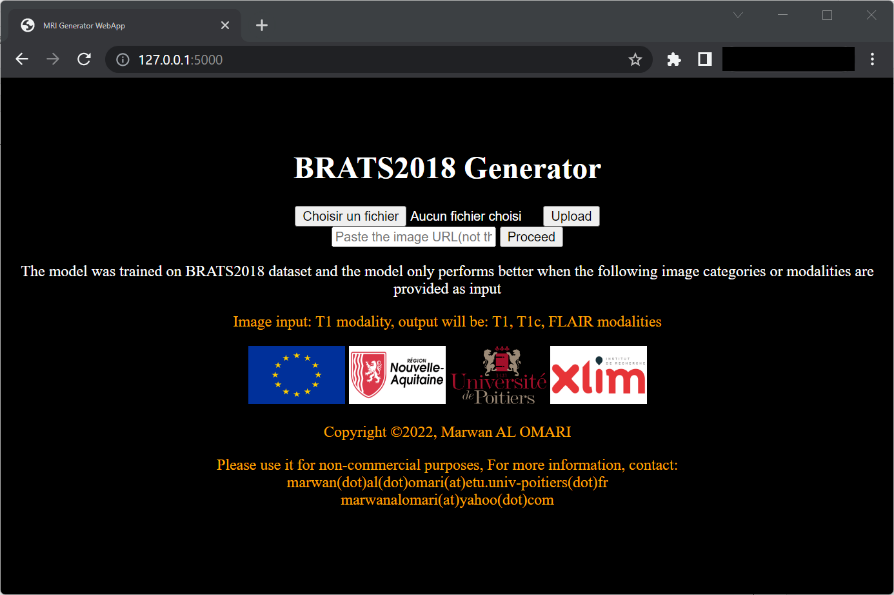
SIMO Model

As most of the research stops at the development of the algorithm, a prototype of SIMO model is deployed as a microservice, using the technology of flask and python.

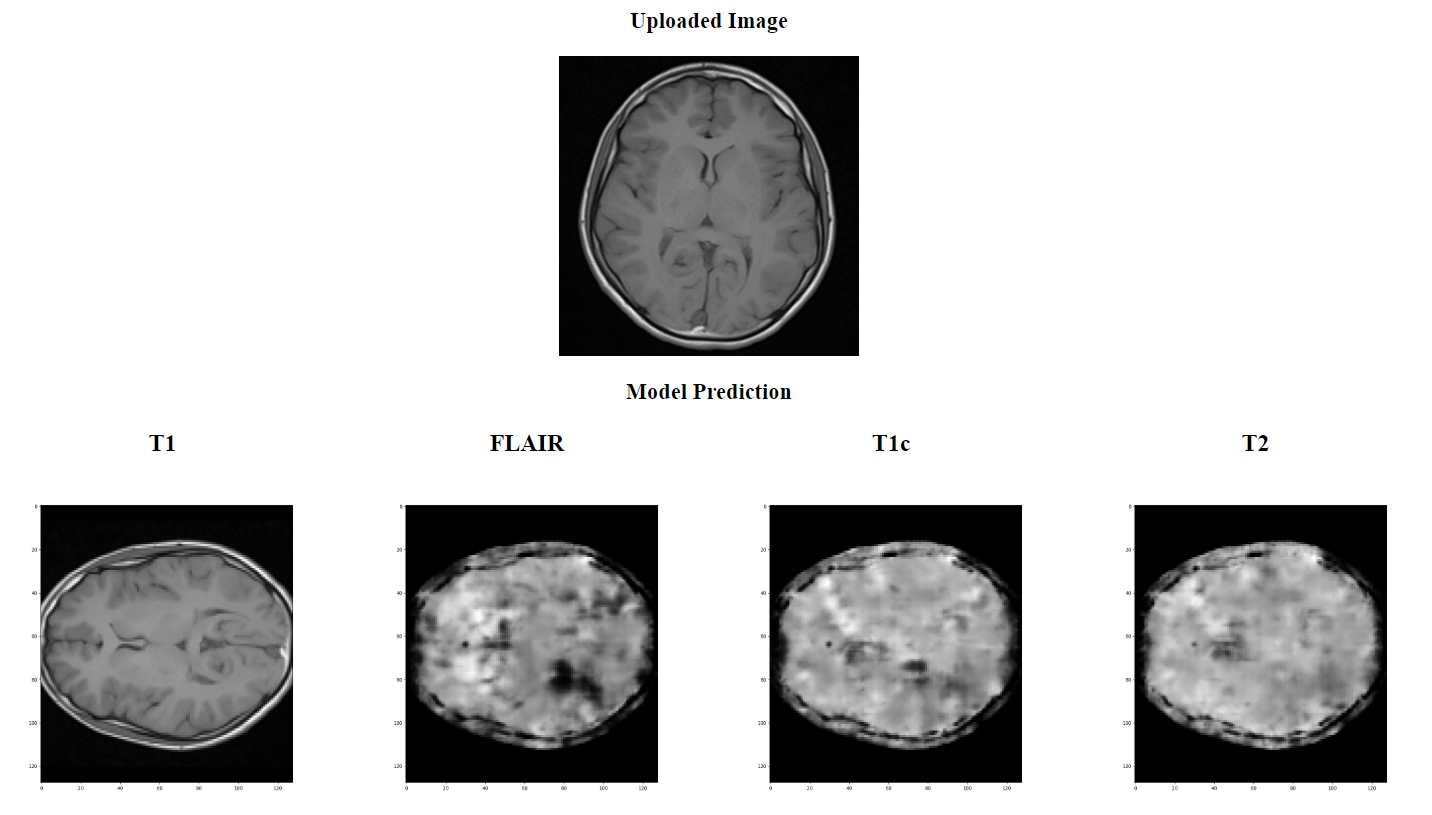
**Figure 40**- The model SIMO, as a microservice, accessible on local service for debugging 127.0.0.1:5000



**Figure 41**- The model SIMO, as a microservice, accessible on local service for debugging 127.0.0.1:5000



**Figure 42**- The model SIMO, as a microservice, accessible on local service for debugging 127.0.0.1:5000



1. www.zotero.org [↑](#footnote-ref-1)
2. The name of the wanted application. [↑](#footnote-ref-2)