



Degree in Computer engineering for Management and Information Systems

**GENERATIVE ADVERSARIAL NETWORKS AND SCORE – BASED MODELS
FOR MEDICAL IMAGING GENERATION**

Report

Kelly Zhen Zhou

Tutor: Xavi Font Aragones

2022 / 2023

Dedication

The project is dedicated to all those who are dedicating themselves to the research of
improvements and cures for diseases

Acknowledgement

Thanks to the professionals and researchers who have collaborated during the development
of this project.

Abstract

The following paper aims to be an approach to research the best generative models to generate carcinogenic cells given a dataset of computer tomography images. The final goal is to identify reliable features on the dataset for medical imaging generation, to be used as a support for the doctors' revision of cancer diagnoses. The results of this approach should not replace the radiologist's review of images. Two approaches of Generative Models are under study for this purpose.

Resum

El següent document té com a objectiu apropar-se a la recerca del millor model generatiu per generar cèl·lules cancerígenes a partir de un conjunt de dades de tomografies per ordinador. El objectiu final es identificar característiques fiables en el conjunt de dades per la generació de imatges mèdiques, per ser utilitzades com un suport a la revisió mèdica dels diagnòstics de càncer. Els resultats d'aquest enfoc no ha de reemplaçar en cap cas la revisió de un radiòleg de les imatges. Dos apropaments de models generatius son estudiants per aquesta finalitat.

Resumen

El siguiente documento tiene como objetivo acercarse a la investigación del mejor modelo generativo para generar células cancerígenas a partir de un conjunto de datos de tomografías computarizadas. El objetivo final es identificar características fiables en el conjunto de datos para la generación de imágenes médicas, para ser usadas como un soporte a la revisión médica de los diagnósticos de cáncer. El resultado de este enfoque no debe reemplazar en ningún caso la revisión de un radiólogo de las imágenes. Dos acercamientos de modelos generativos son estudiados para esta finalidad.

Table of contents

List of Figures.....	III
Glossary of terms.....	IV
1. Object of the project.....	7
2. Preliminary Study.....	8
2.1.Context.....	8
2.1.1. Theoretical framework.....	8
2.1.2. Functional study.....	9
2.1.3. Technical study.....	10
2.2. Background.....	10
2.3. Market and competition Analysis.....	11
3. Objectives and Coverage.....	12
3.1.Objectives.....	12
3.1.1. Client objectives.....	12
3.1.2. Product objectives.....	12
3.1.3. End user objectives.....	13
3.2.Coverage.....	13
4. Methodology.....	14
4.1.Research Methodology.....	14
4.2.Project Management.....	15
4.3.Resources and Data Management.....	16
5. Definition of functional and technological requirements.....	17
5.1.Functional requirements.....	17
5.2.Technical requirements.....	17
6. Development.....	19
6.1. Project set up.....	19
6.2. Data collection.....	20
6.3. Data preparation.....	21

6.4. First data images preview.....	22
6.5. Loading Data.....	22
6.6. Model Analysis.....	23
6.6.1. Style2GAN+ADA , the GAN approach.....	24
6.6.2. Denoising Diffusion (DD).....	28
6.6.3. Denoising Diffusion Probabilistic Model (DDPM++ Continuous.....	29
6.7. Model Training.....	30
6.7.1. Style2GAN+ADA.....	30
6.7.2. Denoising Diffusion Probabilistic Model	31
7. Conclusions.....	33
7.1. Further project development and improvement.....	36
8. Bibliography.....	37

List of Figures

Fig. 6.4.1. Plotting from .dcm CT to numpy arrays from subject - 0001

Fig. 6.7.1. Discriminative and Generative models comparison

Fig. 6.7.1.1 Basic Generative Adversarial Model Architecture

Fig. 6.7.1.2 Style2GAN+ADA Architecture

Fig. 6.7.2.1 U-Net architecture from Wikipedia commons , Mehrdad Yazdani

Fig. 7.1.1 Real sample images

Fig. 7.1.2 Euler-Maruyama Sampler results for DDPM++

Fig. 7.1.3 Predictor-Corrector Sampler results for DDPM++

Table. 7.1.4. DDPM++ results parameters comparison

Glossary of Terms

AI	Artificial Intelligence
AI Model	Algorithm that uses datasets to find patterns within the data
GM	Generative Models, models that generate data.
GAN	Generative Adversarial Network, generative model that works with a discriminator to generate data.
StyleGAN2	Generative adversarial model introduced by Nvidia in 2018
FID	Fréchet inception distance , a metric that indicates the quality of images created by generative models. The lower the value results in a better-quality image that preserves better correlation even under further distortion of images.
ADA	Adaptative discriminator augmentation, discriminator for StyleGAN2 in the context of this project.
EBMs	Energy Based Models, generative models
Denosing Diffusion	Synonym of Score-based generative model, an energy-based model

SCRUM	Agile framework for project management and development.
Diffusion model (DF)	Name to refer to models that gradually add Gaussian noise and reverse to generate images
Unsupervised Learning	Describe algorithms that learn , discover patterns and associate data without human intervention
MVP	Minimum Viable Product
MCMC	Markov Chain Monte Carlo
API	Application Programming Interface
LIDC	The Lung Image Database Consortium
LIDC-IDRI	The Lung Image Database Consortium and Image Database Resource Initiative
TCIA	The Cancer Imaging Archive
DICOM	Digital Imaging and Communication in Medicine
Gradients	Derivative of a function that as more than one input

Backpropagation Algorithm

Backward pass to adjust a models
parameter to minimize mean squared errors

1. Object of the project

The Project is highly focused on the research and applications of Generative Adversarial Network and Score – based models' accuracy to generate new data given a dataset.

The topic chosen to be researched belongs to different field studies related to Data Analytics, Machine and Deep Learning and Big Data. Its application for this project follows in the footsteps of similar recent research and workshops on the latest Neural Information Processing Systems convention.

The importance of this research relies on current needs in the CT (Computer Tomography) field for more accurate and clean tomography, in this case, will be focused on identifying and creating Machine Learning nodules of carcinogenic cells. After the generation of the cells, its accuracy to real carcinogenic cells CTs and its utility to improve its recognition will be measured. The final objective of the project includes identifying characteristics of real CT images and measuring how correct or common these traits are by generating new cells.

Computer Tomography improvement is a topic that leading technology and biomedical journals and articles are talking about and is under intensive study and research. The Fréchet Inception distance is a metric that measures the quality of images created by generative models. The lower the value results in a better-quality image that preserves better correlation even under further distortion of images. Following recent research in Deep Generative Modelling [1], the results of Generative Adversarial Networks and Energy Based Models, showed the lowest FID (Fréchet Inception Distance) results which indicates that the quality of the images generated is more accurate. Therefore, the models to be applied to reach the end of the project will be the three with lowest FID and analyzed to the chosen dataset.

2. Preliminary study

The following chapter will elaborate on the studied context for the development of the project.

2.1. Context

2.1.1. Theoretical framework

There have been previous studies on the current topic that may have or not direct application to the same topic. However, many factors can condition the results of these different investigations, such as the initial dataset, the way the data is analyzed and cleaned, and the different models used.

Given the variety of existing models both research on *Deep generative models: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models* [1] and *A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications* [2] get to the conclusion that the GM (Generative Model) approach for generation are more parameter efficient and have already been used in medical field for generating DNA[3] and medical image processing.

Going into the GAN (Generative Adversarial Network) approach, many papers point StyleGAN2 with ADA (Adaptive Discriminator Augmentation) discriminator as the best approach to medical image generation, already proven on Chest X-ray image generation[4][5]. This information matches with Deep generative modeling research pointing that the lowest FID in GAN models is hold by StyleGAN2 with ADA (2.42) [6]

Because there is a difference between the information in this document and other data found on the generation of X-ray images, for the project, only StyleGAN2 will be used , because we found two better options in the energy-based models.

Under the Energy-based models the two with lowest FID are Denoising Diffusion which has a slightly higher FID (3.17) compared to StyleGAN2. The lowest FID is for DDPM++ Continuous (2.40), this information does not match with StyleGAN2 with ADA in DDPM papers, here, the GAN model scores a 2.92 FID. However, even in DDPM++ Continuous paper, it is stated that is the model with lowest FID (2.41).

In addition to the previously mentioned GM approaches , early neural generative models , that were energy-based , required Markov Chain Monte Carlo (MCMC) sampling during training and inference , because the models struggled in scaling high dimensional data as images were (Bond-Taylor et al., 2022). Markov Chain based algorithms are now being used in the most popular and powerful models, in the last months and during this project development , the field has been moving towards Diffusion models as released in August 2022, Stable Diffusion , OpenAI's DALL-E or Midjourney. Therefore, during the development of this project the analysis of Diffusion models will be taken into account.

2.1.2. Functional study

The final goal of the project should be for the algorithm created during the research, to be able to display sets of generated computer tomography images containing nodules of carcinogenic cells.

Document should also contain for the dataset, a previous data science study to understand and compare the ending results to the original provided data. This approach is for three different models that belong to two different types of Generative Models, there should also be conclusions of the generated results.

To evaluate how much of the goal is achieved, is considered the time spent in training, the dataset, final FID and visually identifiable features should be considered as part of the conclusions.

2.1.3. Technical study

The project does not need a specific architecture being a research, however, if the study is aimed to be used, the implementation of the previously mentioned GM needs to be considered. There are papers that refer to the inner architecture of StyleGAN2[7] and Denoising Diffusion models.[8]

For the project, in the case it is used in the development of, for example, an API that generates medical images for each of the trained models, a complex architecture will not be necessary. Creating an API (Application Programming Interface) that calls the already trained models is the most opportune to be integrated into other existing architectures. As an alternative a Docker Container for the models may be also an option. The technical requirements can be found in section 5.2 of this document.

In addition, during the project development it is found that the dataset that will be used has been previously studied in *The Lung Image Database Consortium (LIDC) Data Collection Process for Nodule Detection and Annotation* [10] and images may be differentiated , specific classification of the images will be specified in the development chapter. An added possibility for the project will allow one to choose which type of nodule will be generated.

2.2. Background

The context of the project starts from the need for improving the way of detecting nodules , groups of carcinogenic cells , from computer tomography images. Position Emission Tomography, Computed Tomography, and the combination of both are the current best ways to approach the detection of cancer, being the second one the best option. CT scan is used to take images of the inside of a body and from those images the doctor will identify unusual patterns.

The recognition of these patterns should be done by the different models as a doctor would do and after the recognition, the model should be able to generate new images accurately to the ones that are inputted.

2.3. Market and competition analysis

This project aims to be a research and not a product, there are no products directly related to the end goal of this research, but there are similar research that can be used to reach the same end goal. Similar research includes the use of GAN for medical image generation, but do not test for StyleGAN2 with ADA and results may vary from the use of different starting datasets and manipulation of the data during the analysis and cleaning process. Other research includes a comparison of GANs such as Pix2Pix GAN and CycleGAN . [9]

The market is open to any company that would like to integrate in its system a product that generates medical imaging or apply the main features extracted from computer tomography during the development of the project.

What makes the difference with this project is not only the use of different parameters for the investigation, but also the fact that it can follow the latest trend on generative models.

3. Objectives and coverage

The following chapter will elaborate on the objectives to be accomplished during the project development and the coverage for each defined.

3.1. Objectives

3.1.1 Client objectives

This project may be open to a wide variety of clients belonging to the health area and specifically radiology. The tool or API that can be created from this research can become a support for professionals in the area. The sample client for a product created from this research may be individuals , groups or corporations. The clients for a product want :

- To integrate the models into a software under development, or develop a software around the research.
- To continue the training of the models to expand to other types of carcinogenic cells different from the used dataset.
- To allow the models to get new input data for continuous training.

3.1.2 Product objectives

The main objectives of the project are:

- To generate Computer Tomography images of carcinogenic cells with two approaches: GAN and Score – based models. For this purpose, there are three models that will be used: StyleGAN2 + ADA, Denoising Diffusion model and DDPM++ Continuous
- To analyze the results of different models and find the most accurate results compared to the provided dataset.

3.1.3 End user objectives

The end user of a product created under this research should always be a specialist in radiology. There is no use for the product or its results for users that do not understand the output images.

- To generate medical images of a specific type of carcinogenic cells : the approach of the research is limited to the chosen dataset, but following the steps of the research further datasets may be prepared to replicate them.
- To choose the quantity of generated images.
- To choose between the recognized patterns that the research results will bring.
- To contribute new data to the trained models.

3.2. Coverage

The previous steps will be done with current existent software, existing models and image segmentation techniques will be used. Steps that the project will not cover and define the scope:

- To search and develop in depth topics related to 4D image segmentation and image processing.
- To create new models.
- To test more models than the ones stated in this document.
- To train more models than the ones stated in this document.
- To create an API or end user approachable design.

4. Methodology

Research strategy is explained in this paragraph. The following resources and approaches were decided to be used for the project, resources and data management.

4.1. Research methodology

The main source of information for the research in the project is online, because the topics that are being addressed in this investigation are very recent, are evolving and are a subject in focus, it has been decided to discriminate the information so that files with dates after 2019 will be discarded. With the only exception, that they are papers that have been mentioned multiple times by several more papers that pass the date condition.

The main search engine for the project has been Google Scholar, but it has also been made use of the free trial of the Connected Papers page, which allows to find for a paper all those who have mentioned it or have a connection by date and number of connected nodes. In this way it has been possible to find the most relevant bibliography for the investigation.

Following these research methodologies , key names of experts in the field and attending conferences should also be mentioned as part of the research method being used.

Interviews with radiologists have also been used, which has so far helped to find a network of public radiology datasets.

This is the workflow that will follow when information is found.

1. Finding information: Search engines, books, papers and interviews are the main source of information.
2. Information is understood, personal annotations of the information for each should be written including conclusions, this should go inside of the Research folder to its respective type of information.

3. During the information critique the content is analyzed based on what is known from before, therefore every analysis should contain the dates in which it was done, in case there are further reconsiderations and conclusions about it.

4.2. Project Management

The free software from Atlassian Trello was created as a personal tracker of all the needed steps to accomplish. In comparison with other management tools like Jira, it is free, which would reduce the project costs and there is no need for the size of a Jira-managed project as a team of no more than three experts is needed.

The Set up for this project management is based on the Agile Scrum Board. This modality was compared to the Kanban Board and has been chosen because of the needs of time management that are explained in the headline 6.1 . The columns for the management of requirements are the following :

- Backlog: Non-written stories (requirements) that will be needed further.
- To – Do: Stories already written and ready to be started.
- In Progress: Stories that are already in progress to be done.
- Ready for testing: Stories done that need confirmation.
- Done: Completed Stories

Any completed story that needs to be reopened will directly go back to In Progress and any completed story that needs to be reviewed in dept will go back to Ready for testing.

Considering the three important points written before there were three types of stories based on them. Each story name uses hyphens to indicate a subcategory of that point. (i.e.: Dataset – Research on how to read 4D CT cell images). The stories that are not part of any of the three points should also be covered, therefore, any need that does not belong to any of them has its own personal and unique title.

4.3. Resources and Data Management

This headline will explain in detail the management and treatment of found information and posterior analysis.

Storage: The storage is the part of the project that stores all the information being used or consulted. The storage used will be Google Drive, because of its easy synchronization between devices, download and user management access to the files. The storage folders structure follows as:

Storage main, contains:

- README: will contain the information about how to understand the Storage
- Documentation: Folder with all information during the research and development. This folder will include this document.
- Preprocessed_batch_data : Folder contains zip files of batch of data ready to upload and train
- Subset_example: Folder containing 500 example real images
- Trained models : Folder with the trained models ready to load in both PyTorch and Pickle format
- Utils : Folder with other utilities
- All notebooks are at the very first level of the folder and have a number indicating the order of execution and creation.

5. Definition of functional and technological requirements.

5.1. Functional requirements

The following paragraph will list the functional requirements of the research.

1. To display the contents of the research user-friendly and easily executable.
2. To use a dataset clear and easy to manipulate, too big datasets will increase the environmental impact as more heavy processing hardware and techniques are needed.
3. To provide an executable file that will install the required software to run the models.
4. To properly store the trained models without needing to be re-trained to show results.
5. To provide the reader of the research with different graphs and comparisons between input and output images.
6. To contribute to the research in an open environment for further investigation.

5.2 Technical requirements

The following paragraph will list the technical requirements of the research that should satisfy the functional requirements above.

1. To accomplish functional requirement one, by using Google Collab. Will ensure cloud execution and the use of Python Notebooks will allow it to be easily read also in other environments as Anaconda.
2. To use Pickle library to store trained models.
3. To use PyTorch library for model training
4. To use Google Drive to store the progress and files of the project.

5. To improve the training speed of the three models.
6. To improve FID score for the three models.
7. To use a dataset that fulfills legal requirements and conditions of use should allow the development of the project.

The following technology stack has been evaluated to accomplish the technical requirements. Find the evaluation in the Viability Analysis of the project , Technical Viability chapter.

- Google Colab
- Google Drive
- Python
- Python Notebooks (Jupyter Notebooks)
- PyTorch
- TCIA REST API
- NBIA Data Retriever
- dicom2jpg python library
- Pickle

6. Development

6.1. Project Set up

During the early development of the project more libraries and software was added to the technology stack . The set up, as stated, was done in Google Colab and a first environment test was conducted which included checking the correct import of needed libraries and software .

6.2. Data collection

Finding a data collection to fulfill the requirements took less time than the hours calculated in the initial Viability Study. The final decision was to take TCIA (The Cancer Image Archive) data collection LIDC (Lung Image Data Consortium) as the collection was recommended by the supervisor professor Xavi Font Aragonès . Before finding the specific data collection, the recommendation of TCIA collections was already provided by radiologist Dr. Bernabé Meza Araiza .

With both recommendations over the same page and the found previous study over the specific data collection [10] , the content and origins of the dataset were studied in order to fulfill both the needs of the project and the viability analysis requirements. LIDC data collection contains CT information from a total of 1010 subjects in .dcm (DICOM, Digital Imaging and Communication in Medicine) format. This data collection, as stated in the creation papers [10] , was gathered and evaluated by experts from separate radiology centers.

In the TCIA page, the data collection will redirect to its current total collection named LIDC-IDRI as it included Image Database Resource Initiative. The main page states that the license for this dataset is under Attribution 3.0 Unported (CC BY 3.0) which allows sharing and adapting.

6.3. Data preparation

Once the data collection is decided , a key point in the project is to prepare the data. The process of retrieving the data from the TCIA needed the software NBIA Data Retriever to be downloaded and after choosing the subjects , a manifest document is downloaded from the website and this document is the one that allows NBIA Data Retriever to download locally the selected subjects files.

During the selection process of the subjects, when the 200th subject was selected the website displayed the weight of the download would be about 22GB . Given that the files had to be downloaded locally and then uploaded to Google Drive , there was a limit on the quantity the local machine could store at once (33 GB) . In addition to that, the data collection had to be processed to an easier to manipulate dataset and mapped to the labels.

To overcome the storage issue, a python script was created. Because there are 1010 subjects and the first batch of 200 subjects had a weight of 22 GB , the solution raised is to download , upload and then process the data in batches of 200 subjects and a final batch of 10 subjects. This solution is based on the remaining time of the project , because it is important to deliver results in a limited amount of time the first 200 subjects totally processed accomplishes the MVP (Minimum Viable Product) and the project objectives while allowing the research to be scalable and study the behavior of the results during the scaling process of the dataset.

6.4. First Data Images Preview

As mentioned in the previous section , the Python pipeline to process the batches of data up to 200 subjects was created and tested roughly over the first images to ensure the pipeline was properly working.

Another issue during the data preparation was the format of the data. Note that in radiology and medical imaging the .dcm (DICOM , Digital Imaging and Communications in Medicine) format is commonly used . The format also requires specific software to be viewed but for the aim of this project, what we need to study the data is having images that can be processed.

For that purpose, the library `dicom2jpg` is found, usage allowed under MIT license. The library was easily imported in Python and presented the first images, plotted with `matplotlib`.

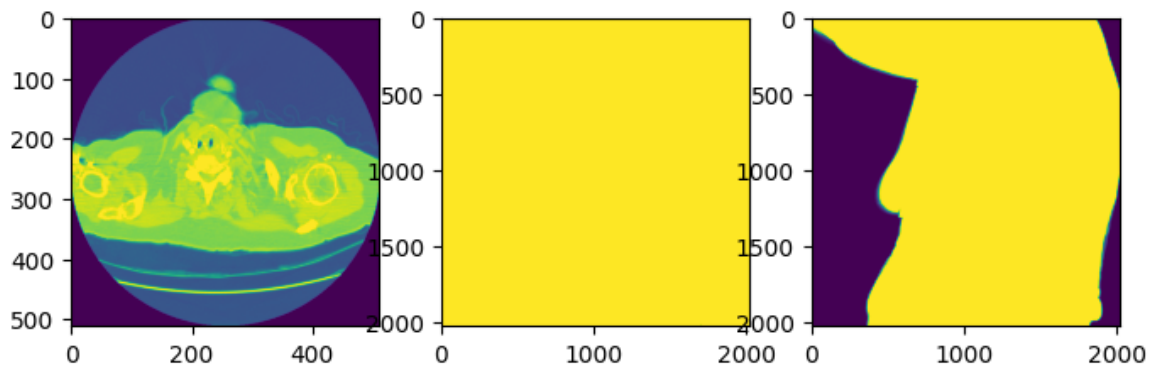


Fig. 6.4.1. Plotting from .dcm CT to numpy arrays from subject - 0001 from *TCIA LIDC-IDRI collection* [12]

At the moment of the generation the images are also stored in another folder with a .jpg version.

6.5. Loading data

For the models to be trained the images should be previously labeled. The process that is usually made with human validation is already done for LIDC - IDRI dataset, as stated in previous chapters for each subject there is a consensus of the evaluation of the nodule.

Following what is mentioned in LIDC papers [10] there are three main classifications:

1. Nodules equal or bigger than 3 mm in diameter
2. Nodules under 3 mm in diameter
3. Non-nodules equal or bigger than 33 mm in diameter

A new issue is raised around the data structure , the collection is divided into folders for each subject, and inside of each subject's folder there are more folders with separated .dcm files and one .xml file with the information about the subject's data status. Data and labels were scattered over different folders.

To overcome this problem it is found that TCIA has a Rest API , (TCIA REST API) that provides support for data projects and allows to call data collection information (all .xml files) in a single list. The processing of the retrieved information however, does not contain the specific information about the nodules.

Second attempt started by run through the downloaded manifest files. The official TCIA LIDC-IDRI page provides a detailed description of the .xml files for each subject and four sessions of reading by different radiologists. To get that information is processed the xml files related to the subjects number and session . This way there are four diagnoses assigned to each nodule and the final label decision about a nodule is made based on the most repeated result between the radiologists.

More research time was taken to find a proper version of the dataset which led to two Kaggle pages of previous works on the same dataset :

- The LIDC-IDRI data preparation by Wei Zhang , made originally for nodule classification.
- The LIDC-IDRI data loading and preparation by Andre Ye, which allows to show over the nodule images the different radiologist annotations.

6.6. Model Analysis

After collecting the necessary data in the following chapter there will be the study of the three different models that are going to be used. As stated in previous chapters the models used are the ones that previous researches with different datasets demonstrated lower FID values, a GAN model (Style2GAN + ADA) and two score - based models (DDM and DDPM++ Continuous) . Note that the difference between DDM (Denoising Diffusion Model) and DDPM++ Continuous (Denoising Diffusion Probabilistic Model) relies on the last one being an extension of DDM.

The first step is to understand what a generative model is. The word generative is assigned to a class of statistical models, in contrast to discriminative models :

- Generative models can generate new data. Generative models will get $P(X, Y)$ for a labeled dataset or $P(X)$ for a non labeled dataset. A generative model will compute the probability of X
- Discriminative models , discriminate types of instances in the data. Discriminative Models will get the conditional probability $P(Y | X)$. A discriminative model will compute the probability of X given Y

To understand how data is distributed , a discriminative model will separate data into instances X or Y and a generative model will group data into similarities.

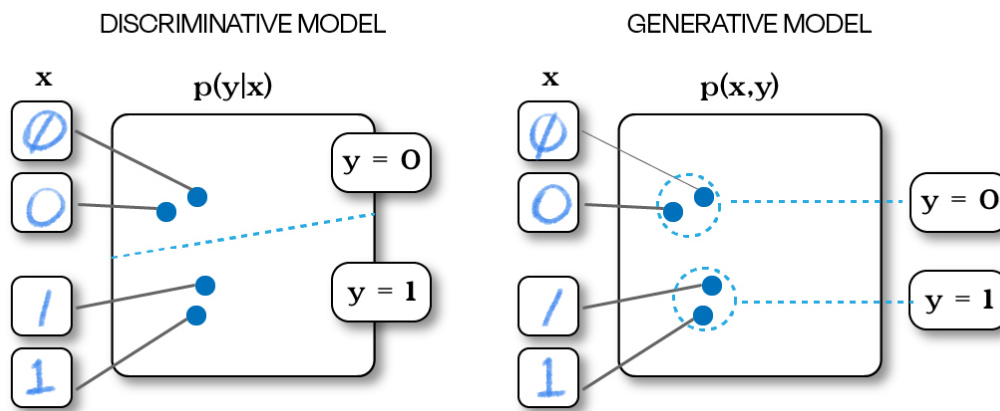


Fig. 6.7.1. Discriminative and Generative models comparison

6.6.1. Style2GAN + ADA , the GAN approach

Style2GAN is a generative adversarial model that follows previous versions of StyleGAN architecture. Corrects previous StyleGAN water droplet artifact error that may randomly appear in the generated images [15] .

Imported model follows Training Generative Adversarial Networks with Limited Data [16] by NVIDIA , the model is already fitted to train a limited and reduced amount of data , described in the papers as some thousands, similar to the dataset used in this project. To understand the architecture of this GAN the following steps must be understood :

- Basic GAN Architecture:

The first step is to understand the GAN architecture. There are two main neuronal networks :

1. The Generator : Creates generated fake images. The input for this neural network is random noise.

2. Discriminator : Evaluates the generated images. In this neural network there are two inputs : The fake image generated by the Generator, and a real image from the sample. The discriminator will penalize the Generator when noticeable fake results are detected.

The workflow starts with a random input of noise that will evolve during the training as the discriminator realizes what is fake, in the end , what is expected is the discriminator to be worse at differentiating between the fake and the real images. Using the backpropagation algorithm the classification that the discriminator does goes back to the Generator to update its weights , the parameters and connections between the units of a neural network.

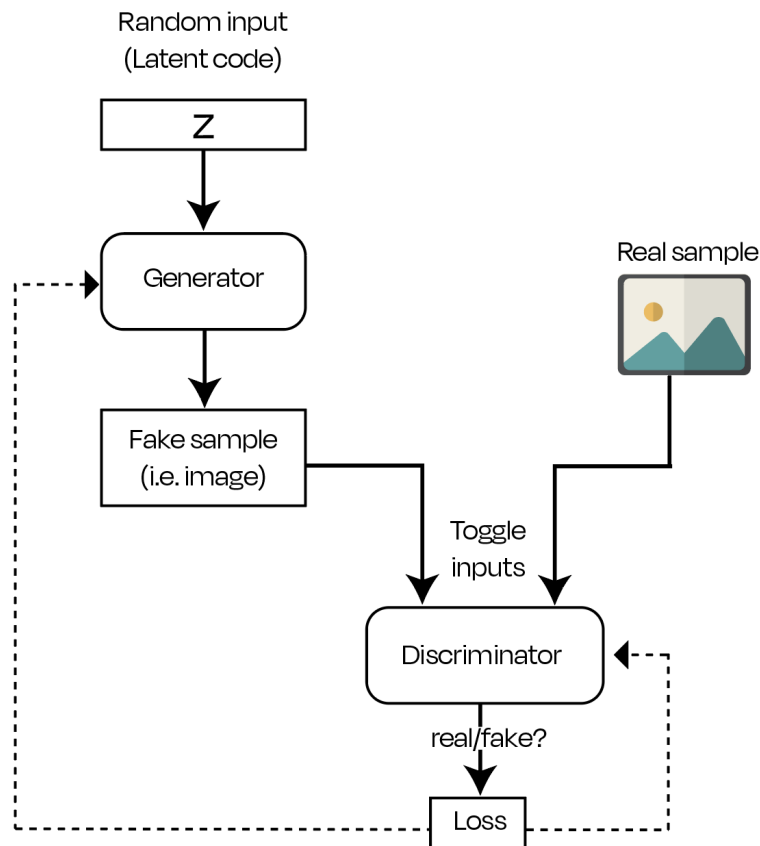


Fig. 6.7.1.1 Basic Generative Adversarial Model Architecture

Going in depth on the Backpropagation loss , the Discriminator is connected to two loss functions that may differ from one neural network to the other.

- Discriminator loss : During the Discriminator training it ignores the Generator loss and will only use Discriminator loss. The function penalizes Discriminator by wrongly classifying a real input as false or vice versa. After the penalization, Discriminator updates its weights to reduce the mean squared error (backpropagation)
- Generator loss : When the Discriminator correctly classifies the images the loss is for the Generator. In this case the Generator gets penalized and its weights are updated to reduce the mean squared error. The loss is propagated from the discriminator to the generator to get gradients and the gradients are used to update the weights.
- Style2GAN Architecture

To understand the architecture of Style2GAN it is necessary to elaborate on how the generator and discriminator layers work.

The Generator

A random latent input enters the generator and the augmentation starts : StyleGAN architecture allows a better control of the layers (stylized) of the neuronal network what consequently also allows control over the expected output. The layers allow the StyleGAN architecture to have a progressive training (augmentation) , the model trains initially with low resolution images and the resolution increases to generate high quality images with unsampling.

Between the convolutions before sampling, there is the styling module : noise for generating specific stochastic details (i.e: hair, facial details) and the AdaIn (Adaptive Instance Normalization) module applied.

The AdaIN operation in Style2GAN+ADA removes the Normalization to solve the droplet artifact problem. To understand how this is applied in the following figure, the AdaIN operation is separated between the normalization mean and the modified mean. It is important to note that Style2GAN papers remove the mean in the normalization as it is not needed.

- Style2GAN + ADA : Adaptive Discriminator Augmentation.

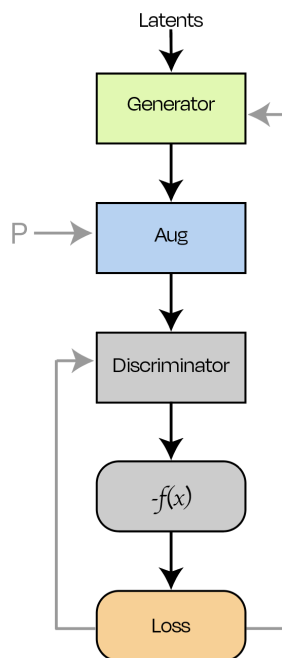


Fig. 6.7.1.2

Style2GAN+ADA

Architecture

The adaptive discriminator augmentation is introduced to train GAN with limited data, using stochastic discrimination augmentation the augmentation probability controls the strength of the augmentations.

The augmentation is applied after the image exits the Generator but before entering the Discriminator. Style2GAN papers [16] point as most effective augmentations : pixel blitting (horizontal flips, 90 degree rotations and integer translations) , geometric augmentation and color transforms. The higher the strength the less clean images resulted, hence , finding a reasonable value is needed. Papers state, that $p = 0.8$ may lead to too messy images .

Some of the drawbacks related to GAN:

- Vanishing gradient problem : Because GAN uses backpropagation algorithm, if the discriminator training goes too good then the generator training will fail if it learns

nothing. The gradient will be vanishingly small, preventing the weight from updating.

- If Generator and Discriminator are not balanced properly may overfit : data too adjusted and similar to the training dataset , hence generating already existing images.
- GAN are very sensitive to hyperparameters, and small changes may have a high impact on performance.

6.6.2. Denoising Diffusion (DD)

The Denoising Diffusion models are widely being used in image generation , with recent examples such as GLIDE or DALL-E-2. Although both GAN and diffusion models use data and generate a new output the architecture is very different, researchers tried to find a different generative approach that produces better results due to GAN problems listed in the previous chapter.

The generation process of the diffusion models consists of two main steps: a Forward Diffusion and a Backward Diffusion propagation process. The goal of a diffusion denoising model is to learn from the reverse (backward) process :

- Forward Diffusion : The input is an image that will gradually be covered in noise . At the end of the forward process the image should become unrecognizable.

The forward diffusion takes concepts from the Markov Chain, the process is applied to N successive steps and the prediction of probability density at a time (t) is dependent on time $(t - 1)$

- Backward Diffusion : After the forward process is finished , the backward process, the idea is to reverse the forward process and the image will be denoised. At each layer of the backward process the model predicts the parameters that were part of the forward process. In this process the start input is a noisy image that will return to an image that may have a similar data distribution of the ones on the dataset.

The approach is considered to be more stable than other generative models like GAN or variational autoencoders. The network architecture of the denoising diffusion models is usually U-net architecture, because it is needed that the input and output dimensionality are identical.

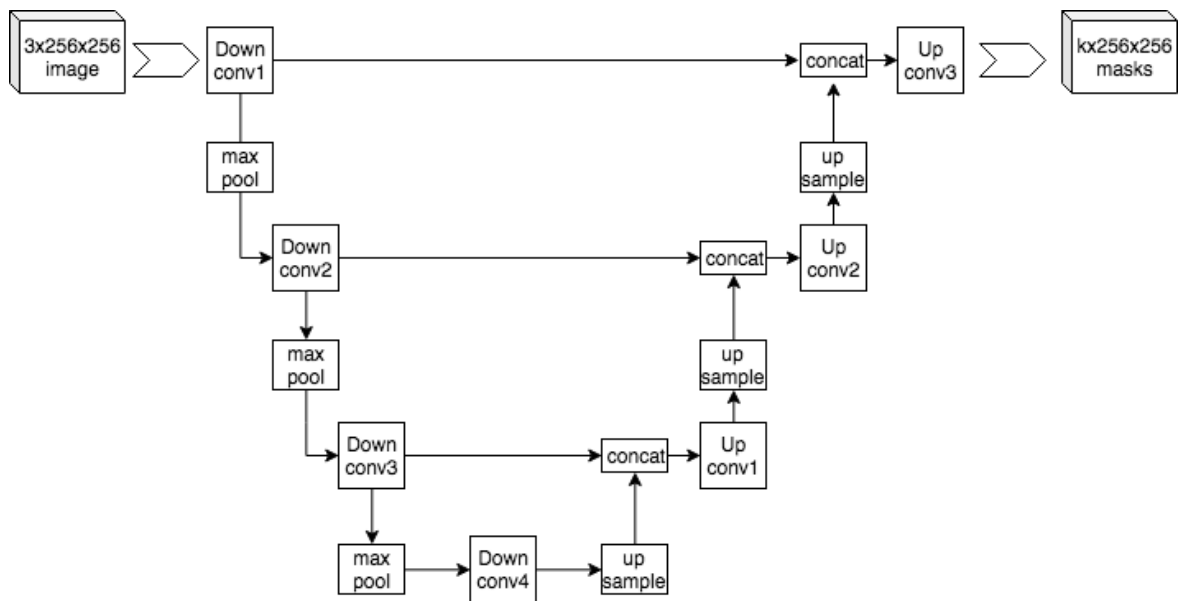


Fig. 6.7.2.1 U-Net architecture from Wikipedia commons , Mehrdad Yazdani

6.6.3. Denoising Diffusion Probabilistic Model (DDPM++ Continuous)

Many variants of the denoising diffusion models are being released as the approach gains more popularity in recent years. The Denoising Diffusion Probabilistic Model are a specific type of denoising diffusion model that differ in the following:

1. Noise formulation : DD takes a discrete distribution of noise while DDPM++ formulates noise based on continuous stochastic equations. DD will shape as a sequence of reversible transformations while DDPM++ uses noise as a distribution of continuous probability.
2. Images : DD images are represented as static pixels within a fixed range. DDPM++ images are represented continuously, which allows a more accurate representation of both noise and the image.

3. Probability Density Estimation: In DD, the probability density of the noise is estimated and used to denoise. In DDPC, the density of probability is estimated and relates a noisy image to its clean version.
4. Flexibility : Because the representation on DDPM++ images is continuous it has more flexibility to get larger. distributions of probability
5. Complexity : As the DDPM++ model is continuous its computational complexity is higher compared to DD with the addition of the computation of density conditional probability.

6.7. Model Training

The chapter will cover the training process and evaluation of each model.

6.7.1. Style2GAN + ADA

Model presented incompatibilities with the proposed environment , NVLabs original repository requires PyTorch 1.7.1 as the `conv2d_gradfix` function changed between versions. The PyTorch version in the environment (Google Colab) goes over PyTorch 1.10.0

Consequently, to install PyTorch 1.7.1 , Python needs to be downgraded to Python 3.7. This caused another issue during the development , as Google Colab retired support for Python 3.7 which was manually installed.

The installation allowed to add the correct PyTorch version, however the kernel of Google Colab failed and was not able to restart runtime to reflect these changes. As another solution it is found in a pull request in the NVidia GitHub repository which proposes a solution to the `conv2d_gradfix` issue.

Importing the head of the branch would solve the issue but raise another one : the sampling grid file would need a tuple type, which was added correctly , however because of the redefinition of the function would ask an output mask. The topic is still being discussed up

to the date of the project [17] , while the proposed solution may work for StyleGan3 is not the case for this model.

Because of the amount of environment issues, other environments were used :

- Miniconda : Installed over Google Colab to simulate an Anaconda environment , would generate issues regarding shell commands, accessing the local data from other code cells was not possible.
- Tensorflow : Because the issues were regarding PyTorch the other option studied in this paper was taken into account as there is also an NVidia implementation for TensorFlow of this model, similar to PyTorch would require a downgrade of Python to 3.7 and TensorFlow 1.x.x , it made TensorFlow version incompatible with Colab : the kernel fails to restart after downgrading Python.
- Anaconda : Setting up separately an Anaconda for this model was also one of the approaches, this one was quickly discarded because it relied on the local computer capabilities which would not satisfy the GPU requirements.
- Docker : a version of the Style2GAN + ADA for docker was found and pulled, the image installed and was able to run with no issues, however, similar to the problem with Anaconda it needed NVidia GPU, Docker must be run over the NVidia GPU docker, resulting in a need of more disk space which was not available for the local computer.

After all the approaches used above , to not take time from other models to train the GAN approach will remain for further development and amplification.

6.7.2. Denoising Diffusion Probabilistic Model

The implementation of this models follows last updates added during the development of the current project, implements code from the papers *Score-based Generative Modelling Through Stochastic Differential Equations*[18] modified to fit the current dataset.

Changes applied to the original model :

- Added a custom dataset function to prepare the medical images, note that the original code would use only one color channel for the generation of black and white images, however, because of the preprocessing performed in previous steps the channels in the images of the project dataset have 3 channels (RGB). The images need to be uploaded by mounting Drive in Colab. The files are unzipped to avoid Drive blocking because of the amount of calls.
- Changes on ScoreNet layers : because of the RGB format some tensors are updated to read 3 channels (self.conv1), these changes are applied in both forward and backward functions.
- Images are escalated to 128 x 128 , because it is the size used in the papers , other sizes may be used for further development.
- Updated dimensionality in Euler Maruyama and Predictor-Corrector samplers to fit the dimensionality of the use dataset. ODE sampler was discarded because will take dimensionality operations.
- Trained scaling amount of data and parameters. Training was done over datasets of 5 hundred , 5 thousands and 45 thousands with the following parameters
 - Epoch : 50 and 200
 - Batch size : 32 and 64

7. Conclusions

The conclusions are broken down up to the point the project has reached . The time invested in the training of the models should be increased, because the hours stated in the initial planning are below the ones spent .

Out of the three selected models the models that generated results is Denoising Diffusion Probabilistic Model Continuous , because the Denoising Diffusion training is still on going the conclusions will be analyzed over DDPM++ results.

There are two samplers provided in the original model code, having the ODE sampler removed because the dimension variations would not fit .

Euler-Maruyama sampling presented the most noisy results , all the results by this sampler have a solid noise image compared to the Predictor-Corrector sampler.

Real Sample

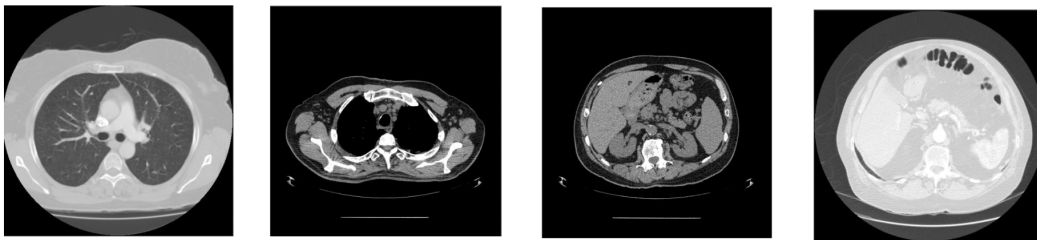


Fig. 7.1.1 Real sample images

Euler-Maruyama Sampler

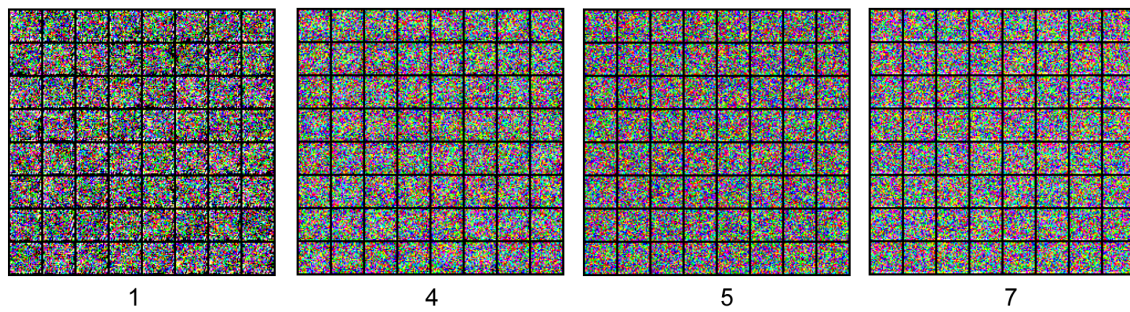


Fig. 7.1.2 Euler-Maruyama Sampler results for DDPM++

Predictor-Corrector Sampler

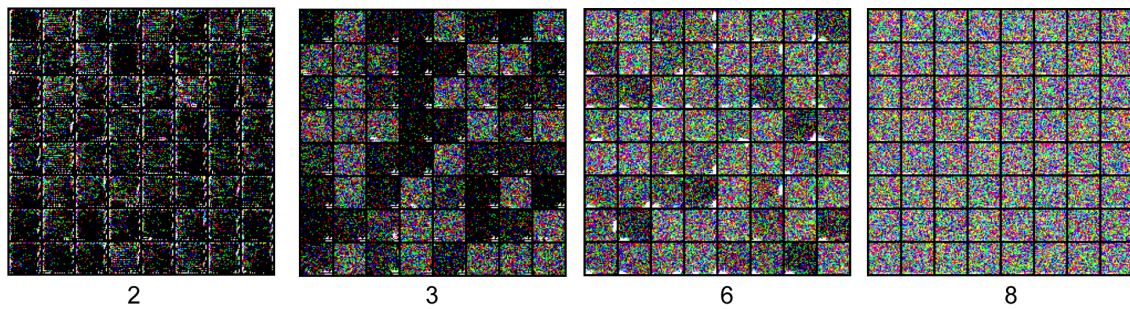


Fig. 7.1.3 Predictor-Corrector Sampler results for DDPM++

In the figure above the results generated compared to real dataset images, the following table will compare the executions:

N.	Sampler	Training Time	Average Loss	Dataset size (images)	Epochs	Batch size
1	Euler-Maruyama	5 min	42817.168250	500	50	32

2	Predictor-Corrector	5 min	42817.168250	500	50	32
3	Predictor-Corrector	11 min	33808.021788	5000	50	32
4	Euler-Maruyama	11 min	33808.021788	5000	50	32
5	Euler-Maruyama	1h 30 min	33400.657628	45000	50	32
6	Predictor-Corrector	1h 30 min	33400.657628	45000	50	32
7	Euler-Maruyama	6 h	33373.818287	45000	200	64
8	Predictor-Corrector	6 h	33373.818287	45000	200	64

Table. 7.1.4. DDPM++ results parameters comparison

Looking at the images in the figure and the table values it is noticeable that too many epoch will overfit the model , while the Average Loss is lower the best generated images are from output 6 , the biggest dataset with low epoch and batch size for Predictor - Corrector sampler.

Between Predictor - Corrector outputs the evolution of the images generated is better as far as it has low epoch / batch and a lot of data. The first outputs are mostly black , as the real images are in black and white, it is noticeable how as the data increases we can see in output 3 that it recognizes some white in the corner bottom right of the images. This white corner persists over bigger datasets with more variations over the placement which may be caused by the transformation layers of the model.

Results suggest that the ideal parameters would be having a larger dataset but keeping the epoch number and batch lower than 200 / 64

Because images are not accurately generated between at least two models there is no computation of the FID value to compare between them. Further project development and improvement in the next chapter will cover next steps.

7.1. Further project development and improvement

The next steps for this project are finishing properly the adjustments for DDPM++ and correctly train the missing models., as the conclusions show lower time training gets better results than the largest performed. Variations of the layers may be an option out of the scope of the project that can be explored.

After generating comparable images the FID and other measures may be applied to the results and validated with real experts, Although the project would require more hours of dedication, once it is completed some of the experts in the field who have collaborated on it can provide validation and act as a discriminator by observing the false images with the true ones mixed and measuring at the user level the distinction between the model results.

Other implementations outside the scope that can serve as steps, for a complete study of generative models of the LIDC-IDRI dataset, would be the implementation of variational autoencoder.

It is also a proposal to create an architecture that combines the results of the different models in a classifier that presents to the final user the images with the most correct validation values.

8. Bibliography

- [1] S. Bond-Taylor, A. Leach, Y. Long, C. G. Willcocks . *Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models* . arXiv:2103.04922v4 [cs.LG] 28 Mar 2022
- [2] J.Gui, Z. Sun, Y. Wen, D. Tao, J. Ye . *A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications* . arXiv:2001.06937v1 [cs.LG] 20 Jan 2020
- [3] N. Killoran, L. J. Lee, A. DeLong, D. Duvenaud, and B. J. Frey, *Generating and designing dna with deep generative models*, arXiv preprint arXiv:1712.06148, 2017
- [4] W. Dai, J. Doyle, X. Liang, H. Zhang, N. Dong, Y. Li, and E. P. Xing, *Scan: Structure correcting adversarial network for chest xrays organ segmentation*. arXiv preprint arXiv:1703.08770, vol. 1, 2017
- [5] M. Woodland , J. Wood , Brian M. Anderson, S. K. E. Lin , E. Koay , B. Odisio et al. *Evaluating the Performance of StyleGAN2-ADA on Medical Images* , arXiv:2210.03786v1
- [6] B. Senegal, D. M. Rubin, G. Rubin, A. Pantanowiz. *EVALUATING THE CLINICAL REALISM OF SYNTHETIC CHEST X-RAYS GENERATED USING PROGRESSIVELY GROWING GANS* , arXiv:2010.03975v2 [eess.IV] 10 Mar 2021
- [7] 6 GAN Architectures you really should know [online] [consulted : 10/01/2022] . Available at : <https://neptune.ai/blog/6-gan-architectures>
- [8] T. Karras , S. Laine, T. Aila . *A Style-Based Generator Architecture for Generative Adversarial Networks*. arXiv:1812.04948v3 [cs.NE] 29 Mar 2019
- [9] A. Aljohani , N. Alharbe, *Generating synthetic images for healthcare with Novel Deep Pix2Pix GAN*, Electronics 2022, 11, 3470.

- [10] McNitt-Gray MF, Armato SG 3rd, Meyer CR, Reeves AP, McLennan G, Pais RC, et Al. *The Lung Image Database Consortium (LIDC) data collection process for nodule detection and annotation*, Acad Radiol. 2007 Dec;14(12):1464-74. doi: 10.1016/j.acra.2007.07.021. PMID: 18035276; PMCID: PMC2176079.
- [11] Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, et al. (2013). *The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository. Journal of Digital Imaging*, 26(6), 1045–1057. <https://doi.org/10.1007/s10278-013-9622-7>
- [12] Armato III, S. G., McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, et al. (2015). Data From LIDC-IDRI [Data set]. *The Cancer Imaging Archive*. <https://doi.org/10.7937/K9/TCIA.2015.LO9QL9SX>
- [13] Wei Zhang , (2020) [Data set] LIDC-IDRI <<https://www.kaggle.com/datasets/zhangweiled/lidcidri>>
- [14] Andre Ye (2021) Loading and Reading LIDC-IDRI <<https://www.kaggle.com/code/washingtongold/loading-and-reading-lidc-idri/notebook>>
- [15] T. Karras , S. Laine, T. Aila et Al . *Analyzing and improving the Quality of StyleGAN*. arXiv:12006.06676 [cs.CV] 3 Dec 2019
- [16]T. Karras , S. Laine, T. Aila et Al . *Traininig Generative Adversarial Networks with Limited Data*. arXiv:2006.06676 [cs.CV] 11 Jun 2020
- [17]zikpefu , (2022) RuntimeError: aten::grid_sampler_2d_backward() is missing value for argument 'output_mask'.<<https://github.com/pytorch/pytorch/issues/75018>>
- [18] Yang Song et Al.*Score-Based Generative Modeling through Stochastic Differential Equations* arXiv:2011.13456 [cs.LG] 26 Nov 2020

