

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculté des Sciences et de la technologie

Département d'Electronique

Mémoire

Présenté pour obtenir

LE DIPLOME DE MASTER

FILIERE : ELECTRONIQUE

Spécialité : Systèmes de Télécommunication

Par

- **ABDENNOUR MOULOU**
- **BELFEROUM AHMED ELZINE**

Intitulé

Débruitage d'images médicales à faible SNR a l'aide de Réseau CycleGAN

Soutenu le : 2022/06/21

Devant le Jury composé de :

<i>Nom & Prénom</i>	<i>Grade</i>	<i>Qualité</i>	<i>Etablissement</i>
<i>Mr A. HACINE GHARBI</i>	<i>MCA</i>	<i>Président</i>	<i>Univ-BBA</i>
<i>Mme Z.MESSALI</i>	<i>Prof</i>	<i>Encadreur</i>	<i>Univ-BBA</i>
<i>Melle S. SID AHMED</i>	<i>MCB</i>	<i>Co-Encadreur</i>	<i>Univ-BBA</i>
<i>Mr. D. BOUDECHICHE</i>	<i>MCB</i>	<i>Co-Encadreur</i>	<i>Univ-BBA</i>

Année Universitaire 2021/2022

Remerciements

*En guise de reconnaissance, nous tenons à témoigner nos sincères remerciements à toutes les personnes qui ont contribué de près ou de loin au bon déroulement de nos stages de fin d'étude et à l'élaboration de ce modeste travail.

*Nous remercions notre encadrante MESSALI Zoubeida pour le travail énorme qu'elle effectue pour nous créer les conditions les plus favorables pour le déroulement de nos études. Grand merci pour ses conseils, ses directives et son aide pour finaliser ce travail.

* Nos sincères remerciements à nos co-encadreurs Dr. Boudechiche Djamel et Dr. Sid Ahmed Soumia qui ont assisté ce travail et qui l'ont enrichi par leurs idées, leurs conseils. Ils étaient toujours disponibles pour finaliser la partie simulation

*Nos sincères gratitude à tous les profs pour la qualité de leurs enseignements, et leurs conseils et leur intérêt incontestable qu'il porte à tous les étudiants.

Je tiens à remercier l'ensemble du personnel pour leur patience, leurs conseils pleins de sens et pour le suivi et l'intérêt qu'ils ont portés à nos travaux.

*Dans l'impossibilité de citer tous les noms, nos sincères remerciements vont à tous ceux et celles, qui de près ou de loin, ont permis par leurs conseils et leurs compétences la réalisation de ce mémoire.

Merci à tous...

‘Dédicace’

- *A nos chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.*
- *A nos chères sœurs pour leurs encouragements permanents, et leur soutien moral.*
- *A nos chers frères pour leur appui et leur encouragement*
- *A toute nos familles pour leur soutien tout au long de nous parcours universitaire,*
- *Que ce travail soit l’accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,*
- *Merci d’être toujours là pour nous.*

‘Que DIEU vous accorde santé et prospérité’.

Résumé

Notre travail s'inscrit dans le cadre du problème de débruitage d'images dans un contexte d'apprentissage profond. Nous implémentons deux variantes du réseau antagonistes discriminateurs GAN, à savoir Deep Convolutional GAN (DCGAN) et CycleGAN. Les deux réseaux sont appliqués pour améliorer la qualité des images TEP à faible dose. Pour évaluer les performances des deux réseaux, deux métriques sont pris en considération : le rapport max signal sur bruit (PSNR) et l'indice de similarité structurelle (SSIM) . Les résultats obtenus sont acceptables et prouvent la supériorité du réseau CycleGAN ent termes des deux critères et de la qualité des images obtenus.

Abstract

Our work deals with image denoising problem in the context of deep learning. We implement two generative adversarial networks (GAN): deep convolutional networks and CycleGAN. Two metrics are used in performance assessment of the considered denoising algorithms Peak signal to noise ratio (PSNR) and Structural Similarity Index (SSIM). The obtained results demonstrate the superiority of CycleGAN in the sens of PSNR, SSIM and visual quality.

ملخص

اليوم يتحدث الجميع عن الذكاء الاصطناعي والتعلم العميق. كيف لا ، وهذا المجال ينمو يومياً وبسرعة كبيرة. جعلت مزاياها العديدة وغير المسبوقة من أهم الأشياء في عصرنا.

وصلت مزاياها إلى التأثير في تصنيف الصور من خلال الشبكات العصبية (CNN)

في هذا الموجز ، اخترنا استخدام هذا العلم في مجال الطب من خلال التقاط صور منخفضة الجودة (جرعة منخفضة) وتحويلها إلى صور يمكننا رؤيتها بوضوح. النتائج التي تم الحصول عليها مقبولة ومثبتة بواسطة مقياسين psnrssim

والجودة المرئية

Sommaire

1.1	Introduction.....	3
1.2	Deep Learning.....	4
1.3	Réseaux de neurones (Neural Networks NN).....	5
1.3	Modèle du Perceptron [3].....	5
1.4	Structure des réseaux de neurones multicouche (multi layer NN).....	7
1.5	Types d'apprentissage.....	8
1.5.a	L'apprentissage supervisé [6]:.....	8
1.5.b	L'apprentissage non supervisé.....	8
1.6	Algorithmes d'apprentissage [7].....	9
1.6.a	Regression logistique.....	9
1.6.b	Fonction cout (Cost Function).....	9
1.7	Propagation directe et inverse dans un réseau.....	10
1.8	Optimisation dans l'apprentissage automatique.....	10
1.8.a	Algorithme Gradient Descent [8].....	11
1.9	Batch Size.....	11
1.10	Ensemble de données (data sets) dans un réseau.....	12
1.11	PARAMETERS VS HYPERPARAMETERS :.....	12
1.12	Conclusion.....	13
2.1	Introduction.....	15
2.2	Formulation du débruitage d'images [9].....	15
2.3		
	Principe des réseaux antagonistes génératifs (Generative Adversarial Networks GAN)	Principe 16
2.3.1	(Fonction loss) du discriminateur et du générateur [11]	Fonctions pertes 19
2.3.2	Algorithme d'apprentissage du GAN.....	19
2.4	Deep Convolutional GAN (DCGAN).....	20
2.5	Réseau Cycle GAN.....	23
2.6	Architecture de CycleGAN.....	24
2.7	Critères d'évaluation.....	25

3.1	Introduction.....	27
3.2	Nature des Données.....	27
3.2.1	Base MNIST	27
3.2.2	Base Big-Medical-Data-set [14]	27
3.3	Présentation des outils de développement..	28
3.4	Implémentation du Réseau DCGAN.....	28
3.5	Implémentation du réseau CycleGAN	34
3.6	Conclusion.....	39

Listes des Figures

Figure 1.1 Sous domaines d'IA	page 3
Figure 0-1 Principe de Deep learning	page 4
Figure I 3 principe de machine learning	page 4
Figure 0-2 neurone biologique vs neurone artificiel	page 5
Figure I 7 processus optimization	page 10
Figure 0.8 Ensemble de données	page 12
Figure 2.1 Exemple de filtrage d'images par méthodes conventionnelles	page 16
Figure 2.2 Schéma bloc du GAN	page 17
Figure 2.3 Image to image (Paired Images) page 18	page 18
Figure 2.4 Pixel to Pixel (unpaired images) page 18	page 18
Figure 2.5 Architecture du discriminateur d'un DCGAN	page 20
Figure 2.6 Architecture du générateur du DCGAN	page 21
Figure 2.7 Architecture générale du DCGAN [12]	page 22
Figure 2.8 Principe du CycleGAN	page 23
Figure 2.9 Architecture du CycleGAN	page 24
<i>Figure 2.10 PSNR formule</i>	page 24
<i>Figure 2.11 SSIM formule</i>	page 24
Figure 3.5 Etapes d'implémentation du réseau DCGAN	page 29
Figure 3.6 Réseau DCGAN	page 30
Figure 3.7.a Image input de chiffres bruit du DCGAN	page 31
Figure 3.7.b Déroulement du programme DCGAN	page 32

Figure 3.8 Images débruitées par DCGAN à différents nombres de batchs page 34

Figure 3.9 Image TEP (a) *Full dose* et (b) *Low dose images* page 35

Figure 3.10 Architecture de CycleGAN appliqué en débruitage page 36

Figure 3.11 Organigramme de fonctionnement du CycleGAN page 37

Figure 3.12 Résultat des images reconstruites obtenues par CycleGAN

à partir des images LD page 38

Figure 3.13 Fonctions loss du générateur et du discriminateur en fonction du nombre d'epochs. page 39

Notation ET Abréviation

CONV : couche de convolution

ReLU : Unité de rectification linéaire

SGD : Stochastique Gradient Descente

Tanh : Tangente Hyperbolique

POOL: Pooling.

SSIM : Mesure de similarité

CNN : convolution neurons network

CPU : Central Processing Unit

GPU : Graphics Processing Unit

PSNR : Peak Signal to Noise Ratio

SSIM : Structural Similarity

Introduction Générale

La science d'aujourd'hui a rendu plus facile pour une personne de vivre sa vie en faisant des choses que même une personne ne peut pas faire parfois, tout cela en utilisant uniquement un ordinateur.

L'apprentissage automatique (Machine Learning ML) est le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés. ML est l'une des technologies les plus excitantes que l'on aurait jamais rencontrées. Comme son nom l'indique, il donne à l'ordinateur ce qui le rend plus semblable à l'homme : La capacité d'apprendre. L'apprentissage automatique est activement utilisé aujourd'hui, dans beaucoup plus d'endroits qu'on ne le pense.

Aussi La vision par ordinateur évolue rapidement de jour en jour. L'apprentissage en profondeur fait partie du domaine ML. Les réseaux neuronaux convolutifs (abrégié en CNN) ont montré leurs efficacité dans plusieurs applications. De plus les réseaux antagonistes générateurs (Generative Adversarial Network GAN) sont actuellement utilisés pour générer de nouvelles images à partir d'images existantes. Leurs applications est très large.

Notre travail s'inscrit dans le cadre de débruitage d'images dans un contexte d'apprentissage profond. Plus précisément, nous nous sommes basés sur deux types de réseaux GAN, à savoir Deep Convolutional GAN DCGAN et CycleGAN pour améliorer la qualité visuelle des images TEP à faible dose. Pour évaluer les performances du réseau CycleGAN, deux critères sont calculés : le rapport max signal sur bruit (PSNR) et l'indice de similarité structurel (SSIM). Les résultats obtenus sont prometteurs et démontrent la supériorité du CycleGAN.

Organisation du Manuscrit

Ce manuscrit est constitué de la présente introduction et de trois chapitres plus la conclusion générale.

Chapitre 1 introduit les concepts de base l'apprentissage et l'apprentissage profond. Les paramètres et les hyperparamètres des réseaux neuronaux profonds sont définis. Quelques architectures de réseaux neuronaux sont également présentées. L'avantage d'utiliser le deep learning est souligné.

Chapitre 2 traite les réseaux antagonistes génératifs (Generative Adversarial Networks GAN). Le problème de débruitage d'images est d'abord présenté. Le principe du GAN est largement détaillé. Les deux types DCGAN et CycleGAN sont expliqués. Une grande attention est donnée aux algorithmes d'optimisation utilisés tel que descente gradient. Les fonctions objectives du générateur et du discriminateur sont bien définies.

Chapitre 3 est entièrement dédié à la présentation et à la discussion des résultats d'implémentation des algorithmes DCGAN et CycleGAN. Les deux réseaux sont appliqués sur des images détériorées. CycleGAN est utilisé pour améliorer la qualité visuelle des images TEP à faible dose (Low Dose LD). Deux bases de données sont utilisées : la base MNIST et la base Big-medical-data. L'évaluation qualitative du CycleGAN est basée sur le calcul des PSNR et SSIM en plus de la qualité d'images débruitées. Les résultats obtenus sont satisfaisants et prometteurs en termes de critères considérés..

Chapitre 1

Concepts de Base de l'apprentissage profond

Résumé

Dans ce chapitre, nous présenterons un aperçu sur l'intelligence artificielle ainsi que les concepts de l'apprentissage et l'apprentissage profond (Machine Learning ML et Deep Learning DL) . Nous introduisons tous les paramètres et hyperparamètres du réseau de neurones

Sommaire

1.1 Introduction

1.2 Deep Learning

1.3 Réseaux de neurones (Neural Networks NN)

1.4 Modèle du Perceptron

1.5 Types d'apprentissage

1.6 Préparation de l'ensemble de données

1.6 Conclusion

1.1 Introduction

L'intelligence artificielle (IA) en tant qu'intelligence démontrée par les machines est une approche efficace pour comprendre l'apprentissage et le raisonnement humain.

L'IA comprend les sous domaines suivants comme c'est illustré dans la Figure 1.1 :

- machine learning
- deep learning
- natural language processing
- cognitive computing
- computer vision
- neural network

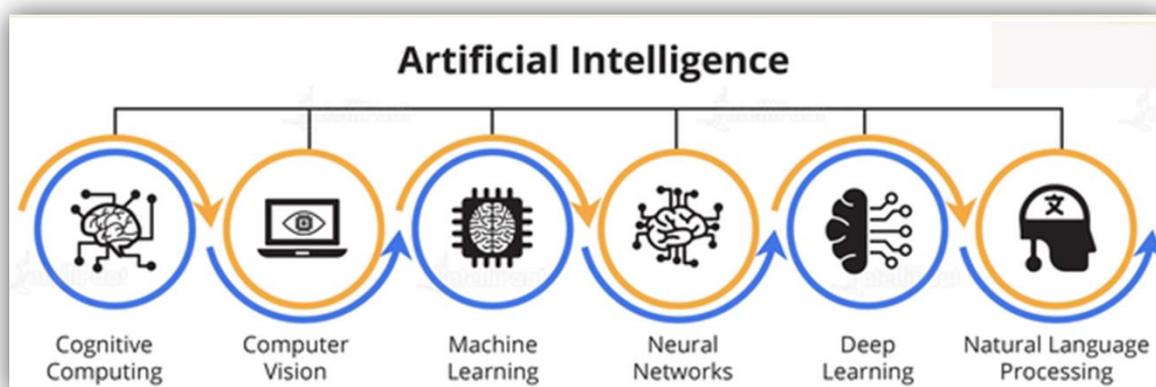


Figure 1.1 Sous domaines d'IA [1]

Les applications de l'IA sont multiples. Nous citons à titre d'exemple : la cybersécurité, la santé, le commerce, la recherche

Les réseaux de neurones (NN) et l'apprentissage profond (deep learning (DL)), qui font partie de l'IA, apportent actuellement la meilleure solution à de nombreux problèmes dans les domaines de traitement d'images, de la voix et du traitement du langage naturel.

Dans ce Chapitre, nous exposerons les concepts de base de l'apprentissage profond. Nous mettrons l'accent sur la modélisation du neurone, la structure d'un réseau

profond ainsi que les algorithmes d'optimisation utilisées et les hyperparamètres d'un réseau.

1.2 Deep Learning

Deep Learning est une technique de machine learning. Il enseigne à une machine à traiter les entrées par couches afin de classer, inférer et prédire le résultat comme c'est montré sur la Fig. 1.2. L'apprentissage automatique (machine learning) a la capacité d'apprendre à travers des Entrainement avec des modèles . La fin de cette formation lui donne la capacité de distinguer les choses. Dans Figure I-3, par exemple, le réseau essaie d'apprendre à différencier les choses. Bien que l'apprentissage en profondeur et l'apprentissage automatique soient devenus très connus et utiles de nos jours, il existe une différence distincte entre eux :

« deep learning » est un domaine de l'intelligence artificielle basé sur des réseaux de neurones et ne nécessite aucune interférence humaine pour apprendre, comme présenté dans la figure 1.2, cependant, l'apprentissage automatique est un domaine de l'intelligence artificielle qui nécessite des conseils humains, comme illustré dans la figure I.3.

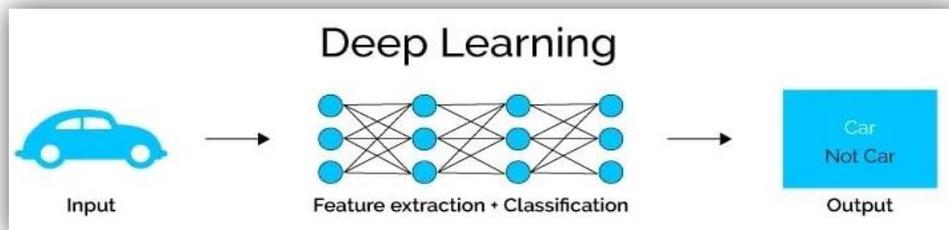


Figure 0-1 Principe de Deep learning [3]

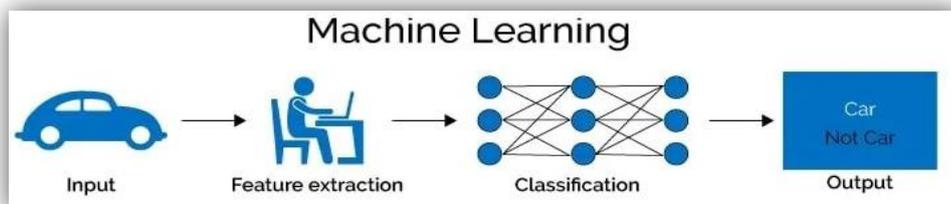


Figure 0-3 principe de machine learning [4]

1.3 Réseaux de neurones (Neural Networks NN)

La structure des réseaux de neurones est dérivée du cerveau humain. Elle simule la façon dont les vrais neurones communiquent entre eux [2]. La Figure I-4 illustre la structure d'un perceptron biologique et son modèle simulé. Les données d'entrée sont traitées à travers différentes couches de neurones artificiels empilés ensemble pour produire la sortie désirée.

L'architecture du réseau neuronal est composée d'unités individuelles appelées neurones qui imitent le comportement biologique du cerveau.

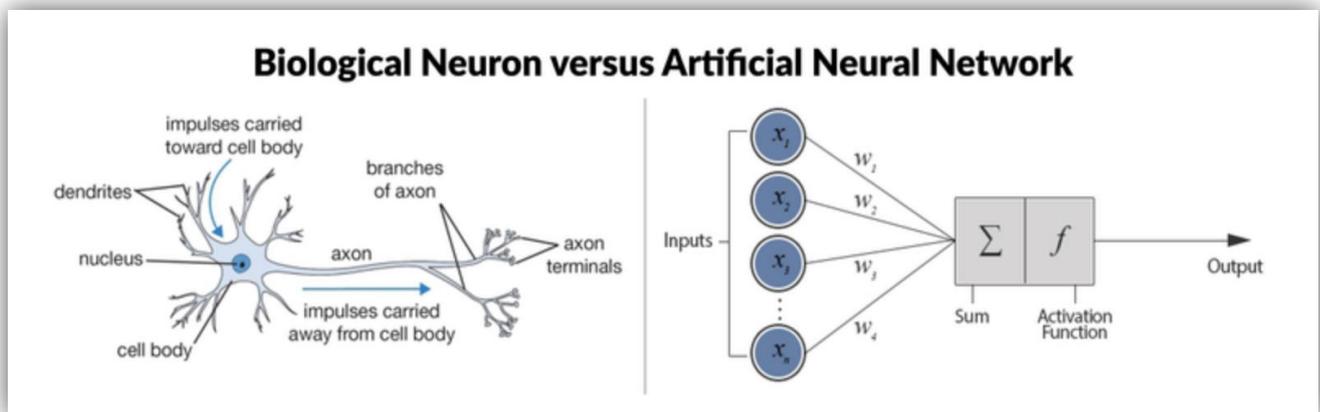


Figure 0-1 neurone biologique vs neurone artificiel [3]

1.3 Modèle du Perceptron [3]

Les différents composants d'un neurone sont :

- **Entrée (Input)**

C'est l'ensemble des caractéristiques qui sont intégrées dans le modèle pour le processus d'apprentissage. Par exemple, l'entrée dans la détection d'objet peut être un tableau de valeurs de pixel relatives à une image.

- **Poid (Weight)**

Sa fonction principale est de donner de l'importance aux caractéristiques qui contribuent d'avantage à l'apprentissage. Il le fait en introduisant la multiplication scalaire entre la valeur d'entrée et la matrice de poids.

- **Biais**

Le rôle du biais est de déplacer la valeur produite par la fonction d'activation. Son rôle est similaire au rôle d'une constante dans une fonction linéaire (équivalent à un seuil négatif).

- **Pas d'apprentissage (learning rate)**

Le taux d'apprentissage est un hyper-paramètre qui contrôle dans quelle mesure nous ajustons les poids du réseau. Il est calculé à l'aide du gradient de perte. Plus la valeur est faible, plus nous mettons à jour les pondérations lentement (obtenons de meilleurs résultats), mais l'étape d'apprentissage prendra plus de temps.

- **Fonction de transfert (Transfer function)**

Le travail de la fonction de transfert est de combiner plusieurs entrées en une seule valeur de sortie afin que la fonction d'activation peut être appliquée. Cela se fait par une simple sommation de toutes les entrées de la fonction de transfert.

- **Fonction d'activation (Activation Function)**

Elle introduit la non-linéarité dans le fonctionnement des perceptrons pour considérer la linéarité variable avec les entrées. Sans cela, la sortie ne serait qu'une combinaison linéaire de valeurs d'entrée et ne serait pas en mesure d'introduire la non-linéarité dans le réseau. La Figure 1. 5 illustre différentes fonctions d'activation, tel que :

x représente les entrées

ω_i représente les poids

b_i le biais

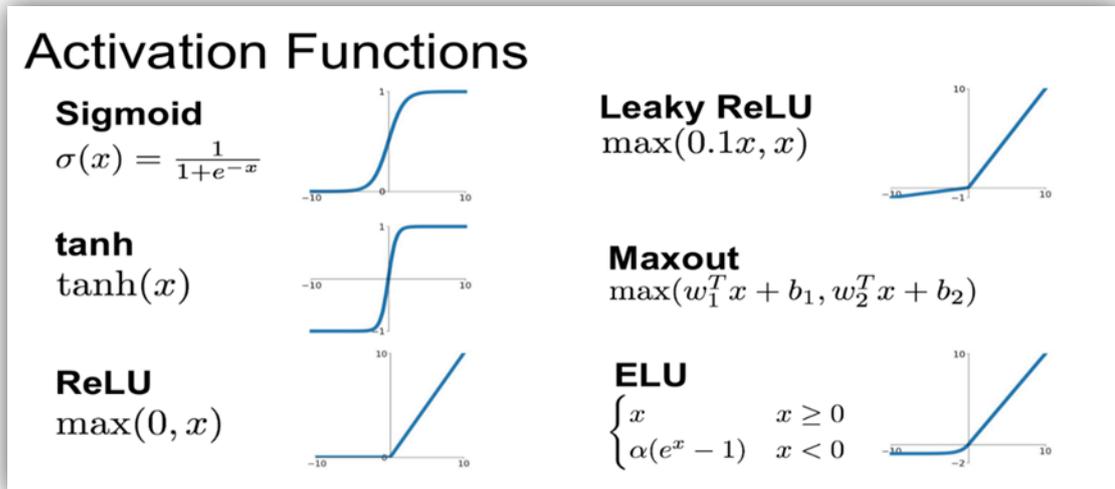


Figure 0-1 Types de fonctions d'activation [3]

1.4 Structure des réseaux de neurones multicouche (multi layer NN)

Lorsque plusieurs neurones sont empilés ensemble dans une rangée, ils constituent une couche, et plusieurs couches empilées les unes à côté des autres sont appelées un réseau neuronal multicouche [5].

La Figure I-6 montre les principales composantes de ce type de structure.

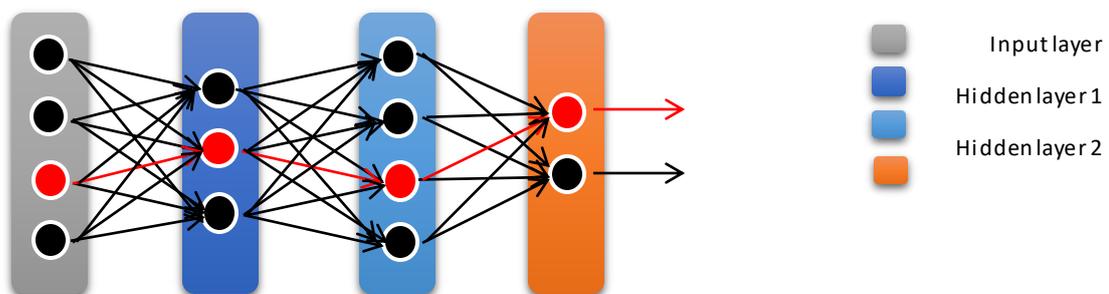


Figure 0-1 Structure d'un réseau de neurone multicouche

- **Couches cachées (Hidden Layers)**

Les couches cachées sont ce qui fait de l'apprentissage profond ce qu'il est aujourd'hui. Ce sont des couches intermédiaires qui font tous les calculs et extraient les caractéristiques des données. Il peut y avoir plusieurs couches cachées interconnectées qui prennent en compte la recherche de différentes fonctionnalités cachées dans les données.

- **Couche de sortie (Output Layer)**

La couche de sortie prend les entrées des couches cachées précédentes et arrive à une prédiction finale basée sur les apprentissages du modèle. C'est la couche la plus importante où nous obtenons le résultat final.

Dans le cas des modèles de classification/régression, la couche de sortie a généralement un nœud unique. Cependant, il est complètement spécifique au problème et dépend de la façon dont le modèle a été construit.

1.5 Types d'apprentissage

Au sein de l'intelligence artificielle (IA) et de l'apprentissage automatique, il existe deux approches de base : l'apprentissage supervisé et l'apprentissage non supervisé.

1.5.a L'apprentissage supervisé [6]:

L'apprentissage supervisé est une approche d'apprentissage automatique qui se définit par son utilisation d'ensembles de données étiquetés. Ces ensembles de données sont conçus pour entraîner ou « superviser » des algorithmes afin de classer les données ou de prédire les résultats avec précision. À l'aide d'entrées et de sorties étiquetées, le modèle peut mesurer sa précision et apprendre au fil du temps.

Les problèmes d'apprentissage supervisé peuvent être regroupés en problèmes de régression et de classification

- La régression : les données sont dans un ensemble continu tels que les projections de chiffre d'affaires pour une entreprise donnée.
- La classification : les données sont dans un ensemble discret, exemple: classer les e-mails comme spam et non spam.

1.5.b L'apprentissage non supervisé

Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé a ces données d'entraînement comme des entrées sans sorties connues. Ce type d'apprentissage est utilisé pour certain type de problèmes comme étudier les caractéristiques et le prétraitement des données.

L'objectif du modèle est de grouper avec succès les observations disponibles en catégories.

1.6 Algorithmes d'apprentissage [7]

1.6.a Regression logistique

La régression logistique est une régression linéaire au sens large, et c'est aussi un algorithme d'apprentissage utilisé dans l'apprentissage supervisé. Il est généralement utilisé dans les problèmes où la valeur de sortie est 0 ou 1, c'est-à-dire "oui" ou "non". C'est le problème de classification binaire.

1.6.b Fonction cout (Cost Function)

C'est une fonction qui mesure le rendement d'un modèle d'apprentissage automatique pour des données. La fonction coût quantifie l'erreur entre les valeurs prévues et les valeurs attendues (désirées) et la présente sous la forme d'un seul nombre réel. Selon le problème, la fonction coût peut être formée de nombreuses façons différentes. L'objectif de la fonction coût est d'être soit:

- Minimisé - la valeur retournée est habituellement appelée coût, perte ou erreur. Le but est de trouver les valeurs des paramètres du modèle (les poids et le biais), pour lesquels la fonction de coût renvoie le plus petit nombre possible. Elle est généralement définie en norme L_2 , c.à.d une erreur quadratique moyenne entre la sortie estimée et la sortie désirée :

$$\mathcal{L}(\theta) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \quad (1.1)$$

Avec :

θ le vecteur des paramètres (poids et biais)

\mathbf{y} et $\hat{\mathbf{y}}$ sont les vecteurs de sorties désirées et sorties estimées respectivement.

- Maximisé Le but est de trouver des valeurs de paramètres de modèle pour lesquelles le nombre retourné est aussi grand que possible.

Pour les algorithmes utilisant Gradient Descent pour optimiser les paramètres du modèle, chaque fonction doit être différentiable.

1.7 Propagation directe et inverse dans un réseau

il y a deux types de propagation dans NN comme présenté dans la figure I-8

La "propagation directe « avant » fait référence au processus de calcul, aux valeurs des couches de sortie à partir des données d'entrée. Il traverse tous les neurones de la première à la dernière couche.

Une fonction de perte est calculée à partir des valeurs de sortie.

Et puis "propagation inverse « recto » ou en arrière fait référence au processus de comptage des changements de poids (apprentissage de facto), en utilisant un algorithme de descente de gradient (ou similaire). Le calcul est effectué à partir de la dernière couche, en remontant jusqu'à la première couche.

1.8 Optimisation dans l'apprentissage automatique

L'optimisation est la capacité de comprendre un problème réel, puis de le transformer en un modèle mathématique qui peut être étudié pour extraire les propriétés structurelles et caractériser les solutions aux problèmes.

L'optimiseur utilise le score obtenu à partir de la fonction de perte pour ajuster les valeurs de poids, ce qui réduit le score de perte. L'organigramme est illustré en figure I-7

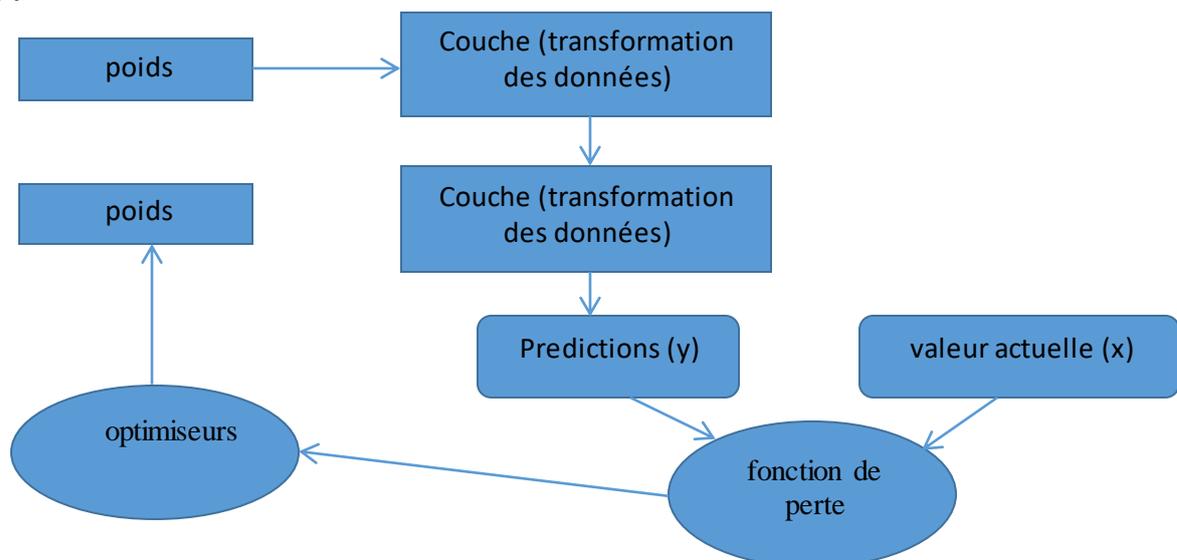


Figure 0-7 processus optimization

1.8.a Algorithme Gradient Descent [8]

L'algorithme de descente en gradient et ses variantes (Adam, SGD etc.) sont devenus des algorithmes de formation (optimisation) très populaires dans de nombreuses applications de machine learning. Les algorithmes d'optimisation peuvent être regroupés de façon informelle en deux catégories : ceux basés sur les gradients et ceux sans gradient. La descente gradient est un algorithme basé sur des gradients.

L'objectif de la formation d'un modèle d'apprentissage automatique est de minimiser la perte ou l'erreur entre les vérités au sol (ground truth) et les prédictions (les estimées) en modifiant les paramètres de formation (ajustant les poids et les biais). Le gradient, qui est l'extension de dérivé dans l'espace multidimensionnel, indique la direction dans laquelle la perte ou l'erreur est optimalement minimisée. La formule pour la descente de gradient est simplement :

$$\text{repetition jusq au convergence } \left\{ \frac{\partial}{\partial \theta_j} J(\theta) \alpha \rightarrow \theta_j \rightarrow \theta_j \right\} \quad (1.2)$$

Ou θ_j est le vecteur des paramètre entraînable, α est le taux d'apprentissage. $J(\theta)$ est le fonction coût.

1.9 Batch Size

On peut faire un pas dans la descente du gradient, et mettre à jour les paramètres de différentes manières. L'approche choisie définira considérablement la performance et la convergence du modèle. Il est essentiel de souligner les concepts d'époque et de batch. Une fois que notre ensemble d'entraînement complet est vu par le modèle, nous disons qu'une époque est terminée. Pendant la formation, nous pouvons mettre à jour les poids et le gradient après qu'un sous-ensemble de l'ensemble de la formation ait été calculé par le modèle. Nous appelons cela un batch de sous-ensemble ou un mini-batch. Préparation de l'ensemble de données

1.10 Ensemble de données (data sets) dans un réseau

L'ensemble de données (data set) est une collection de données. En d'autres termes, un ensemble de données correspond au contenu d'une seule table de base de données, ou d'une seule matrice de données statistiques, où chaque colonne de la table représente une variable particulière, et chaque ligne correspond à un membre donné de l'ensemble de données en question.

Dans les projets d'apprentissage automatique, nous avons besoin d'un ensemble de données de formation. C'est l'ensemble de données réel utilisé pour former le modèle pour effectuer diverses actions

On divise les données en quatre parties

Supposons que nous ayons un ensemble de données complet. Nous pouvons diviser cet ensemble de données en trois sous-ensembles comme c'est montré sur la Figure 1.8



Figure 0.8 Ensemble de données

- 1. Ensemble d'apprentissage (Training Set):** Nous formons le modèle sur les données de formation.
- 2. Dev Set:** Après avoir formé le modèle, nous vérifions comment il fonctionne sur le dev set.
- 3. Ensemble de Test (test data Set):** Lorsque nous avons un modèle final (c.-à-d. le modèle qui a donné de bons résultats dans les deux formations) ainsi que dev set), nous l'évaluons sur l'ensemble de test afin d'obtenir une estimation impartiale de la façon dont notre algorithme fait

1.11 PARAMETERS VS HYPERPARAMETERS :

La principale différence entre les paramètres et les hyperparamètres est que les paramètres sont appris par le modèle pendant le temps de formation (ω et b), alors que les hyperparamètres peuvent être modifiés avant la formation du modèle.

il y a beaucoup d'hyperparamètres pour un NN profond, y compris:

- Taux d'apprentissage
- Nombre d'itérations
- Nombre de calques masqués (couches cachées)
- Unités dans chaque couche cachée
- Choix de la fonction d'activation

1.12 Conclusion

Dans ce chapitre, nous avons présenté les concepts fondamentaux de l'IA et l'apprentissage profond et les différentes structures. Nous avons présenté le fonctionnement, les différentes architectures et les algorithmes d'entraînement ainsi que les caractéristiques principales des réseaux de neurones artificiels. L'accent est mis sur le modèle du perceptron ainsi que l'algorithme d'apprentissage descent gradient.

Le chapitre suivant détaillera un autre type de réseaux de neurones qui est le réseau de neurone convolutif (CNN), choisie dans notre thème de recherche. Nous exposerons le réseau antagoniste génératif GAN (Generative Adversarial Network) ainsi que l'architecture "Cycle_GAN". Ces deux réseaux font l'objet de notre étude et vont être implémentés pour la problématique du débruitage des images.

Chapitre 2

Concepts de base des réseaux Antagonistes Génératifs GAN

Résumé

Dans ce chapitre, nous présenterons les réseaux antagonistes génératifs (Generative Adversarial Network GAN). Plus précisément, nous détaillerons deux types principaux de ce réseaux : réseau GAN convolutif profond (Deep convolutional GAN DCGAN) et CycleGAN. Ces deux réseaux seront implémentés pour débruiter des images. Pour cela nous introduisons d'abord la formulation du problème de débruitage que nous adoptons

.dans notre étude

Sommaire

1.1 Introduction

1.2 Formulation du débruitage d'images

1.3 Principe des réseaux antagonistes génératifs

1.4 Fonctions pertes (Function loss)

1.5 Algorithme d'apprentissage du GAN

1.6 Réseau Cycle GAN

1.6 Conclusion

2.1 Introduction

Dans ce Chapitre, nous présenterons les concepts de base du réseau antagoniste génératifs (Generative Adversarial Network GAN). Nous mettrons l'accent sur les réseaux: Deep Convolutional DCGAN et CycleGAN. Dans notre étude, nous allons implémenter les réseaux DCGAN et CycleGAN pour le débruitage d'images médicales TEP à faible dose. Nous allons donc rappeler la problématique du débruitage d'images. Nous définissons ensuite les concepts de base du réseau GAN. Les réseaux DCGAN et CycleGAN seront ensuite présentés avec leurs architecture. Nous considérons le rapport signal maximum sur bruit (Peak signal to noise ratio PSNR) et l'indice de similarité (SSIM) pour évaluer quantitativement les résultats obtenus du débruitage.

Il convient, avant de détailler les réseaux GAN, d'ouvrir une parenthèse pour rappeler le problème du débruitage d'image et sa formulation.

2.2 Formulation du débruitage d'images [9]

Ce paragraphe s'intéresse à la formulation du problème de débruitage des images numériques, pour augmenter la qualité de leur rendu visuel, ou pour faciliter leur analyse. On cherche donc à atténuer, sinon supprimer une certaine dégradation.

Le modèle du débruitage d'image suppose que :

$$g = f + b \quad 2.1$$

Où g est l'image observée, f est l'image idéale et b est le bruit inhérent aux appareils de mesure physique ou une dégradation telle que la faible dose des images médicales TEP que nous allons traiter dans nos expériences de simulation. Il est nécessaire d'opérer des transformations qui pour chaque pixel tiennent compte des techniques dites de filtrage d'image en algorithmes de débruitage. Dans la littérature du traitement d'images, différentes méthodes de débruitage ont été proposées et développées. Tout d'abord, des méthodes de filtrage spatial ont été proposées. Celles-ci consistent à réduire le bruit dans les zones qui ne présentent pas d'objets intéressants et à accentuer la perception des structures d'intérêt. La Figure 2.1 illustre quelques exemples de filtrage d'images par différents filtres.

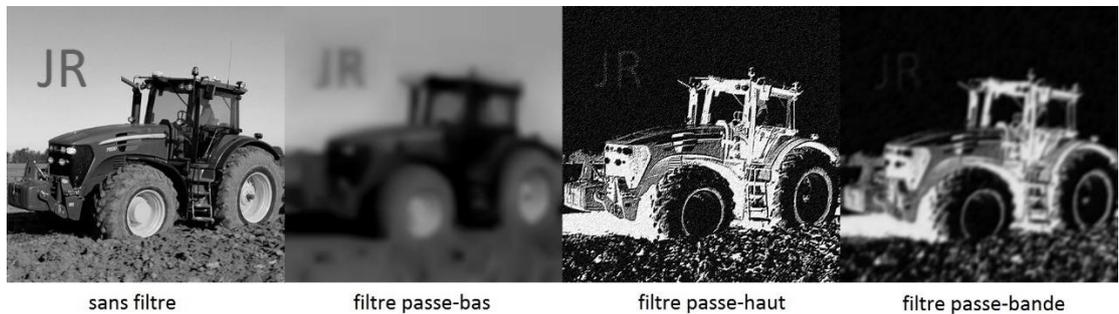


Figure 2.1 Exemple de filtrage d'images par méthodes conventionnelles

Le rôle des filtres passe-haut, est d'extraire les détails dans l'images tel que les contours. Dans un problème de débruitage, on dispose des observations bruitées et on veut reconstruire des images débruitées. L'objectif de cette étude est d'améliorer la qualité d'image détériorée en appliquant les réseaux GAN. On définit d'abord le principe d'un réseau GAN.

2.3 Principe des réseau antagonistes génératifs (Generative Adverserials Networks GAN)

Generative adversarial networks (ou GAN) en abrégé sont l'un des algorithmes d'apprentissage profond les plus populaires développés ces dernières années [10].

Le rôle d'un réseau GAN est de générer des images ou autres données (Fausaire) plus efficace que le détecteur. Il peut le faire lorsqu'il termine Son entraînement Il est par exemple possible de créer de faux visages grâce à cette technologie.

le GAN capable de générer des images fidèles à la réalité, Donc Sa position dans la recherche scientifique et au domaine de l'analyse d'images médicales devient de plus en plus importante (c'est pourquoi on a utiliser pour réaliser notre étude)

Les réseaux GAN sont composés de deux modèles (deux fonctions) :

- Le premier modèle s'appelle un **générateur** et vise à générer de nouvelles données similaires à celles attendues. Le générateur pourrait être assimilé à un faussaire d'art humain, qui crée de fausses œuvres d'art.
- Le second modèle s'appelle le **Discriminator**. L'objectif de ce modèle est de reconnaître si une donnée d'entrée est « *réelle* » — appartient à l'ensemble de données d'origine — ou si elle est « *fausse* » — générée par un faussaire. Dans ce scénario, un discriminateur est analogue à un expert en art, qui essaie de détecter les œuvres d'art comme véridiques ou frauduleuses.

les étapes suivies par un GAN :

- Le générateur prend des nombres aléatoires et renvoie une image.
- Cette image générée est introduite dans le discriminateur avec un flux d'images extraites de l'ensemble de données réel et réel.
- Le discriminateur prend à la fois des images réelles et fausses et renvoie des probabilités, un nombre compris entre 0 et 1, 1 représentant une prédiction d'authenticité et 0 représentant un faux.

La Figure 2.2 illustre ce modèle GAN.



Figure 2.2 Schéma bloc du GAN

Où G représente le générateur, D le discriminateur, G loss et D loss représentent les fonctions objectives du générateur et du discriminateur respectivement qu'il faut minimiser en optimisant à chaque itération.

Le discriminateur doit apprendre la différence entre les données générées (fake) et réelles (authentiques).

L'apprentissage du GAN se décompose en deux phases :

1. D'abord la différence entre les données (Fake) et authentiques pour aider Le discriminateur à distinguer les deux.
2. Le discriminateur informe le générateur comment ces images (fake) doivent être modifiées pour d'avantage ressembler aux authentiques. Cet apprentissage du générateur par le discriminateur se fait à travers l'algorithme de rétropropagation (backpropagation).

Il existe deux cas de génération d'images :

Translation Image to Image

Dans ce cas, on a un objet dans l'image input et on veut obtenir un autre objet dans l'image output. Soit l'input X et on veut obtenir une image générée $Y = G(X)$ comme c'est montré dans la Figure 2.3

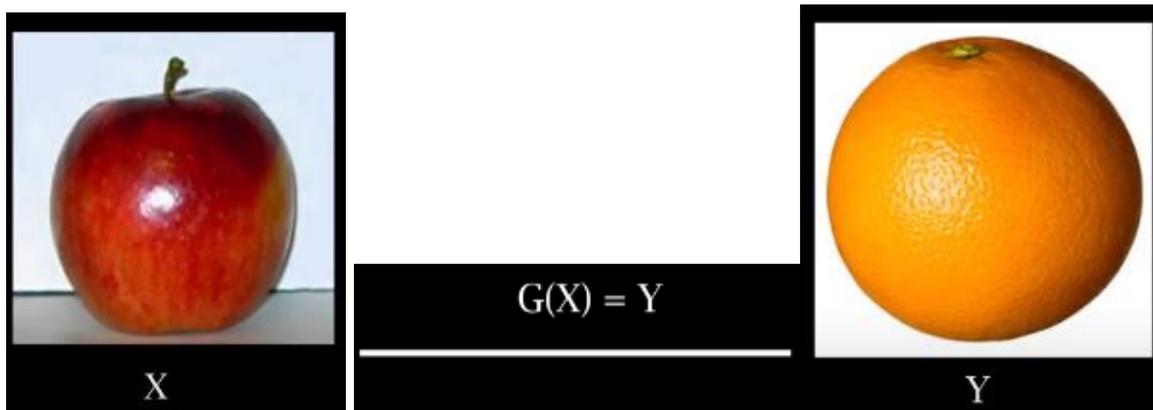


Figure 2.3 Image to image (Paired Images)

Translation Pixel to Pixel (Pix2Pix)

Dans ce cas, à partir d'une image input contenant un objet, on veut obtenir une image output contenant plusieurs objets comme c'est montré sur la Figure 2.4.

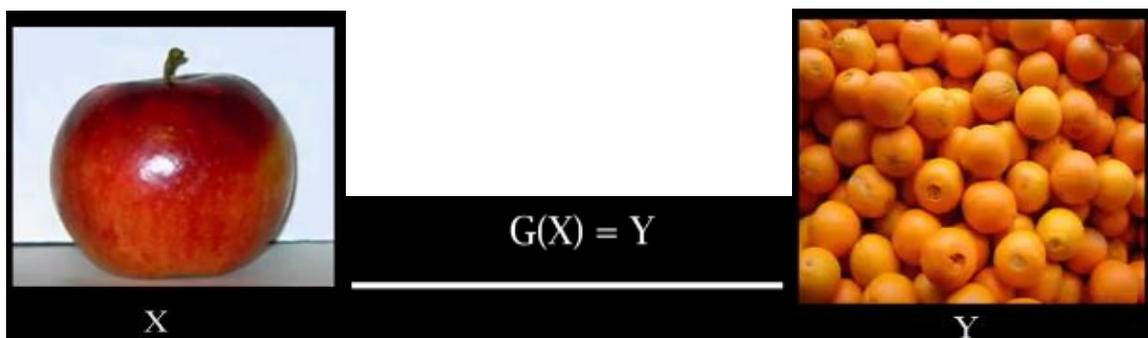


Figure 2.4 Pixel to Pixel (unpaired images)

On doit donc créer deux modèles ou deux fonctions : $G(X)$ pour générer des images à partir du générateur et $F(Y)$ pour reconstruire les images de l'input (\hat{X}). C'est la rétropropagation.

2.3.1 Fonctions pertes (Function loss) du discriminateur et du générateur [11]

Soient m données d'apprentissage $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ avec une distribution data p_{data} , et soient m échantillons bruit $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ avec une distribution p_z . $G(z^{(i)})$ est la sortie du générateur ayant les échantillons bruit en entrée. $D(x^{(i)})$ est la sortie du discriminateur pour les entrées $x^{(i)}$ et $D(G(z^{(i)}))$ la sortie du discriminateur pour les entrées $G(z^{(i)})$. Les fonctions loss du discriminateur et du générateur sont définies respectivement par :

$$J^{(D)} = \frac{1}{2m} \sum_{i=1}^m [(D(x^{(i)}) - 1)^2] + \frac{1}{2m} \sum_{i=1}^m [(D(G(z^{(i)})) - 1)^2] \quad 2.2$$

$$J^{(G)} = \frac{1}{m} \sum_{i=1}^m [(D(G(z^{(i)})) - 1)^2]$$

2.3.2 Algorithme d'apprentissage du GAN

L'algorithme d'apprentissage du GAN est donnée par :

Algorithme d'apprentissage GAN

1. Mapper les données d'apprentissage $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ selon p_{data}
2. Mapper les échantillons bruit $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ selon p_z .
3. Générer les images fausses (fake) à partir du bruit : $G(z^{(i)})$, $i \in \{1, 2, \dots, m\}$
4. Calculer la fonction loss du discriminateur :

$$J^{(D)} = \frac{1}{2m} \sum_{i=1}^m [(D(x^{(i)}) - 1)^2] + \frac{1}{2m} \sum_{i=1}^m [(D(G(z^{(i)})) - 1)^2]$$

5. Actualiser les paramètres du discriminateur
6. Mapper les échantillons bruit $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$
7. Générer les images fausses (fake) à partir du bruit : $G(z^{(i)})$, $i \in \{1, 2, \dots, m\}$
8. Calculer la fonction loss du générateur :

$$J^{(G)} = \frac{1}{m} \sum_{i=1}^m [(D(G(z^{(i)})) - 1)^2]$$

Algorithm 1 GAN Training Loop Pseudocode1: **procedure** TRAINGAN2: Draw m training examples $\{x^{(1)}, \dots, x^{(m)}\}$ from the data distribution p_{data} 3: **Draw m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from the noise distribution p_z** 4: **Generate fake images from the noise: $G(z^{(i)})$ for $i \in \{1, \dots, m\}$** 5: **Compute the (least-squares) discriminator loss:**

$$J^{(D)} = \frac{1}{2m} \sum_{i=1}^m \left[\left(D(x^{(i)}) - 1 \right)^2 \right] + \frac{1}{2m} \sum_{i=1}^m \left[\left(D(G(z^{(i)})) \right)^2 \right]$$

6: Update the parameters of the discriminator

7: **Draw m new noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from the noise distribution p_z** 8: **Generate fake images from the noise: $G(z^{(i)})$ for $i \in \{1, \dots, m\}$** 9: **Compute the (least-squares) generator loss:**

$$J^{(G)} = \frac{1}{m} \sum_{i=1}^m \left[\left(D(G(z^{(i)})) - 1 \right)^2 \right]$$

10: Update the parameters of the generator

2.4 Deep Convolutional GAN (DCGAN)

Le réseau Deep Convolutional GAN DCGAN est un réseau GAN utilisant les réseaux convolutifs (CNN) comme discriminateur et un réseau composé de convolutions comme générateur. Pour implémenter le DCGAN, on doit spécifier trois paramètres:

- Le discriminateur
- Le générateur
- La procédure d'apprentissage (training procedure)

Le discriminateur dans l'architecture du DCGAN est un CNN qui suit le schéma bloc montré sur la Figure 2.5.

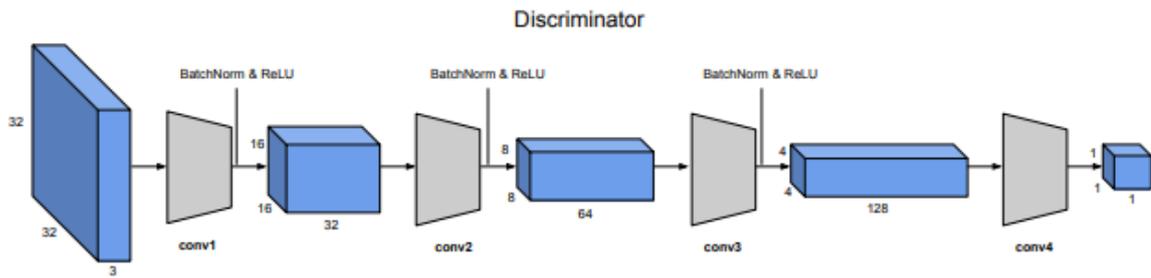


Figure 2.5 Architecture du discriminateur d'un DCGAN

Au niveau de chaque couche de convolution, un sous échantillonnage spatial par facteur de $s = 2$ est effectué. Des opérations de convolutions dans chaque couche sont calculées entre l'input et les noyaux de convolution. Supposant qu'on a des données d'entrées de taille $n_h \times n_w$ et un noyau de convolution (kernel) égale à $k_h \times k_w$. Alors, la taille du modèle de sortie (l'output shape) sera : $(n_h - k_h + 1) \times (n_w - k_w + 1)$. L'étape de padding est effectuée pour palier au problème de bords. Dans notre étude, nous avons considéré le cas de rajouter des zéros pour résoudre le problème de bord dans l'étape de convolution. La convolution est effectuée par des noyaux de convolution dans chaque couche. En général, si on ajoute p_h lignes (en haut et en bas de l'input) et p_w colonnes de padding (des zéros à gauche et à droite de l'input), la taille du modèle de l'output sera $(n_h - k_h + p_h + 1) \times (n_w - k_w + p_w + 1)$. Dans plusieurs cas, on prend $p_h = k_h - 1$ et $p_w = k_w - 1$.

L'étape de convolution est suivie de batch normalisation et la fonction d'activation ReLU. Rappelons que le batch normalisation consiste à

une technique de formation de réseau neuronal très approfondie qui normalise les entrées d'une couche pour chaque mini-batch. Cela a pour effet de stabiliser le processus d'apprentissage et de réduire considérablement le nombre d'époques de formation nécessaires pour former des réseaux profonds. Cela augmente la vitesse d'entraînement

Le **générateur** du DCGAN consiste en une séquence de couche de convolution qui sur échantillonnent progressivement l'input bruit pour générer l'image fausse (fake). La Figure 2.6 illustre l'architecture du générateur.

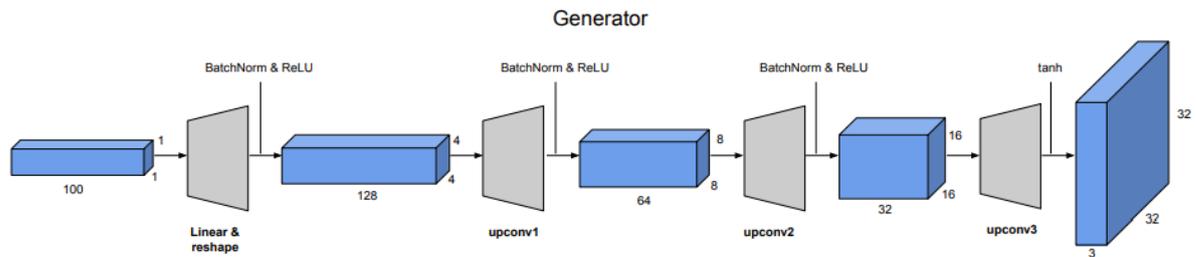


Figure 2.6 Architecture du générateur du DCGAN

Le bloc upconv consiste en une couche de sur-échantillonnage (par simple interpolation bilinéaire) suivie d'une convolution 2D.

L'architecture globale du DCGAN est donc représentée en regroupant le générateur et le discriminateur dans une seule structure comme c'est montré sur la Figure 2.7.

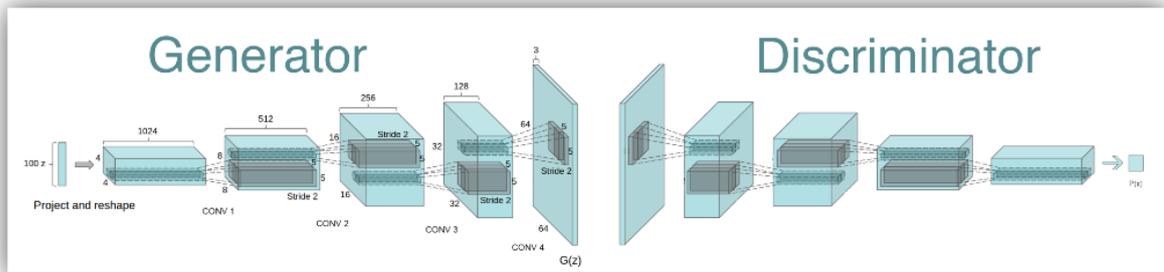


Figure 2.7 Architecture générale du DCGAN [12]

Différence entre GAN et DCGAN

DCGAN est un réseau Deep Convolutional Generative Adversarial qui utilise Deep Conv Nets pour avoir une architecture stable et de meilleurs résultats.

- Le générateur dans GAN utilise un réseau entièrement connecté, tandis que DCGAN utilise un réseau convolutif transposé pour suréchantillonner les images.
- Le générateur et le discriminateur n'utilisent pas de regroupement maximum.
- Le générateur et le discriminateur utilisent tous deux la normalisation par batch.
- Le générateur utilise la fonction Relu comme fonction d'activation pour toutes les couches sauf la sortie, et le discriminateur utilise Relu pour toutes les couches.

2.5 Réseau Cycle GAN

CycleGAN est un modèle conçu pour résoudre le problème de la traduction d'image à image.

Il est utilisé pour transférer des caractéristiques d'une image à une autre, ou peut mapper la distribution d'une image à une autre image. Dans CycleGAN, nous traitons le problème comme un problème de reconstruction d'image. Nous prenons d'abord une entrée d'image (x) et la convertissons en une image reconstruite à l'aide du générateur G . Nous inversons ensuite ce processus de l'image reconstruite à l'image d'origine à l'aide du générateur F , puis nous calculons la perte entre l'image réelle et l'image reconstruite. La caractéristique la plus importante de ce cycle_GAN est qu'il peut effectuer cette transformation d'image sur des images non appariées qui n'ont aucune relation entre l'image d'entrée et l'image de sortie. La Figure 2.8 illustre le concept du CycleGAN

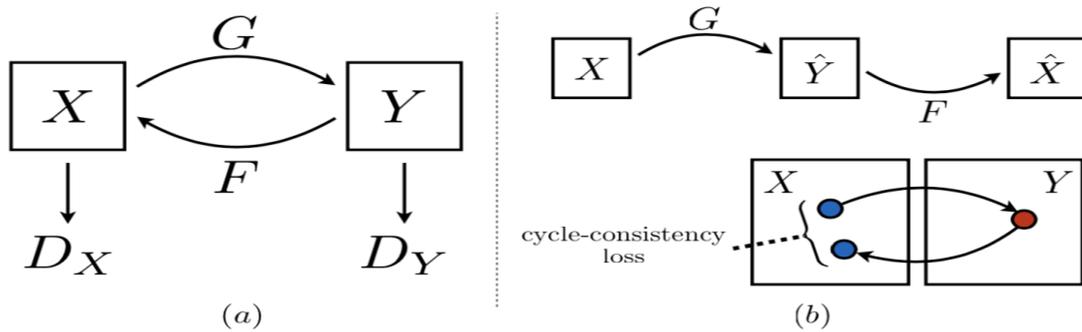


Figure 2.8 Principe du CycleGAN

2.6 Architecture de CycleGAN

Cycle Generative Adversarial Network (*CycleGAN*) sont un type de réseau de neurones basé sur des algorithmes d'apprentissage non supervisé.

Le CycleGAN utilise une extension supplémentaire à l'architecture appelée cohérence de cycle. qui implique l'apprentissage simultané de deux modèles générateurs et de deux modèles discriminateurs. Donc on a deux réseaux gan , Un générateur prend des images du premier domaine en entrée et produit des images pour le second domaine et l'autre générateur prend des images du second domaine en entrée et génère des images pour le premier domaine donc Le processus inverse devrait également être vrai le discriminateur donc dans les deux réseaux qui va juger les photos générées et essayer de deviner si elles sont réelles ou pas

Ces deux réseaux gan sont en compétition pendant le processus d'entraînement c'est ce qui leur permet d'améliorer leurs comportements respectifs La Figure 2.10 Montre l'architecture du CycleGAN.

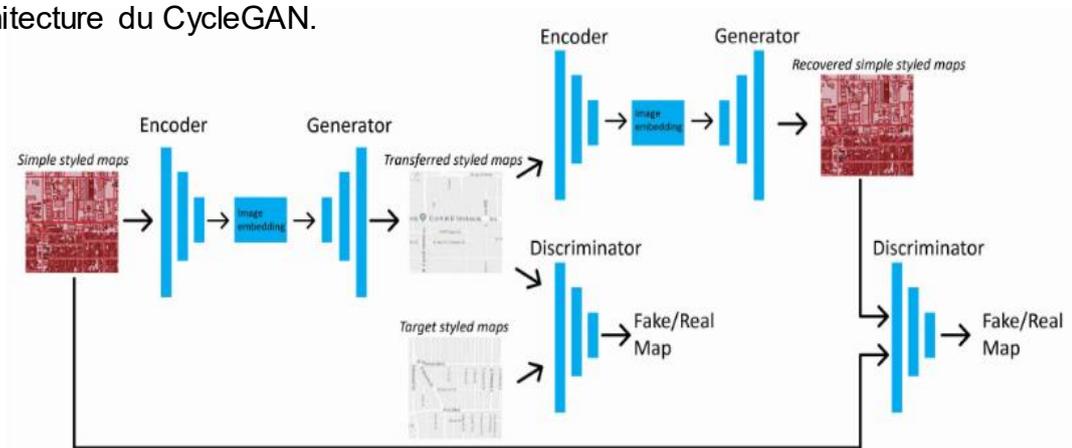


Figure 2.9 Architecture du CycleGAN

2.7 Critères d'évaluation

PSNR (Peak Signal to Noise Ratio)

est un indicateur de mesure de la qualité d'image, plus le PSNR est grand, meilleure est la qualité de l'image. le PSNR exprimé en décibels. est calculé comme suit :

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad 2.3$$

SSIM (Structural Similarity Index)

SSIM est créé pour déterminer plus précisément la différence entre deux images. La particularité est qu'il est toujours compris entre 0 et 1, lorsque sa valeur est égale à 1, cela signifie que nous avons deux images identiques. L'indice SSIM, est utilisé pour indiquer la similitude des informations de structure présentées dans les deux images comparées. Le SSIM des deux images A et B est défini comme suit

$$SSIM(A,B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)} \quad 2.4$$

Où μ est la valeur moyenne de l'intensité de l'image, σ est l'écart-type de l'image utilisé comme estimation du contraste des images A et B, C_1 et C_2 sont des constantes incluses pour assurer la stabilité des petits dénominateurs. Cependant, le SSIM est une approche de full-référence qui nécessite la présence d'une image de référence complète.

Ces deux critères vont être considérés dans notre étude de simulation

2.8 Conclusion

Dans ce chapitre, nous avons détaillé les réseaux GAN et ses deux variantes DCGAN et CycleGAN. Nous avons constaté que le principe du GAN est

essentiellement un apprentissage profond du générateur et du discriminateur à travers une base de données, pour obtenir un autre ensemble de données semblables. La différence entre le GAN et le DCGAN est la présence d'une couche de filtres de convolution dans DCGAN pour améliorer le mappage des images et pour réduire le volume de données par l'opération de sous échantillonnage et de convolution. Quand au CycleGAN, il dispose de deux générateurs et deux discriminateurs dont le principe et le rôle sont les mêmes que le réseau GAN. Les résultats de ces réseaux dans la génération d'ensemble de données, sont spectaculaires, on ne peut pas distinguer si les images générées font partie du monde réel, ou une simple simulation.

Dans notre étude, les réseaux DCGAN et CycleGAN seront implémenter pour débruiter et améliorer les images TEP à faible dose (low dose). Les résultats de simulation seront exposés et interprétés dans le chapitre 3.

Chapitre 3

Implémentation du GAN et Cycle GAN Et Application en Débruitage d'Images Médicales à Faible Dose

Résumé

Ce chapitre est entièrement dédié à la présentation de nos résultats de simulation. Nous implémentons les réseaux DCGAN et Cycle-GAN dans un contexte de débruitage d'images. Deux bases de données sont considérées : La base MNIST et la base Big-medical Data set. Deux critères d'évaluations sont calculés dans notre étude de simulation : PSNR et SSIM. Les résultats obtenus sont satisfaisants en terme de critères PSNR et SSIM mais également en terme de qualité visuelle.

Sommaire

3.1 Introduction

3.2 Nature de Données

3.3 Implémentation des Algorithmes

3.4 Résultats et Discussions

3.5 Conclusion

3.1 Introduction

Dans ce chapitre, nous présenterons les étapes d'implémentation des algorithmes DCGAN et CycleGAN, appliqués pour débruiter des images médicales à faible dose. Le bruit considéré dans les images, étant un bruit gaussien additif à différent niveaux. Nous détaillerons en premier lieu l'étape de préparation des données avant d'appliquer les algorithmes. Nous utiliserons deux bases principales de données : la base MNIST pour le réseau DCGAN et la base Big-medical data set pour le réseau CycleGAN. Plusieurs expériences de simulation sont menées pour tester les performances des deux réseaux et en changeant les paramètres des réseaux. Deux critères d'évaluation sont calculés : le PSNR et le SSIM plus la qualité visuelle des images obtenues. Les résultats de simulation sont présentés et interprétés à la fin de ce chapitre.

3.2 Nature des Données

3.2.1 Base MNIST

Pour l'implémentation du réseau DCGAN, nous avons utilisé la base MNIST [13]. C'est un ensemble de données de chiffres manuscrits. Elle contient 60 000 échantillons de formation et 10 000 échantillons d'essai. Chaque image est représentée par 28x28 pixels, chacun contenant une valeur 0 - 255 qui représente sa valeur en niveaux de gris. Notons que notre choix est justifié du moment que "MNIST" est une base de données de chiffres manuscrits largement utilisés dans le Machine Learning, notamment pour la formation et le test de modèles prédictifs.

3.2.2 Base Big-Medical-Data-set [14]

Pour l'implémentation du réseau CycleGAN, nous avons en premier lieu télécharger la base big-medical-data set qui contient plus de 55600 images. Nous avons ensuite sélectionné plus de 200 images pour notre étude de simulation. Les images sont réparties sous deux classes A et B :

- Classe A : contient les images full dose
- Classe B : contient les images faible dose (low dose).

Les images sont organisées sous forme de dossier (A,B) où chaque dossier contient 50 images.

Les images téléchargées sont sous format DICOM, donc nous les avons d'abord converti en format PNG et de taille 255x255pixels, avant de les traiter par le réseau CycleGAN.

Nous avons pris 80% de l'ensemble d'image pour l'apprentissage du réseau et 20% pour le test du réseau. Ce choix de pourcentage est généralement utilisé en deep learning.

3.3 Présentation des outils de développement

Le matériel utilisé est un PC personnel Lenovo avec un processeur Intel® core™ i3 6th gen, une RAM de 6 Gb ,Windows 10 Pro (64 bit). Les réseaux sont réalisés par le logiciel de programmation python version 3.8 sous l'environnement Anaconda sous le Navigator Spyder. De plus nous avons utilisé Google Colab qui permet d'écrire et d'exécuter le code et d'exploiter le GPU surtout que nous avons des big data. Nous nous sommes basés sur les bibliothèques et les packages de python surtout Kera et Tensorflow qui sont destinés et développés pour le deep learning.

3.4 Implémentation du Réseau DCGAN

Les étapes de simulation de cette partie sont illustrées dans l'organigramme de la Figure 3.5.

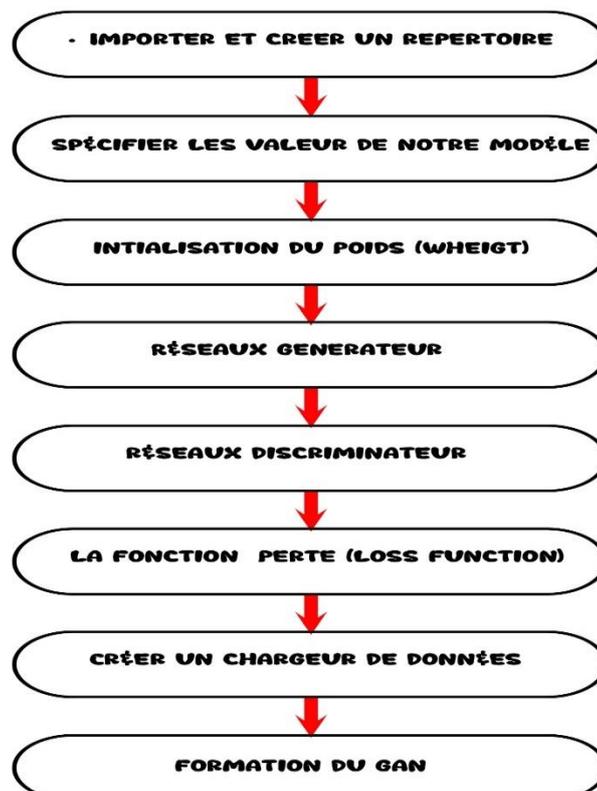


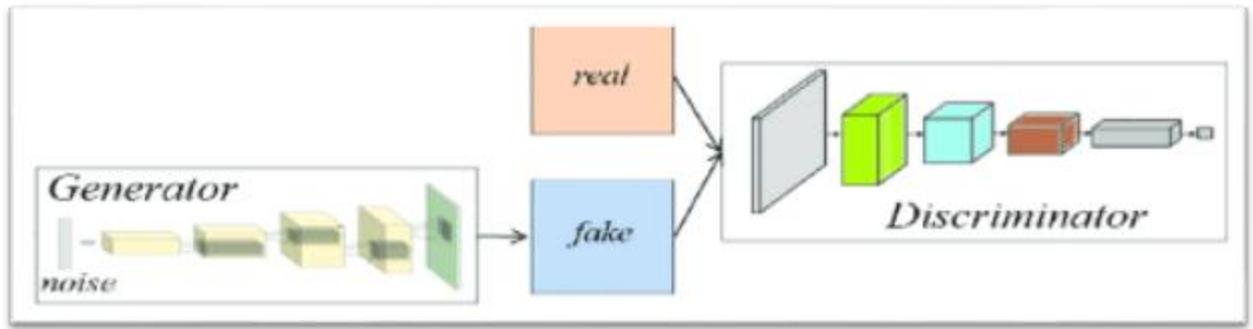
Figure 3.5 Etapes d'implémentation du réseau DCGAN

Nous avons réalisé l'implémentation du réseau DCGAN sous le logiciel Python et Pytorch.

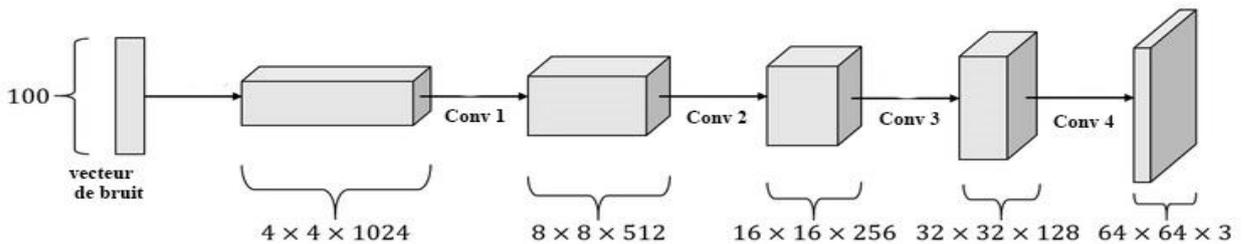
Nous résumons les étapes d'exécution comme suit :

- *Importer et créer un répertoire et installer et gérer les "packages/libraries" du Python (pytorch).*
- *Spécifier Les valeurs de notre modèle : nous avons fixé le nombre d'épochs à 200, la taille de batch à 64, les dimensions des images à 32×32. L'optimiseur choisi est ADAM (learning rate= 0.0002)..*
- *Initialisation des poids (weights) : Applique les "weights" initiaux à certaines couches du modèle.*
- *Réseau Générateur (Generator network) : Définit le réseau du générateur : on a remplacé pooling layers par fractional-strided convolutions, Utilisez l'activation LeakyReLU dans le générateur pour toutes les couches à l'exception de la sortie, qui utilise Tanh.*
- *Réseau discriminateur (Discriminateurs network): Définit le réseau du discriminateur : on a remplacé pooling layers par strided convolutions, Utilisez l'activation LeakyReLU dans le discriminateur pour toutes les couches.*
- *La fonction perte (Loss function) et Paramètre réseau : Définir la fonction de perte, initialiser les "weights", initialiser le générateur/discrimination et définir l'optimiseur.*
- *Créer Un dataloader : créer un chargeur de données , nous allons générer une image en utilisant la base MNIST*
- *Formation Du GAN : L'entrainement du GAN.*

Les DCGAN utilisent les principes de base des CNN. Mais, dans le réseau il y a seulement la couche des masques de convolution (Convolution Layer) il n'y a pas de couche de pooling (pooling layer). La Figure 3.6 illustre l'architecture du réseau DCGAN que nous avons implémenté.



(a) Architecture du réseau DCGAN



(b) Générateur Bloc du DCGAN

Figure 3.6 Réseau DCGAN

Tout d'abord, nous donnons au générateur un vecteur de bruit à 100 dimensions en entrée. Après cela, nous projetons et transformons l'entrée. Ensuite, le réseau effectue quatre opérations de convolution. Chaque fois, nous obtenons un incrément de hauteur et de largeur. Au même temps, le canal ne cesse de se rétrécir. Après la première opération de convolution, nous avons 512 canaux de sortie. Celle-ci continue de diminuer à chaque opération de convolution. Après la troisième convolution, le canal de sortie est 128. Enfin, nous générons des images de taille 64×64 et trois canaux de sortie. Toutes les couches sauf la première couche convolutive ont une division par 2. Les détails du réseaux sont :

- utiliser batchnormalisation à la fois dans le générateur et le discriminateur,
- Utiliser l'activation ReLU dans le générateur pour toutes les couches sauf pour la sortie, qui utilise Tla fonction d'activation anh.,
- Utiliser l'activation LeakyReLU dans le discriminateur pour toutes les couches.
- Tous les modèles ont été entraînés avec une descente de gradient stochastique en mini-batch (SGD) avec une taille de mini-batch de 128.

Notre première expérience de simulation est d'obtenir une image de chiffres lisibles à partir d'une image bruit. Le bruit est un bruit gaussien de moyenne nulle et variance 1. La Figure 3.7.a illustre l'image input.

Nous introduisons dans le réseau DCGAN l'image bruit et nous fixons tous les paramètres.

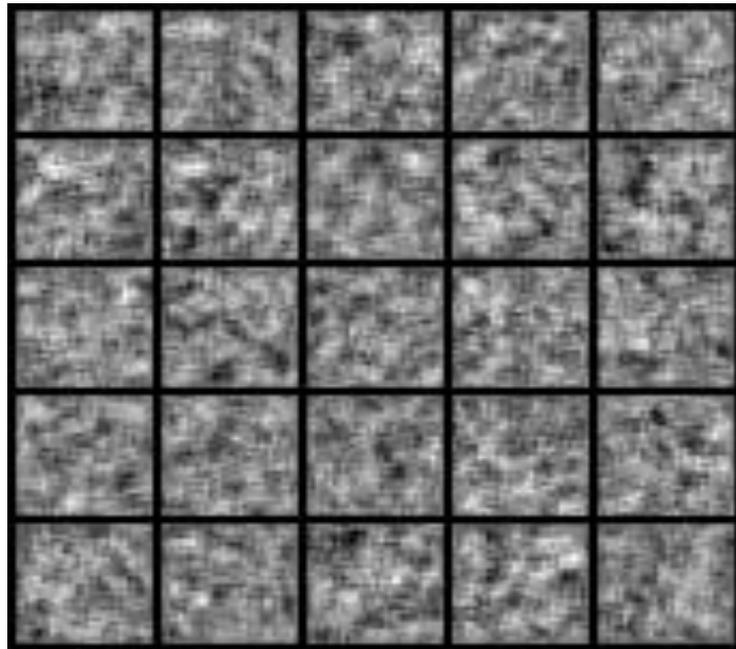


Figure 3.7.a Image input de chiffres bruit du DCGAN

La Figure 3.7.b illustre les détails du réseau DCGAN ainsi que les étapes intermédiaires d'exécution du DCGAN.

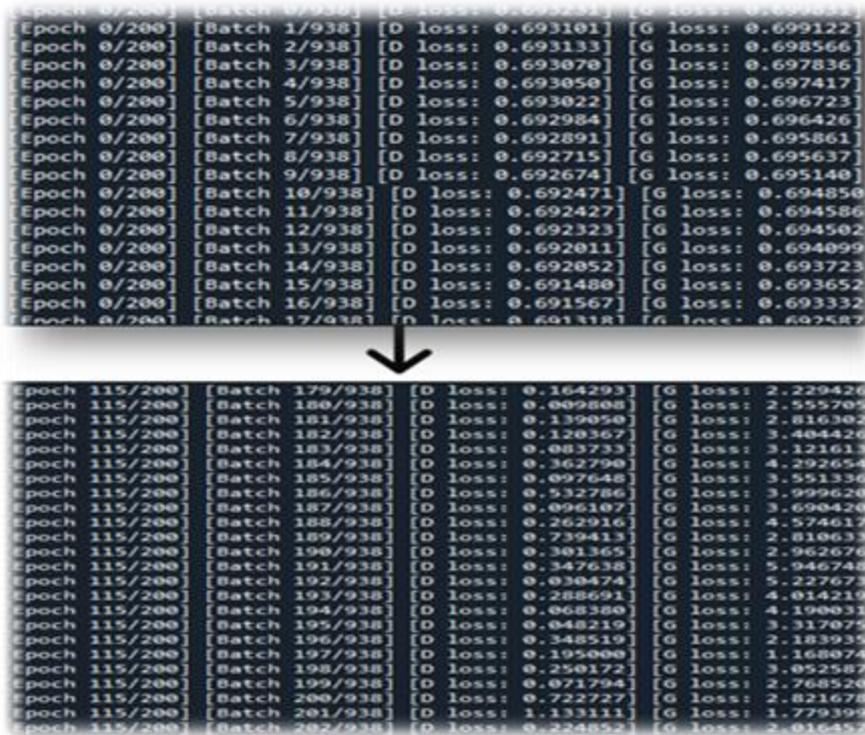


Figure 3.7.b Déroulement du programme DCGAN

Nous visualisons clairement sur la Figure 3.7.b le détail des paramètres du réseau à savoir, le déroulement du nombre d'epoch jusqu'à 200, le nombre de batch correspondant 1000, les valeurs des fonctions loss du discriminateur et du générateur *ED loss* et *EG loss* respectivement. Nous avons noté que la minimisation de ces deux fonctions est très rapide pour les premiers epochs de 1 à 115 mais fluctue légèrement après 115 epoch effectués jusqu'à la stabilisation à 200 epochs effectués. De ce fait, nous avons fixé le nombre maximum d'epochs à 200.

La Figure 3.8 illustre les premières images obtenues par le DCGAN, à chaque nombre de batch effectué. Il est clair que les chiffres dans les images commencent à être lisible à partir de 3600 batch effectué. Pour 12 batch effectué, les chiffres restent illisibles. Au-delà de 108000 batch effectué, la qualité des images obtenue est la même et il n'y a pas d'amélioration. Ce résultat est satisfaisant.

batches faits	Images Débruitées
400	
3600	
62400	
108000	

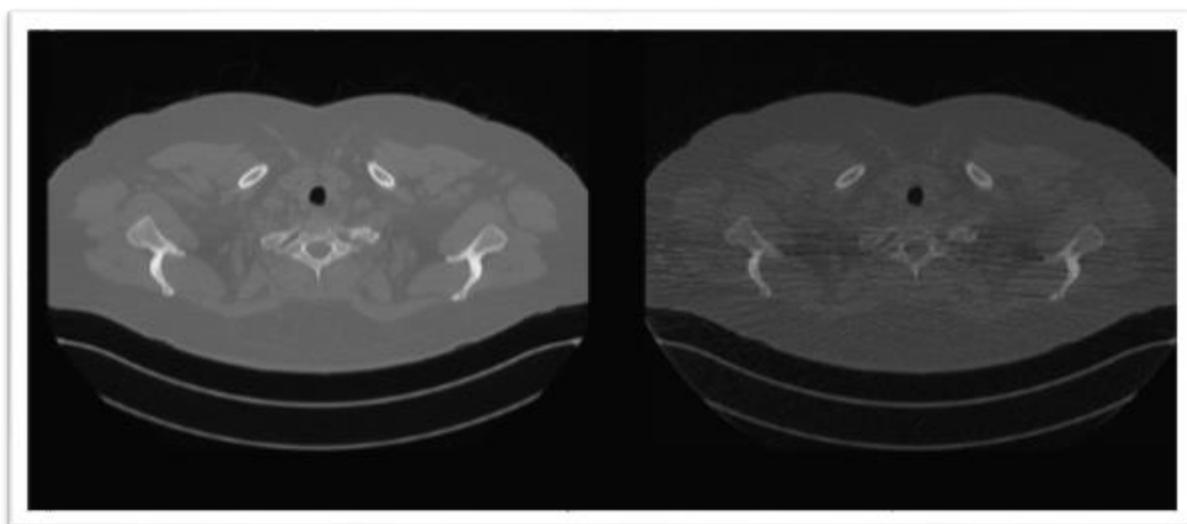
Figure 3.8 Images débruitées par DCGAN à différents nombres de batchs

Cela montre que notre générateur a appris assez bien et a pu produire de très bonnes images en termes de qualité visuelle qui semblent avoir été extraites directement des données MNIST.

3.5 Implémentation du réseau CycleGAN

Dans cette partie, nous utilisons la base Big data medical data set, où nous avons les images low dose et full dose. L'objectif est de montrer les performances du réseau CycleGAN dans l'application de débruitage de ce type d'images. Rappelons que la modalité de ces images est TEP, comme nous l'avons mentionné au chapitre 2. La Figure 3.9 illustre la paire d'image TEP full et low dose respectivement. Le compromis dans ces applications est la diminution de la dose de traceur radioactif pour protéger le patient, mais sans détériorer la qualité d'image obtenue.

Le principe du réseau CycleGAN est semblable au principe du réseau DCGAN. Seulement, le CycleGAN dispose de deux générateurs et deux discriminateurs. La Figure 3.10 présente l'architecture de CycleGAN que nous avons adopté et implémenté pour améliorer les images TEP.



(a)

(b)

Figure 3.9 Image TEP (a) Full dose et (b) Low dose images

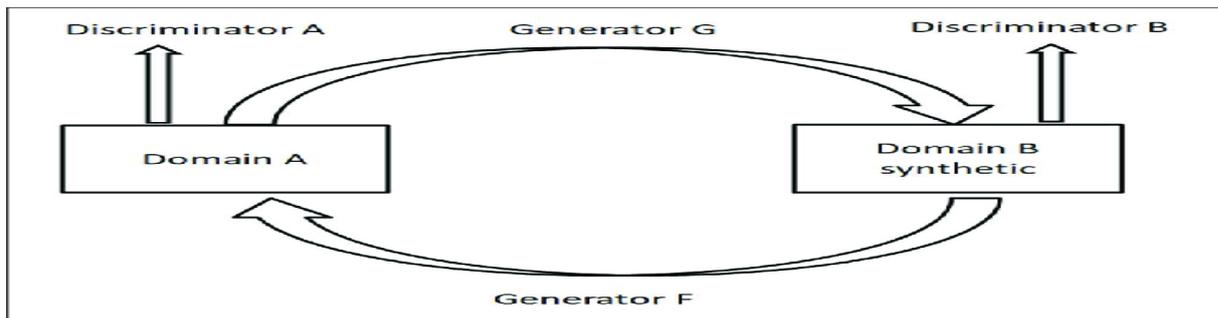


Figure 3.10 Architecture de CycleGAN appliqué en débruitage

Dans ce modèle CycleGAN, nous avons deux réseaux de GAN. Le générateur du premier réseau GAN prendra l'image originale (full dose) et génère une image 'low dose' (image générée) à partir de celle full dose.

Le générateur du deuxième réseau GAN prendra cette image générée (low dose) et doit générer une image 'full dose' et c'est l'objectif de l'application, obtenir des images full dose à partir d'images low dose. Nous avons deux discriminateurs dans les deux réseaux GAN. Le rôle des discriminateurs consiste à détecter si l'image obtenue est réelle (originale) ou non (générée).

Les modèles générateurs seront mis à jour en fonction de la réduction de la différence entre l'entrée du GAN 1 et la sortie du GAN 2, nous utilisons donc le 'cycle consistency loss' qui calcule la différence entre l'image input du GAN1 et l'image output du GAN2 et le modèle du générateur est actualisé en terme de réduction de la différence des deux images. La Figure 3.11 illustre les étapes d'implémentation du CycleGAN pour l'obtention des images full dose à partir d'images low dose.

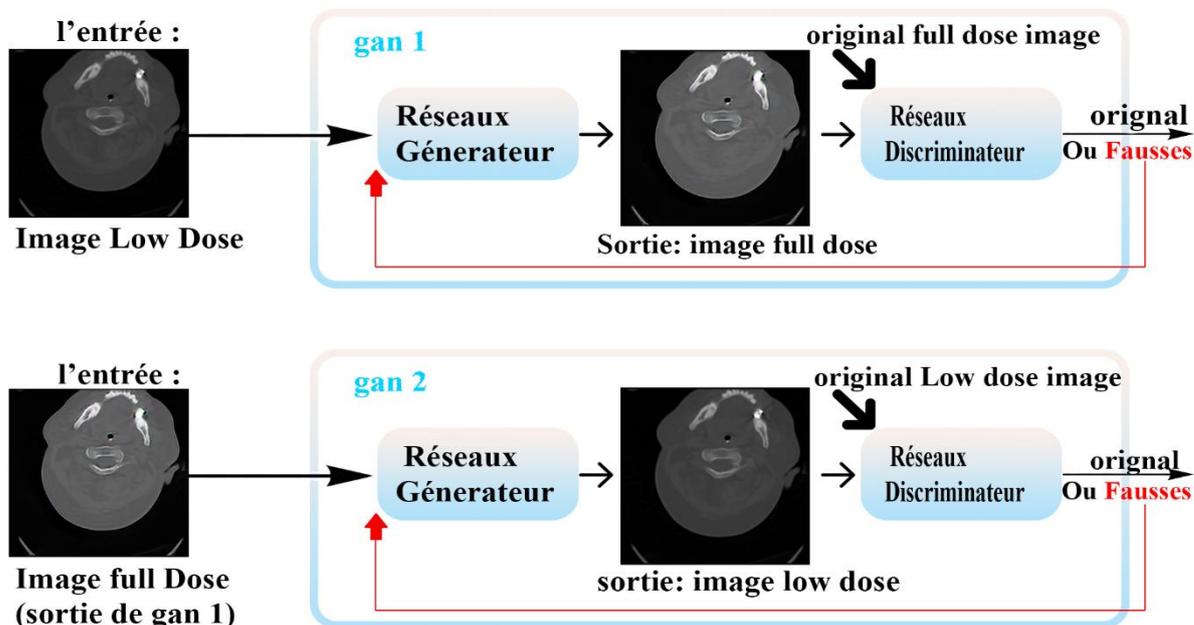
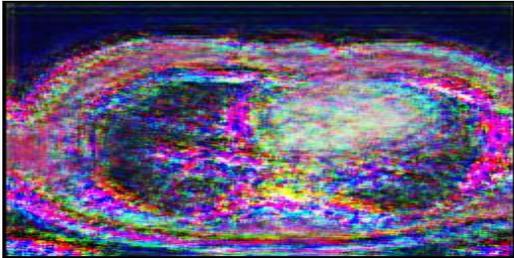
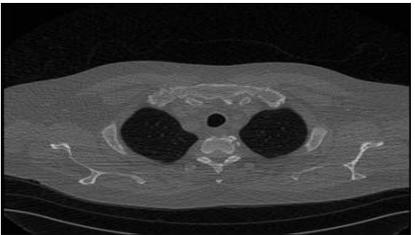
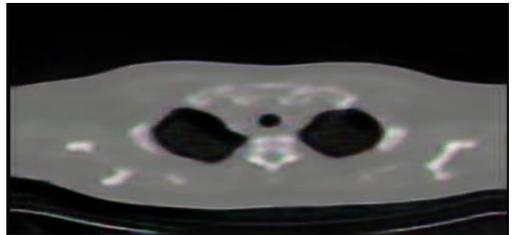
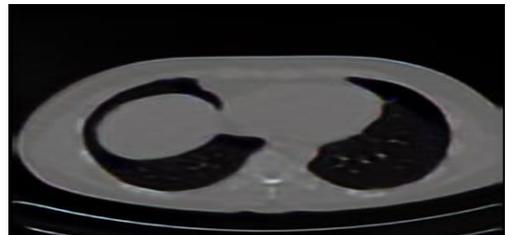
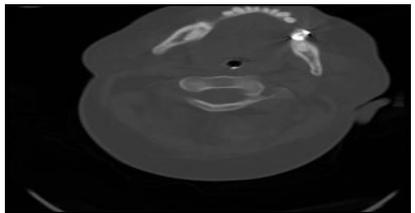
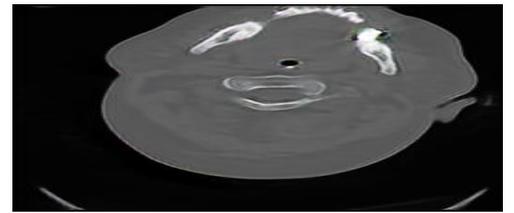
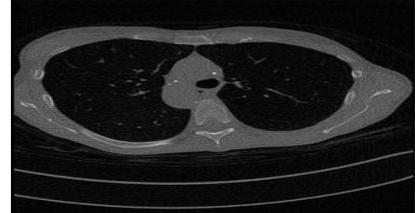
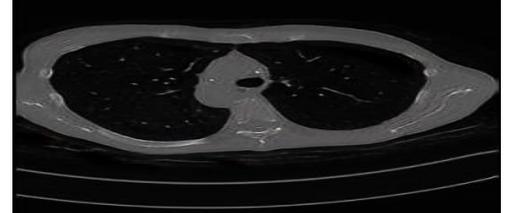


Figure 3.11 Organigramme de fonctionnement du CycleGAN

Les résultats obtenus en implémentant le CycleGAN sont représentés dans la Figure 3.12. La première ligne des images représente l'image originale et l'image générée LD respectivement. Nous avons présenté les images obtenues en fonction du nombre de batchs pour mieux distinguer l'évolution de l'amélioration de la qualité des images. Il y a une amélioration considérable pour un nombre de batch égale à 6450. Cette amélioration est argumentée par les valeurs des PSNR et SSIM représentées dans Tableau 3.2. Au-delà de cette valeur, l'amélioration des résultats en termes de qualité visuelle, PSNR et SSIM est négligeable.

batc h faits	Image original	Image Générer
0		
200		
1000		
3020		
5000		

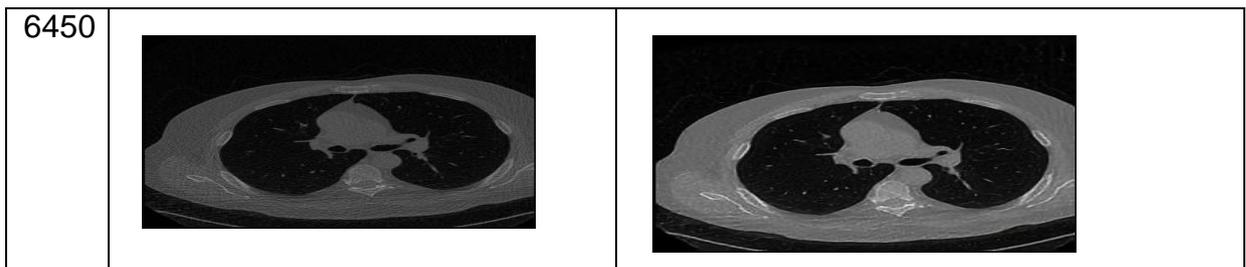


Figure 3.12 Résultat des images reconstruites obtenues par CycleGAN à partir des images LD

Tableau 3.2 Critères d'évaluation PSNR & SSIM en fonction de nombre de batchs effectués

Nombre de batchs	PSNR	SSIM
0	28.12dB	0.06
200	29.06dB	0.47
1000	30.26dB	0.50
3020	30.66dB	0.81
5000	32.06dB	0.70
6450	30.30dB	0.66

Pour mieux démontrer les performances du réseau CycleGAN, nous avons calculé les fonction loss du générateur et du discriminateur en fonction de nombre d'epochs. La Figure 3.13 montre bien la diminution de ces fonction en augmentant le nombre d'epochs, plus le nombre d'epochs est grand, moins sont les fonctions loss du générateur et du discriminateur, et cela confirme l'apprentissage du réseau.

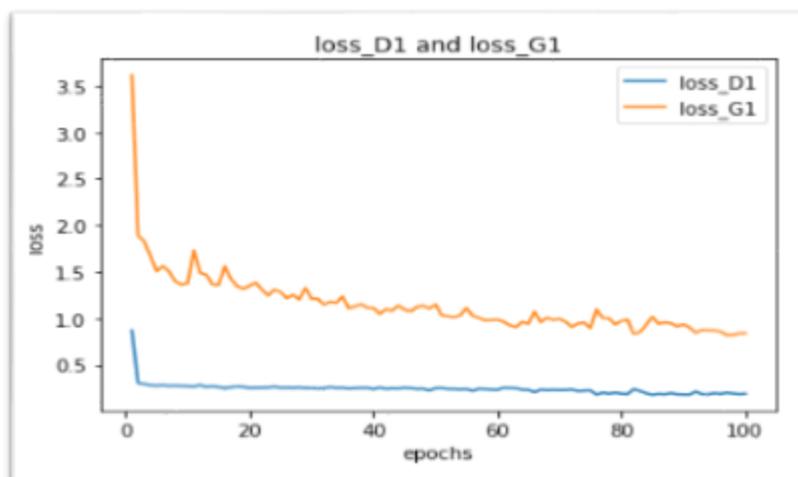


Figure 3.13 Fonctions loss du générateur et du discriminateur en fonction du nombre d'epochs.

3.6 Conclusion :

Dans ce chapitre, nous avons exposé nos résultats d'implémentation des réseaux DCGAN et CycleCGAN. L'objectif pour DCGAN était d'obtenir des images avec des chiffres lisibles à partir d'images bruitées. Le bruit que nous avons considéré est un bruit gaussien additif de moyenne nulle et de variance variable. La base utilisée est la base MNIST. L'objectif pour CycleGAN, était d'obtenir des images médicales full dose à partir d'images low dose. La base utilisée dans cette partie était la base Big medical data set. La première étape dans notre implémentation était de préparer l'ensemble d'images d'entraînement. Les images sont regroupées sous formes de paires (full dose, low dose). Les données sont ensuite converties du format DICOM au format PNG. La deuxième étape était de générer d'abord les images low dose, puis appliquer le CycleGAN pour l'obtention des images full dose. A partir des résultats de simulation et en augmentant le nombre d'epochs, nous avons pu aboutir à des images full dose nettement meilleurs en termes de qualité visuelle et en terme de deux critères de performances : PSNR et SSIM. Le CycleGAN a surpassé les méthodes conventionnelles de filtrage.

Conclusion Générale :

Dans ce travail de Master nous nous sommes intéressés à l'étude du problème de débruitage des images dans un contexte d'apprentissage profond. Plus précisément, nous intéressons aux images médicales type TEP à faible dose (low dose LD). Nous avons implémenté un type de réseaux de neurones, qui est le réseau antagoniste génératif (generative adversarial network GAN). Nous avons implémenté tout d'abord un des types de GAN qui est le deep convolutional GAN (DCGAN). Nous avons appliqué le DCGAN sur la base MNIST pour tester l'apprentissage du réseau. Rappelons que DCGAN se base essentiellement sur une structure de réseaux convolutifs (CNN). Puis Nous avons implémenté le réseau CycleGAN qui se compose de deux générateurs et deux discriminateurs. Ce dernier est appliqué pour améliorer des images médicales TEP à faible dose (LD). Dans cette partie nous avons utilisé la base de données Big medical data set. Une étape de préparation de données a été effectuée tel que la conversion du format des images du Dicom au format PNG. Les images sont regroupées en paires (full dose, low dose). Puis nous avons précisé tous les paramètres du CycleGAN.

Pour une évaluation objective des performances du réseau, nous nous sommes appuyés sur le calcul des critères PSNR et SSIM qui ont montré la supériorité du CycleGAN. Ce dernier réseau a permis une amélioration considérable des images LD. Les résultats de simulation ont montré l'efficacité du CycleGAN. Bien que l'objectif de cette étude, qui l'exploitation d'une structure deep learning pour le débruitage d'images, il reste encore plusieurs pistes à élaborer et à étudier tel que le développement de nouvelles architectures du GAN.

Références

- [1] Sandhya Pati/2019/Artificial Intelligence Course in Bangalore - AI Training & Certification <https://morioh.com/p/7ccb1d0d114d>
- [2] Copeland, B.J.. "Artificial intelligence". Encyclopedia Britannica, 18 Mar. 2022, <https://www.britannica.com/technology/artificial-intelligence>. Accessed 27 April 2022.
- [3] Derek Kwok/May 25, 2018/Machine Learning for My Grandma
medium.com/@dkwok94/machine-learning-for-my-grandma-ca242e97ef62
- [4] *talks.navixy/ May 07, 2020/Deep Learning in Telematics*
<https://talks.navixy.com/reviews/deep-learning-in-telematics/>
- [5] Farhad Malik /May 18, 2019/Neural Networks Bias And Weights
<https://medium.com/fintechexplained/neural-networks-bias-and-weights-1>
- [6] Acervo Lima/septembre 2021/GBlog, Machine Learning, Technical
Scripter/Régression et classification | Apprentissage automatique supervisé
<https://fr.acervolima.com/regression-et-classification-apprentissage-automatique-supervise/>
- [7] Jason Brownlee, “ Logistic Regression for Machine Learning”, on April 1, 2016 in
Machine Learning Algorithms
- [8] Gradient Descent and Cost Function à partir de zéro en utilisant Python /karthik kulkarni
- [9] Institut Supérieur d’Informatique Université Tunis Elmanar/Cours : Traitement d
images ’images/présenté par:Mohamed Sahbi Bahroun.
- [10] A Beginner's Guide to Generative Adversarial Networks (GANs)/
wiki.pathmind.com/generative-adversarial-network-GAN
- [11] <https://iq.opengenus.org/content/images/2019/08/1-8.PNG>

