الجمهورية الجزائرية الديمقراطية الشعبية

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



FACULTÉ DE LA TECHNOLOGIE

DU DÉPARTEMENT DU GENIE ÉLECTRIQUE

Mémoire de fin d'études de Master

Filière : Télécommunications

Spécialité : Systèmes de télécommunications

Débruitage des images basées sur l'apprentissage profond

Préparé et présenté par

Berraf Issam Abdelwahab

Titri Mohamed Mounir

Proposé et dirigé par

M. Boucherit Ismail

Promotion : 2022

Résumé

موجز

الهدف الرئيسي من هذا المشروع هو بناء نظام ذكاء اصطناعي قادر على توفير صور طبية ذات نوعية جيدة للخبراء في المجال الطبي حتى يتمكنوا من تشخيص الحالات الحرجة بسهولة، خاصة بعد الأوبئة الضخمة التي حدثت في السنوات الأخيرة مثل سارس كوف-2 (كوفيد-19) والعديد من الأوبئة الأخرى التي كلفت الكثير من الأرواح البشرية.

سنستخدم نظام الشبكة العصبية التلافيفية لمعالجة الصور وتقليل تشويش مجموعة بيانات الأشعة السينية للصدر لمرض كوفيد-19.

الكلمات المفتاحية: الصور الطبية، الأشعة السينية، التعلم العميق، الترميز التلقائي، الشبكات العصبية، سي إن إن، تقليل الضوضاء، بايثون.

RÉSUMÉ

L’objectif principal de ce projet est de construire un système d'intelligence artificielle susceptible de fournir des images médicales de bonne qualité aux experts du domaine médical afin qu'ils puissent facilement poser leur diagnostic sur des cas critiques notamment après les pandémies massives qui ont eu lieu ces dernières années telles que SARS-CoV-2 (COVID-19) et bien d'autres qui ont coûté tant de vies humaines.

Nous utiliserons un système de réseau de neurones convolutifs (CNN) pour le traitement d'images et le débruitage d'un ensemble de données de radiographie pulmonaire (rayons X) de la maladie covid-19.

Mots clés : Imagerie médicale, rayons X, apprentissage en profondeur, auto-encodeur, réseaux de neurones, CNN, débruitage, python.

ABSTRACT

The main objective from this project is to build an artificial intelligence system could potentially provide good quality medical images for the experts in the medical field so they can easily make their diagnosis on critical cases especially after the massive pandemics that which took place in the few past years such as SARS-CoV-2 (COVID-19) and many others which took so many human lives.

We will use a convolutional neural network (CNN) system for image processing and denoising of a chest x-ray dataset of the covid-19 disease.

Key words: Medical imagery, x-ray, deep learning, autoencoders, neural networks, CNN, denoising, python.

Liste des acronymes et abréviations

AE : AutoEncoder

AI : Artificial Intelligence

ANN : Artificial Neural Network

AWGN : Additive white Gaussian noise

CNN / **ConvNet** : convolutional neural network

COVID-19 : coronavirus disease of 2019

CT : computed tomography

CXR : Chest x-ray

DAE : Denoising AutoEncoder / Deep AutoEncoder

DL : Deep Learning

FC : Fully Connected

FCN : Fully Convolutional Network

GUI : Graphical User Interface

ML : Machine Learning

MRI : Magnetic resonance imaging

MSE : Mean squared error

PSNR : Peak signal-to-noise ratio

RNN : Recurrent Neural Network

ReLU : Rectified Linear Unit

SARS-CoV-2 : severe acute respiratory syndrome coronavirus 2

SNR : Singal-to-noise ratio

Remerciement et dédicace

Tout d'abord, nous voudrions remercier Allah de nous avoir guidés pour prendre les bonnes décisions et atteindre ce niveau d'éducation, un grand merci à notre encadreur ***M. Boucherit*** *pour son aide tout au long du chemin, merci à vous monsieur.*

Merci à tous les enseignants d'avoir tout donné pour transmettre le message d'apprentissage, qu'Allah récompense vos efforts.

Merci à nos chers parents d'avoir cru en nous et d'avoir fourni tout ce qu'ils pouvaient pour que nous puissions avoir une part d'apprentissage, qu'Allah vous bénisse

Et enfin, nous voudrions dédier ce modeste travail à toutes nos familles et collègues ainsi qu'aux enseignants, merci à tous pour les précieux souvenirs, nous souhaitons à tous une vie joyeuse remplie de succès et de bonheur insha'allah.

Sommaire

INTRODUCTION GENERALE 1

Chapitre I : L'imagerie médicale

I. Introduction 4
1 L'histoire de l'imagerie médicale 4
2 Différentes techniques d'imagerie médicale 5
3 Imagerie par rayon X 5
3.1 Les rayons X 5
3.2 Comment produire une radiographie ? 6
3.3 Quand les radiographies médicales sont-elles utiles ? 7
3.3.1 Diagnostique 7
3.3.1.1 La radiographie aux rayons X 7
3.3.1.2 Mammographie 8
3.3.1.3 CT (tomodensitométrie) 9
3.3.1.4 Fluoroscopie 9
3.3.2 Thérapeutique 10
3.3.2.1 Radiothérapie dans le traitement du cancer 10
4 Le bruit dans l'imagerie médicale 10
4.1 Les modèles de bruit dans l'image 10
4.1.1 Le bruit additif 10
4.1.2 Le bruit multiplicatif 11
4.1.3 Le bruit convolutif 11
4.2 Les types de bruit 11
4.2.1 Bruit sel et poivre 11
4.2.2 Bruit blanc Gaussien (AWGN : additive white Gaussien noise) 12
4.2.3 Le flou 13
4.2.4 Bruit multiplicatif Poisson (Poisson noise, shot noise) 14
5 Débruitage 14
5.1 Les critères de choix d'un filtre (sous-titre débruitage) 14
5.1.1 Erreur moyenne quadratique (MSE) 15

5.1.2 Erreur signal sur bruit SNR (Signal to Noise Ratio) 15
5.1.3 Rapport signal sur bruit crête (PSNR) 15
6 Filtrage 15
6.1 Filtre médian (sous-titre filtrage) 15
Conclusion 17

Chapitre II : Apprentissage profond

I. Introduction 19
II. Historique 20
1 Réseaux de neurones 21
2 Réseaux de neurones convolutifs 22
3 Les couches de réseaux de neurones convolutionnels 22
3.1 Couche d'entrée du CNN 23
3.2 Couche de convolution 23
3.3 Couche de mise en commun (Pooling) 25
3.4 Couche entièrement connecté (Fully Connected) 25
3.5 Couche de perte 26
3.6 Couche de sortie (output layer) 26
4 Auto-encodeur 26
5 Opérations de couches convolutives 28
6 Logiciel utilisé dans l'apprentissage profond 28
6.1 Python 28
6.1.1 Bibliothèques utilisées dans ce projet (Python) 28
6.1.1.1.1 OpenCV (Open-Source Computer Vision Library) 28
6.1.1.1.2 Matplotlib 28
6.1.1.1.3 Numpy (Numerical Python) 29
6.1.1.1.4 Keras 29
6.1.1.1.5 Tensorflow 29
III. Conclusion 29

Chapitre III : Simulation et résultats

I. Introduction 31
II. Plan de travail 31
1. Chargement de l'ensemble de données 32
2. Prétraitement des images 33
3. Addition de bruit 33
4. Training, validation, et test de la base de données 34
5. Résultats 35
6. Evaluation 38
6.1 Première manipulation 38
6.2 Deuxième manipulation 38
III. Conclusion 38
Conclusion générale et perspectives 40

Liste des figures

FIGURE 1. UNE PHOTO D'UN MEDECIN TENANT UNE RADIOGRAPHIE. 6

FIGURE 2. UNE ILLUSTRATION D'UN PATIENT QUI PASSE UNE RADIOGRAPHIE THORACIQUE. 7

FIGURE 3. ACQUISITION DE RADIOGRAPHIE PAR PROJECTION MEDICALE. 8

FIGURE 4. UNE ILLUSTRATION QUI MONTRE LA MAMMOGRAPHIE. 9

FIGURE 5. IMAGE DU BRUIT POIVRE ET SEL (A GAUCHE). 12

FIGURE 6. COURBE DE DENSITE DE PROBABILITE D'EQUATION. 12

FIGURE 7. L'IMAGE (A) A UN BRUIT BLANC ADDITIF. 13

FIGURE 8. IMAGE FLOUE A GAUCHE. 13

FIGURE 9. IMAGE DEGRADEE PAR LE BRUIT DE POISSON. 14

FIGURE 10. PRINCIPE DU FILTRE MEDIAN 3X3. 16

FIGURE 11. SCHEMA D'UN RESEAU DE NEURONES ARTIFICIEL 21

FIGURE 12. EXEMPLE D'UN RESEAU CNN POUR LE DEBRUITAGE 22

FIGURE 13. EXEMPLE D'ARCHITECTURE CNN 23

FIGURE 14. EXEMPLE DE PRINCIPE DU FILTRE CONVOLUTIONNEL 24

FIGURE 15. PRINCIPE DE FONCTION RELU 24

FIGURE 16. EXEMPLE DE PRINCIPE DE POOLING. 25

FIGURE 17. PRINCIPE DE LA COUCHE ENTIEREMENT CONNECTEE. 25

FIGURE 18. ARCHITECTURE D'UN AUTOENCODEUR 26

FIGURE 19. ARCHITECTURE D'UN AUTOENCODEUR 27

FIGURE 20. OPERATIONS CONV/DECONV RELU ET MAXPOOLING 27

FIGURE 21. UN ORGANIGRAMME QUI REPRESENTE LE PLAN DE TRAVAIL POUR LA SIMULATION 31

FIGURE 22. APERÇU DE L'ENSEMBLE DE DONNEES 32

FIGURE 23. IMAGES DES RESULTATS DE MANIPULATION 1 DU MODELE CNN 36

FIGURE 24. UN ECHANTILLON QUI MONTRE LES DEUX METHODES DE DEBRUITAGE UTILISEES 36

FIGURE 25. IMAGES DES RESULTATS DE MANIPULATION 2 DU MODELE CNN 37

Liste des tableaux

TABLEAU 1. LES VALEURS DE CRITERE PSNR POUR LA 1^{ERE} MANIPULATION.....35
TABLEAU 2. LES VALEURS DE CRITERE MSE POUR LA 1^{ERE} MANIPULATION 35
TABLEAU 3. LES VALEURS DE CRITERE PSNR POUR LA 2EME MANIPULATION.....37
TABLEAU 4. LES VALEURS DE CRITERE MSE POUR LA 2EME MANIPULATION.....37

Introduction

Générale

INTRODUCTION GENERALE

L'imagerie médicale regroupe l'ensemble des techniques utilisées par la médecine pour le diagnostic mais aussi le traitement d'un grand nombre de pathologies. Elle a révolutionné la médecine en donnant un accès immédiat et fiable à des informations jusqu'alors « invisibles » au diagnostic clinique, comme par exemple aux caractéristiques anatomiques, voire même à certains aspects du métabolisme (imagerie fonctionnelle) des organes. Les techniques d'imagerie médicale ne donnent pas une simple « photographie » du tissu ou de l'organe étudié mais une représentation visuelle fondée sur des caractéristiques physiques ou chimiques particulières. Avec un appareillage certes beaucoup plus lourd que les instruments d'optique (endoscopes, fibres optiques...) - aujourd'hui si petits qu'ils peuvent être introduits dans les étroits canaux du corps humain -, mais sans les contraintes que ces derniers imposent (hospitalisation, anesthésie...). Les appareillages utilisés sont aussi variés que les techniques elles - mêmes ; des techniques qui peuvent être complémentaires les unes des autres. [1]

Dans ce mémoire, nous allons construire une application à la méthode fondée par CHUA et YANG qui s'appelle les réseaux de neurones, et cette application est destinée à travailler sur le débruitage des images médicales, plus précisément des radiographies thoraciques. [2]

Notre mémoire comporte 3 chapitres :

Dans le premier chapitre :

- Se familiariser avec l'imagerie médicale.
- Découvrir les différents types d'images médicales.
- Connaître les facteurs influençant sur l'image médicale.
- Savoir quels filtres sont utilisés pour améliorer la qualité de l'image

Dans le deuxième chapitre :

Nous présenterons la théorie des réseaux de neurones cellulaires (CNN) en se concentrant sur :

- Rencontrez et découvrez le concept de réseaux de neurones et son histoire d'apparition.
- Connaître la structure d'un système CNN et les différentes couches qui les forment.

- Comprendre comment fonctionne l'auto-encodeur.
- Se familiariser avec le logiciel utilisé pour simuler et ses bibliothèques.

Et dans le troisième chapitre, nous procéderons aux simulations, varierons et manipulerons le système, jusqu'à ce que nous obtenions des résultats que nous interpréterons à la fin.

Et à la fin du mémoire, nous terminons par une conclusion générale et donnons quelques conseils et perspectives.

Chapitre I

L’imagerie Médicale

I. Introduction

Aujourd'hui, les systèmes d'imagerie sont au cœur de l'environnement médical. ils aident les médecins et les radiologues dans le diagnostic avancé pour le traitement de maladies complexes. Il existe différentes technologies médicales dans deux domaines, l'imagerie nucléaire et la radiologie. Ce dernier est divisé en trois technologies, les rayons X, les champs magnétiques et les ultrasons. Ce chapitre traite des techniques médicales en radiologie en général, ainsi que des détails de l'imagerie par rayons X. Ce qui est dans notre frontière d'intérêt.

1 L'histoire de l'imagerie médicale

Le concept d'imagerie médicale a commencé en 1895, avec l'invention de la radiographie par Wilhelm Rontgen, un professeur de physique allemand. Le principe des rayons X est basé sur le passage des rayonnements ionisants à travers le corps et la projection des images sur une plaque photosensible placée derrière celui-ci. Lorsque la plaque est développée, les différentes densités des tissus à l'intérieur du corps seront détectées et pourront montrer les éventuelles anomalies qui pourraient être présentes. Il a été découvert au début des années 1900 qu'en utilisant des agents de contraste pharmaceutiques, il était possible de voir des organes et des vaisseaux sanguins. La médecine nucléaire a été utilisée pour la première fois pour diagnostiquer une pathologie dans le corps dans les années 1950.

Ceci est basé sur l'infusion au patient de radionucléotides combinés à des composés pharmaceutiques qui se dirigeront vers des organes ou des groupes de cellules plus actifs que les autres. Une caméra gamma enregistre ces images, ce qui permet de détecter les problèmes médicaux plus tôt que d'autres tests. Le sonar a été utilisé pour la première fois dans les années 1960 après avoir été utilisé pendant de nombreuses années comme outil de guerre pour détecter les navires ennemis pendant la Seconde Guerre mondiale. Les ondes sonores à haute fréquence sont transmises dans le corps via une sonde, puis renvoyées à la sonde, où elles sont converties en impulsions électriques qui affichent des images sur un écran. La tomodensitométrie (tomodensitométrie) a été inventée dans les années 1970.

L'idée derrière cette technologie est de prendre une série d'images de tranches de corps, puis de les assembler avec un ordinateur pour visualiser les structures internes du corps. La technologie IRM, qui fonctionne sur le principe des temps de relaxation magnétique nucléaire, a également été développée dans les années 1970.

L'alignement des protons dans les cellules sera examiné à l'aide de forces magnétiques très puissantes pour voir s'il y a un problème avec les tissus dans le corps. Depuis que les premières radiographies ont été prises il y a plus de 120 ans, l'imagerie médicale a énormément progressé. Il y a beaucoup plus de précision dans le diagnostic d'un problème médical, et à la suite de ces progrès, il y a également beaucoup moins besoin de chirurgie exploratoire. Il y a beaucoup plus de précision dans le diagnostic d'un problème médical, et à la suite de ces progrès, il y a beaucoup moins de besoin de chirurgie exploratoire. Cela devrait entraîner un diagnostic plus précoce et de meilleures options de traitement pour de nombreux patients.[10]

2 Différentes techniques d'imagerie médicale

Actuellement, il existe diverses techniques médicales qui vous permettent de regarder à l'intérieur du corps, d'analyser les organes internes et de distinguer les processus pathologiques en leur sein sans intervention chirurgicale. Les images médicales montrent des anomalies humaines que les médecins ne peuvent pas voir, ce qui facilite le diagnostic. Les médecins peuvent également contacter des experts à distance en envoyant des images pour aider à diagnostiquer et à traiter les patients. Ces images sont généralement des images unidimensionnelles (1D), bidimensionnelles (2D), tridimensionnelles (3D) ou quadridimensionnelles (4D = 3D + temps). De nombreuses méthodes d'imagerie actuelles utilisent des ondes électromagnétiques de différentes longueurs d'onde et énergies pour obtenir les informations souhaitées sans prélever d'échantillons sur le corps humain.

M. Goutayer mentionnait dans ses recherches : « Ces techniques utilisent des rayonnements qui se répartissent sur tout le spectre électromagnétique : les rayons gamma en imagerie nucléaire pour les rayonnements les plus énergétiques, les ultrasons en ultrasons, les rayons X et les rayons X dans les scanners, et enfin ondes radio pour l'IRM. » [9]

3 Imagerie par rayon X

3.1 Les rayons X

Les rayons X, comme la lumière visible, sont un type de rayonnement électromagnétique. Les rayons X, en revanche, ont une énergie plus élevée et peuvent traverser la plupart des objets, y compris le corps humain. Les rayons x médicaux sont utilisés pour créer des images d'organes et de tissus internes. Si les rayons x traversant le corps

traversent également un détecteur de rayons X du côté opposé du patient, une image sera formée qui représente les "ombres" formées par les objets à l'intérieur du corps.

Le film photographique est un type de détecteur de rayons X, mais il existe de nombreux autres types de détecteurs utilisés pour créer des images numériques. Les radiographies sont les images produites par ce processus.



Figure 1. *Une photo d'un médecin tenant une radiographie.*

3.2 Comment produire une radiographie ?

Une radiographie est réalisée en positionnant un patient de telle sorte que la partie du corps imagée se trouve entre une source de rayons X et un détecteur de rayons X. Lorsque la machine est allumée, les rayons X traversent le corps et sont absorbés en différentes quantités par différents tissus, en fonction de la densité radiologique des tissus. La densité radiologique des matériaux imagés est déterminée à la fois par leur densité et leur numéro atomique (le nombre de protons dans le noyau d'un atome). Des structures telles que l'os, par exemple, contiennent du calcium, qui a un numéro atomique plus élevé que la plupart des tissus. En raison de cette propriété, les os absorbent facilement les rayons X et produisent un contraste élevé sur le détecteur de rayons X.

En conséquence, les structures osseuses apparaissent plus blanches que les autres tissus sur une radiographie sur fond noir. Les rayons X, en revanche, voyagent plus facilement à travers des tissus moins denses radiologiquement comme la graisse et les muscles, ainsi que

dans des cavités remplies d'air comme les poumons. Sur une radiographie, ces structures sont représentées en niveaux de gris.



Figure 2. *Une illustration d'un patient qui passe une radiographie thoracique*.

3.3 Quand les radiographies médicales sont-elles utiles ?

Les examens et procédures utilisant la technologie des rayons X pour diagnostiquer ou traiter une maladie sont énumérés ci-dessous :

3.3.1 Diagnostique

3.3.1.1 La radiographie aux rayons X

Détecte les fractures osseuses, certaines tumeurs et autres masses anormales, la pneumonie, certains types de blessures, les calcifications, les corps étrangers, les problèmes dentaires et d'autres conditions. [11]

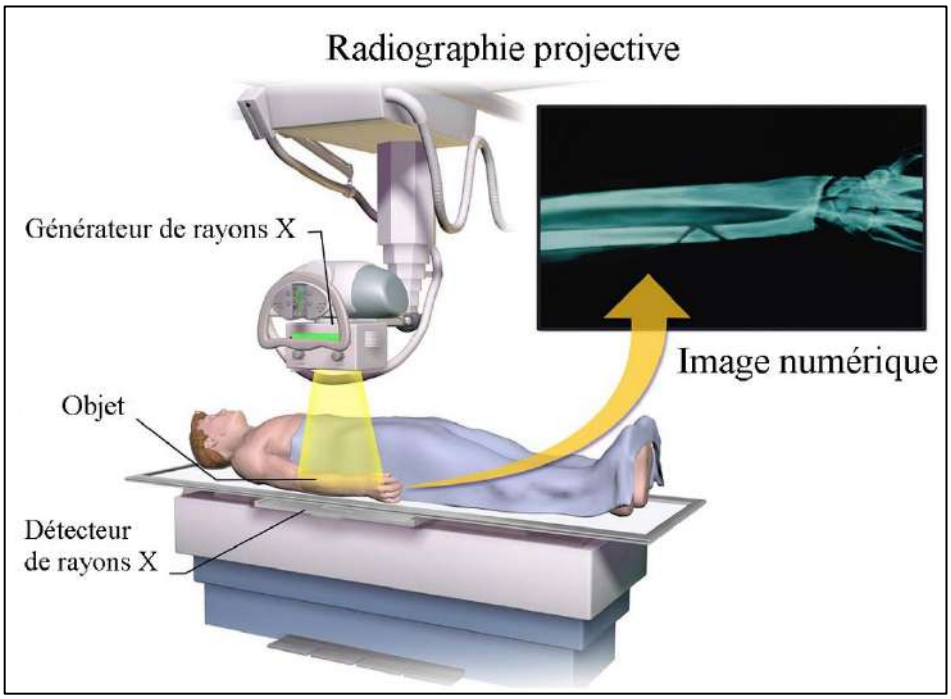


Figure 3. *Acquisition de radiographie par projection médicale.*

3.3.1.2 Mammographie

Radiographie du sein utilisée pour détecter et diagnostiquer le cancer. Les tumeurs apparaissent généralement sur les radiographies sous forme de masses régulières ou irrégulières légèrement plus brillantes que le fond (c'est-à-dire plus blanches sur un fond noir ou plus noires sur un fond blanc). [11]

Les microcalcifications, qui apparaissent sous forme de taches très brillantes sur une mammographie, peuvent également être détectées par mammographie. [11]

Bien que les microcalcifications soient généralement inoffensives, elles peuvent parfois indiquer la présence d'un type spécifique de cancer. [11]

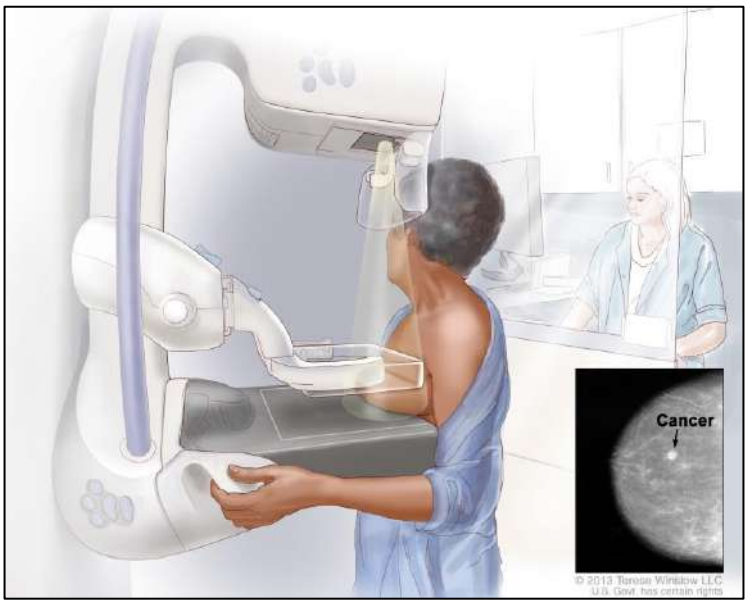


Figure 4. *Une illustration qui montre la mammographie.*

3.3.1.3 CT (tomodensitométrie)

Cette technique combine la technologie traditionnelle des rayons X avec un traitement informatique pour créer une série d'images en coupe transversale du corps qui peuvent ensuite être combinées pour former une image radiographique tridimensionnelle. [11]

Les images CT sont plus détaillées que les radiographies simples et permettent aux médecins de visualiser les structures du corps sous divers angles. [11]

3.3.1.4 Fluoroscopie

Cette technique utilise des rayons X et un écran fluorescent pour obtenir des images en temps réel des mouvements dans le corps ou pour observer des processus de diagnostic, tels que suivre le trajet d'un agent de contraste injecté ou avalé. [11]

La fluoroscopie, par exemple, est utilisée pour examiner le mouvement du cœur battant et, à l'aide d'agents de contraste radiographiques, pour examiner le flux sanguin vers et depuis le muscle cardiaque ainsi que les vaisseaux sanguins et les organes. Cette technologie est également utilisée en conjonction avec un agent de contraste radiographique pour guider un cathéter à filetage interne pendant l'angioplastie cardiaque, une procédure peu invasive utilisée pour ouvrir les artères obstruées qui alimentent le cœur en sang. [11]

3.3.2 Thérapeutique

3.3.2.1 Radiothérapie dans le traitement du cancer

Les rayons X et d'autres formes de rayonnement à haute énergie peuvent être utilisés pour détruire les tumeurs cancéreuses et les cellules en causant des dommages à l'ADN. La dose de rayonnement utilisée pour le traitement du cancer est significativement plus élevée que la dose de rayonnement utilisée pour l'imagerie diagnostique. Le rayonnement thérapeutique peut être délivré par une machine placée à l'extérieur du corps ou par une matière radioactive placée à l'intérieur du corps, à l'intérieur ou à proximité des cellules tumorales, ou injecté dans la circulation sanguine. [11]

4 Le bruit dans l'imagerie médicale

La présence de bruit peut être trouvée dans n'importe quel signal électronique, y compris les images numériques, les trois valeurs de couleur observées dans une image numérique sont incertaines en raison du bruit. Cette incertitude est causée par l'atlas du comptage des photons dans chaque capteur, les valeurs de couleur mesurées sont influencées par le fait que les capteurs reçoivent des photons utiles et parasites et subissent des fluctuations électrostatiques pendant la charge et la décharge. Lorsqu'une caméra reçoit un grand nombre de photons d'une scène bien éclairée, les parasites deviennent insignifiants par rapport au flux de photons réel (utiles). Cependant, même sur une photographie suffisamment exposée, les pixels sombres reçoivent très peu de photons et sont donc "meurtris" [12].

4.1 Les modèles de bruit dans l'image

En général, trois types de bruit sont distingués dans l'image : le bruit additif, le bruit multiplicatif et le bruit convolutif.

4.1.1 Le bruit additif

Le bruit additif peut être défini de la façon suivante : Soit image non bruitée f et g la même image avec un bruit additif b. Chaque pixel j est caractérisé par la relation :

$$gj = fj + b \qquad (1)$$

Où b est une variable aléatoire de moyenne égale à 0.

4.1.2 Le bruit multiplicatif

Le bruit multiplicatif se défini comme suit : Soit une image non bruitée f et g la même image avec un bruit multiplicatif b. Chaque pixel j est caractérisé par la relation :

$$gj = fj \times b \qquad (2)$$

4.1.3 Le bruit convolutif

Un bruit est dit convolutif si étant données une image non bruitée f et g la même image avec un bruit additif b, alors chaque pixel est caractérisé par la relation :

$$gj = fj * b \ \ (Il\ s'agit\ d'une\ convolution) \qquad (3)$$

4.2 Les types de bruit

Dans chaque modèle de bruit il existe plusieurs types. Les plus connus sont le bruit sel et poivre, le bruit gaussien, le flou, le bruit de Poisson (Poisson noise, shot noise), le bruit de speckle. Ces bruits preuve être crée et analysé avec une simulation à l'aide de logiciel image MATLAB.

4.2.1 Bruit sel et poivre

Le bruit impulsionnel, également appelé sel et poivre est une dégradation de l'image. Ce dernier d'ordre n est obtenu en ajoutant n pixels blancs et n pixels noirs aléatoirement dans une image. Il est dû à des défauts de transmission de capteur [13].

Le bruit sel et poivre suit la caractéristique d'une distribution donnée par :

$$P(k) = \frac{e^{-\lambda}\lambda^{k}}{k!} \qquad (4)$$

$p(k)$: La probabilité d'avoir k Pixels affectés par le bruit dans une fenêtre d'une certaine dimension.

λ : est le nombre moyen de pixels affecté dans une fenêtre de la même taille qui est aussi la variance de la distribution de Poisson.

Exemple : la figure 5 nous montre une image dégradée par un bruit sel et poivre



Figure 5. *Image du bruit poivre et sel (à gauche).*

4.2.2 Bruit blanc Gaussien (AWGN : additive white Gaussien noise)

Le bruit gaussien blanc additif (AWGN) est un modèle de bruit fondamental utilisé en théorie de l'information pour simuler l'effet de nombreux processus aléatoires trouvés dans la nature. Les caractéristiques spécifiques sont désignées par les modificateurs : Il est additif car il s'ajoute à tout bruit pouvant exister dans le système d'information [14]. Le bruit gaussien a une fonction de densité de probabilité, elle s'exprime comme suivant :

$$P(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-\mu)^2}{2\sigma 2}} \qquad (5)$$

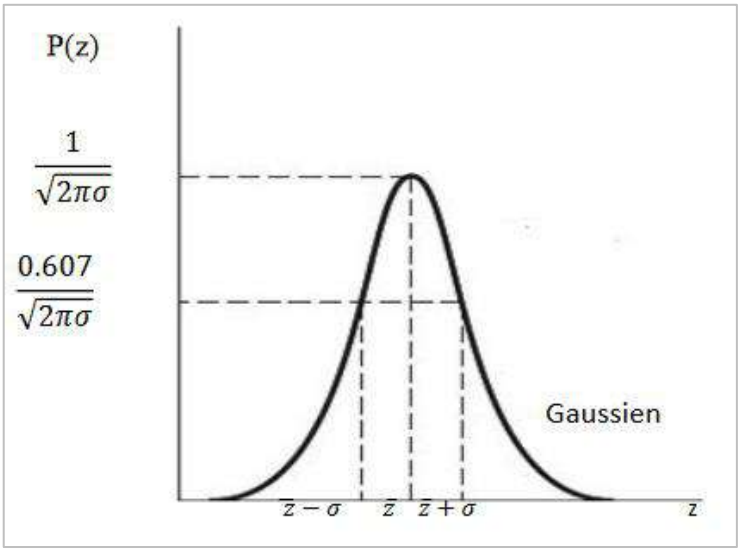


Figure 6. *Courbe de densité de probabilité*



Figure 7. *L'image (a) a un bruit blanc additif.*

4.2.3 Le flou

Le fou est un bruit convolutif (généralement de passage) :

$$g = f0 \times b \qquad (6)$$

qui dégrade la qualité de l'image en raison de divers facteurs tels que le mouvement pendant le processus de capture d'image par le dispositif [15]. La figure 8 représente une image déformée causée par la farine.



Figure 8. *Image floue à gauche.*

4.2.4 Bruit multiplicatif Poisson (Poisson noise, shot noise)

Le bruit de poisson est un type de bruit électronique qui se produit lorsque le nombre de particules porteuses d'énergie, telles que des électrons dans un circuit électronique ou des photons dans un circuit optique, est insuffisant pour provoquer des fluctuations statistiquement discernables dans une mesure. Le nombre de photons mesuré par la surface de détection est aléatoire et dépend de l'éclairement. L'intensité de chaque pixel (m, n) de l'observation y (image dégradée) est donnée par le modèle suivant :

$$\forall m,n \quad \text{tel que} \quad y(m,n) \sim P\big(x(m,n)\big)$$

Ce modèle est utilisé dans le cas d'acquisitions à faible nombre de photons [15].

Loi de Poisson P (λ) :

$$P(k) = \frac{e^{-\lambda}\lambda^{k}}{k!} \qquad (7)$$

Exemple : La figure 9 nous montre une image dégradée par le bruit de Poisson.

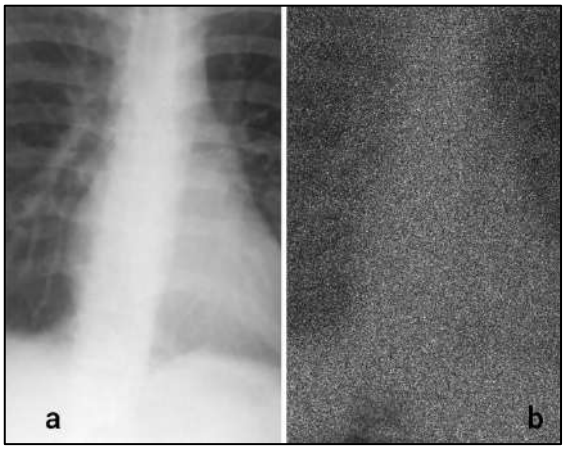


Figure 9. *Image dégradée par le bruit de Poisson.*

5 Débruitage

L'étude d'un signal nécessite l'élimination d'un maximum de bruit parasite dû aux conditions d'acquisition. Un des objectifs du filtrage est d'obtenir un signal utile tout en supprimant le plus de bruit possible tout en conservant le plus d'informations possible.

5.1 Les critères de choix d'un filtre

Les critères de choix d'un filtre sont : MSE, SNR et PSNR

5.1.1 Erreur moyenne quadratique (MSE)

Le plus simple paramètre de mesure de la qualité d'image est l'erreur moyenne quadratique (MSE). C'est la mesure entre l'image de test et l'image reconstruite. Le MSE est défini comme suit :

$$MSE = \frac{1}{M \times N}\sum_{i=1}^{N}\left(f(i,j) - f'(i,j)\right)^2 \quad (8)$$

Avec $f\ (i, j)$: représente l'image originale, $f'(i, j)$: représente l'image reconstruite. M et N sont le nombre de lignes et de colonnes [15].

5.1.2 Erreur signal sur bruit SNR (Signal to Noise Ratio)

Le rapport signal sur bruit (SNR) mesure la quantité de bruit contenue dans le signal. Il s'exprime par le rapport des puissances du signal (P/ S) et du bruit (P /N). Il est souvent donné en décibels (dB) [15].

$$SNR = \frac{Puissance\ de\ l'image\ non\ bruitée}{Puissance\ de\ bruit} \quad (9)$$

5.1.3 Rapport signal sur bruit crête (PSNR)

Le PSNR est défini comme suit :

$$PSNR = 10log(\frac{255}{\sqrt{MSE}}) \quad (10)$$

La faible valeur du PSNR signifie que l'image est de mauvaise qualité [15].

6 Filtrage

Les filtres les plus utilisés sont le médian, moyenneur, gaussien, Wiener. Celui qui nous intéresse dans ce travail est le filtre médian.

6.1 Filtre médian

Il supprime le bruit dans l'image. Son algorithme est relativement simple, le filtre remplace la valeur de chaque pixel par la valeur médiane du pixel et de ses voisins. Le nombre de voisins peut être déterminé à l'aide d'un paramètre 'N'. Ce paramètre indique la taille du carré dans lequel seront effectués les calculs. La figure montre le principe du calcul de la valeur médiane dans le cas d'un filtre médian 3x3 (N=9) [15].

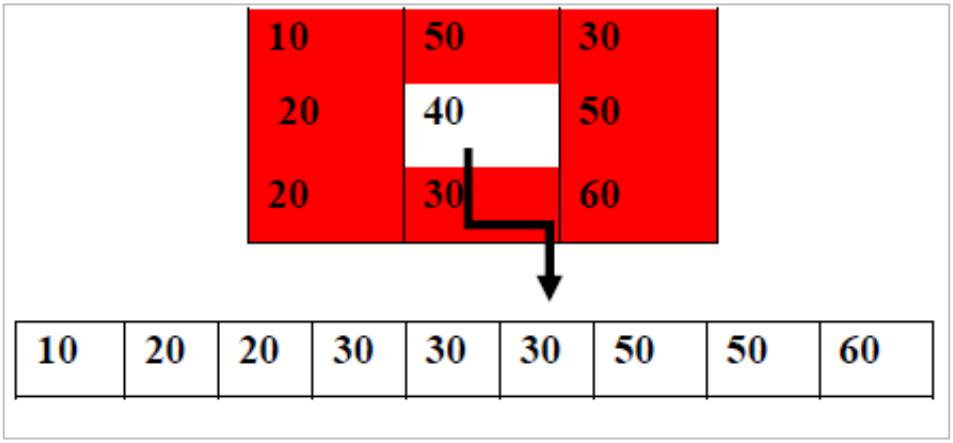


Figure 10. *Principe du filtre médian 3x3.*

II. Conclusion

Dans ce premier chapitre, nous parlons de l'époque où le concept de l'image médicale a commencé, qui remonte à 1895 par Wilhelm Rontgen, nous parlons de la variété des images médicales qui existent de nos jours, des rayons X à l'IRM, nous parlons de la façon dont ces images sont créées, nous mentionnons que ces images ne produisent pas toujours dans une forme parfaite, parfois elles sont accompagnées de bruit et de flou et d'autres distorsions, nous parlons de ces types spécifiques de bruit et définissons leurs équations mathématiques. , avant cela, nous parlons également de l'utilité des images médicales et de la façon dont elles ont aidé le domaine moderne de la médecine nous enrichissons nos recherches avec des illustrations et des images obtenues à partir de la simulation du bruit pour les différents types de bruit discutés.

Dans le prochain chapitre, nous parlerons de l'apprentissage en profondeur et les réseaux de neurones convolutifs.

Chapitre II

Apprentissage profond

I. Introduction

Yann LeCun, spécialiste de l'apprentissage automatique des machines (*machine learning*), est l'un des pères du *Deep Learning* (apprentissage profond), une méthode à laquelle il se consacre depuis trente ans, malgré le scepticisme qu'il rencontre au départ dans la communauté scientifique. [16]

Le *Deep Learning*, qui fait appel à la fois aux connaissances en neurosciences, aux mathématiques et aux progrès technologiques, est aujourd'hui plébiscité comme une véritable révolution dans le domaine de l'intelligence artificielle. Il a déjà permis d'immenses progrès et de multiples applications dans les domaines de la reconnaissance faciale et vocale, de l'étiquetage d'images, du traitement automatisé du langage ou encore de la vision par ordinateur. [16]

« Les cerveaux humain et animal sont " profonds ", dans le sens où chaque action est le résultat d'une longue chaîne de communications synaptiques (de nombreuses couches de traitement). Nous recherchons des algorithmes d'apprentissage correspondants à ces " architectures profondes ". Nous pensons que comprendre l'apprentissage profond ne nous servira pas uniquement à construire des machines plus intelligentes, mais nous aidera également à mieux comprendre l'intelligence humaine et ses mécanismes d'apprentissages », estime Yann LeCun. [16]

Les réseaux de neurones, également appelés réseaux de neurones artificiels (ANNs) ou réseaux de neurones simulés (SNN), sont un sous-ensemble de l'apprentissage automatique qui constitue la base des algorithmes d'apprentissage profond. Leur nom et leur structure sont inspirés du cerveau humain, et ils imitent la façon dont les neurones biologiques communiquent entre eux. Les réseaux de neurones artificiels (RNA) sont constitués de couches de nœuds, chacune ayant une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Chaque nœud, ou neurone artificiel, est lié à un autre et a son propre poids et son propre seuil. Si la sortie d'un nœud individuel dépasse la valeur de seuil spécifiée, ce nœud est activé et commence à envoyer des données à la couche suivante du réseau. Sinon, aucune donnée n'est transmise à la couche réseau suivante. [17]

II. Historique

L'histoire des réseaux de neurones est plus longue que ce que la plupart des gens pensent. Bien que l'idée d'une machine qui pense remonte aux Anciens Grecs, nous nous concentrerons sur les événements clés qui ont conduit à l'évolution de la pensée autour des réseaux de neurones, qui a connu un déclin et une popularité au fil des ans :

1943: Warren S. McCulloch et Walter Pitts ont publié "***A logical calculus of the ideas immanent in nervous activity***". Cette recherche visait à comprendre comment le cerveau humain pouvait produire des modèles complexes à travers des cellules cérébrales connectées, ou neurones. L'une des idées principales qui est sortie de ce travail a été la comparaison des neurones avec un seuil binaire à la logique booléenne (c'est-à-dire 0/1 ou déclarations vraies / fausses) [18].

1958: Frank Rosenblatt est crédité du développement du perceptron, documenté dans ses recherches, "***The Perceptron : A Probabilistic Model for Information Storage and Organization in the Brain***". Il pousse les travaux de McCulloch et Pitt un peu plus loin en introduisant des poids dans l'équation. Grâce à un IBM 704, Rosenblatt a pu obtenir un ordinateur pour apprendre à distinguer les cartes marquées à gauche des cartes marquées à droite [19].

1974: Alors que de nombreux chercheurs ont contribué à l'idée de la rétropropagation, Paul Werbos a été la première personne aux États-Unis à noter son application dans les réseaux de neurones dans sa thèse de doctorat "***Beyond regression : New Tools For Prediction and Analysis in the Behavioral Sciences***" [20].

1989: Yann LeCun publie un article "***Backpropagation Applied to Handwritten Zip Code Recognition***" illustrant comment l'utilisation des contraintes dans la rétropropagation et son intégration dans l'architecture des réseaux de neurones peuvent être utilisées pour entraîner des algorithmes. Cette recherche a réussi à exploiter un réseau neuronal pour reconnaître les chiffres du code postal écrits à la main fournis par le Service postal américain. [21].

1 Réseaux de neurones

Les réseaux de neurones sont actuellement l'un des algorithmes d'apprentissage automatique les plus populaires. Il a été démontré que les réseaux de neurones surpassent les autres algorithmes en termes de précision et de vitesse au fil du temps. Les réseaux de neurones deviennent progressivement pour les scientifiques ou les praticiens de l'apprentissage automatique ce que la régression linéaire était pour les statisticiens, avec diverses variantes telles que CNN (Réseaux de Neurones convolutionnels (en abrégé CNN)), RNN (Réseaux de Neurones Récurrents), Auto-encodeurs, etc.

Les réseaux neuronaux profonds (CNN) ont connu un succès notable dans la reconnaissance et la localisation d'objets dans des images. L'approche fondamentale qui a conduit à CNN est la construction de systèmes fictifs basés sur le cerveau et la vision humaine. Néanmoins, à bien des égards importants, les capacités de CNN sont inférieures à celles de la vision humaine. Une orientation de recherche prometteuse consiste à étudier les similitudes et les différences tout en comblant les lacunes afin d'améliorer le CNN. Les algorithmes d'apprentissage profond sont disponibles dans une variété de saveurs.

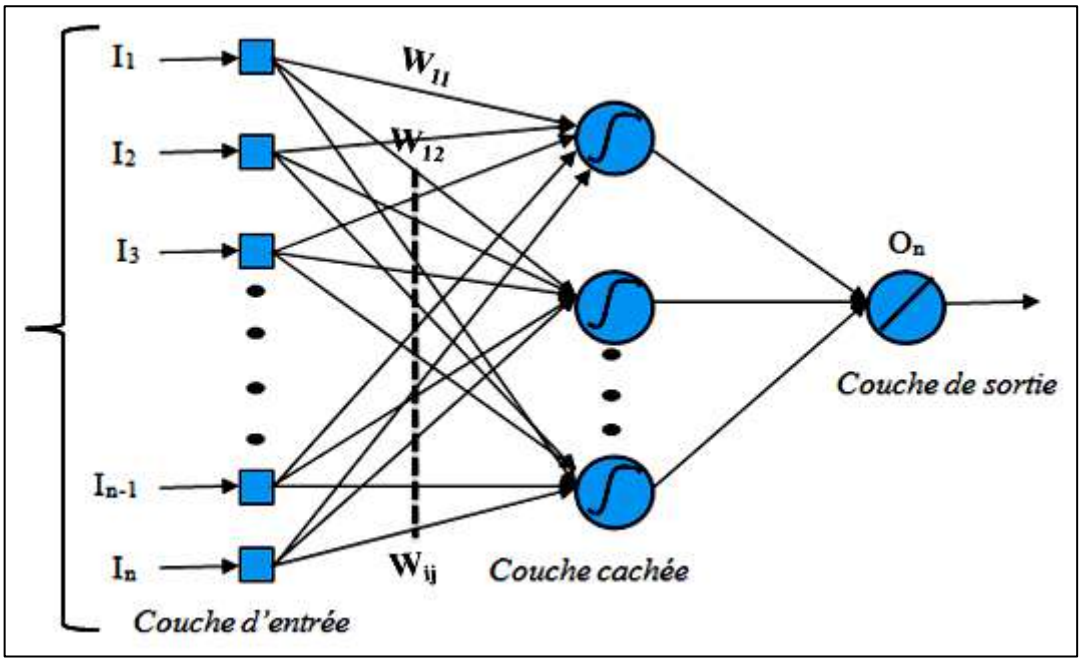


Figure 11. *Schéma d'un réseau de neurones artificiel*

Dans notre projet, les méthodes de débruitage sont ce qui nous intéresse dans le deep learning, nous parlerons donc plus précisément des réseaux CNN et de leur fonctionnement.

2 Réseaux de neurones convolutifs

La vision par ordinateur évolue rapidement de jour en jour. L'une des raisons est l'avancement de l'apprentissage profond. Lorsque nous parlons de vision par ordinateur, le terme "réseau de neurones convolutifs" vient à l'esprit car CNN est fréquemment utilisé dans ce contexte. Des exemples de CNN en vision par ordinateur incluent la reconnaissance faciale, la classification des images, etc. C'est similaire à un réseau de neurones de base. CNN a également des paramètres entraînables, tels que le réseau de neurones, qui comprend les pondérations, les biais, etc.

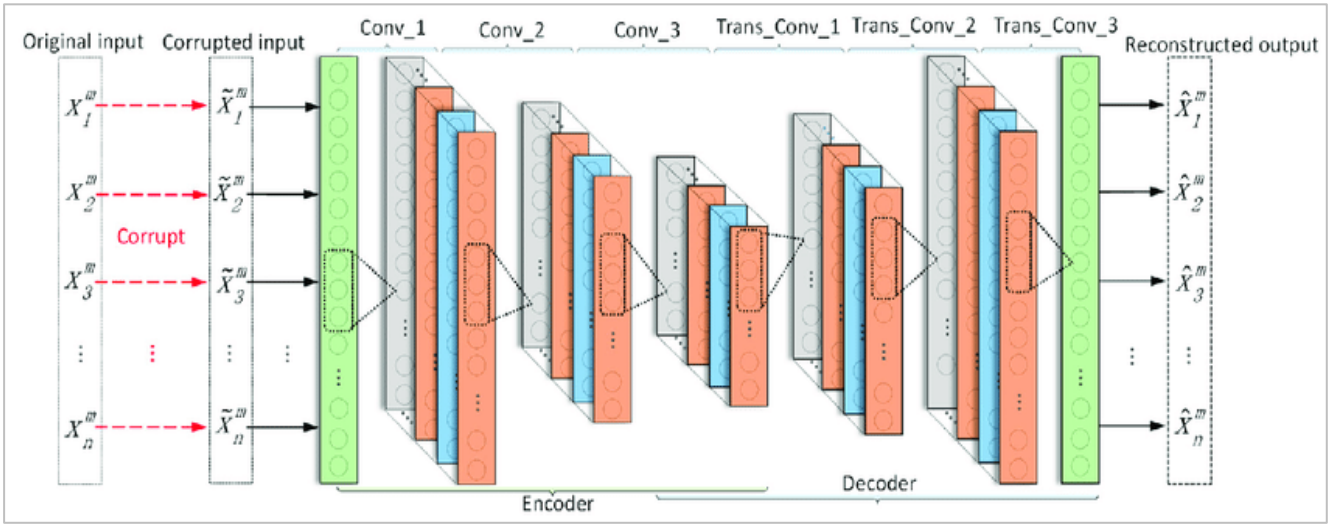


Figure 12. *Exemple d'un réseau CNN pour le débruitage*

3 Les couches de réseaux de neurones convolutionnels

- Couche d'entrée.
- Couche de convolution (Convo layer : Convolution + ReLU).
- Couche de Pooling.
- Couche entièrement connectée (Fully-connected layer).
- Couche de pertes.
- Couche de sortie (Output layer).

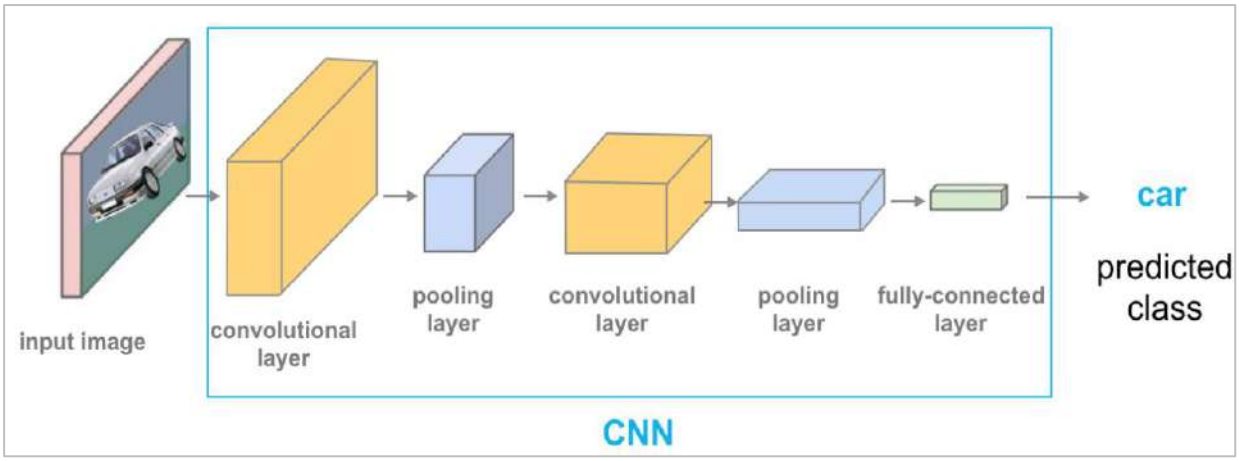


Figure 13. *Exemple d'architecture CNN*

3.1 Couche d'entrée du CNN

La couche d'entrée dans CNN doit contenir des données descriptives d'image. Les données d'image sont représentées par une matrice tridimensionnelle, qui est généralement remodelée en une seule colonne (représentation vectorielle).

3.2 Couche de convolution

La couche de convolution est également appelée couche d'extraction de caractéristiques, car les caractéristiques de l'image sont extraites dans cette couche. Pour commencer, une partie de l'image est connectée à la couche convo afin d'effectuer une opération de convolution et de calculer le produit scalaire entre le récepteur de champ (une région locale de l'image entrante de même taille que le filtre) et le filtre. Le résultat final de l'opération est un volume de sortie unique en son genre. Ensuite, nous allons faire glisser le filtre sur le champ récepteur qui suit la même image depuis le début et répéter le processus. Cette opération est répétée par le même processus jusqu'à ce que l'image entière ait été parcourue.

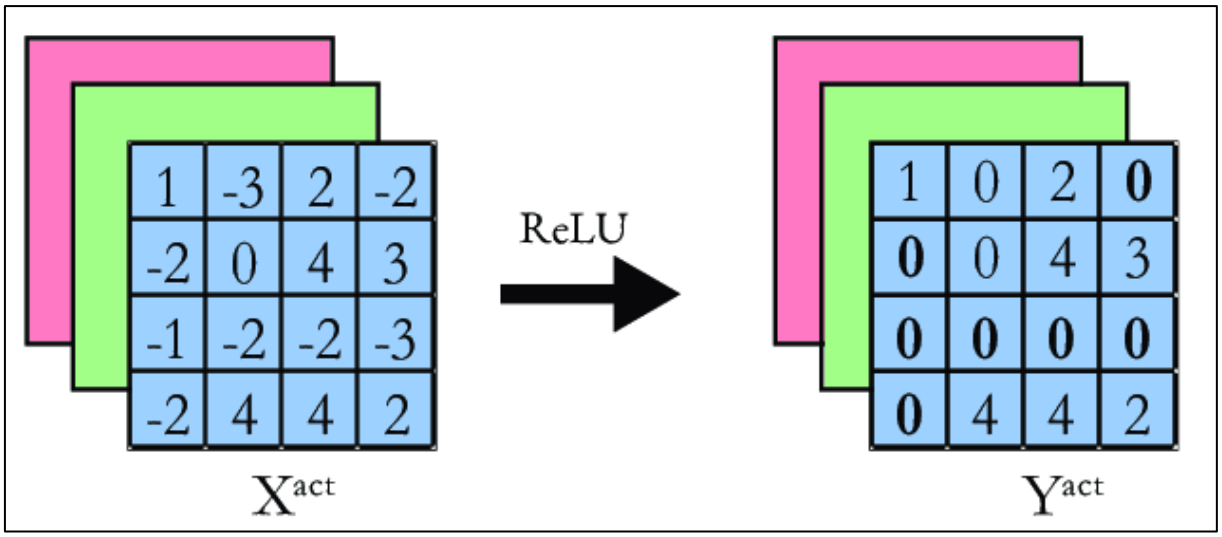


Figure 14. *Exemple de principe du filtre convolutionnel*

La couche Convo inclut également l'activation ReLU, qui neutralise les valeurs négatives, comme le montre la figure 15.

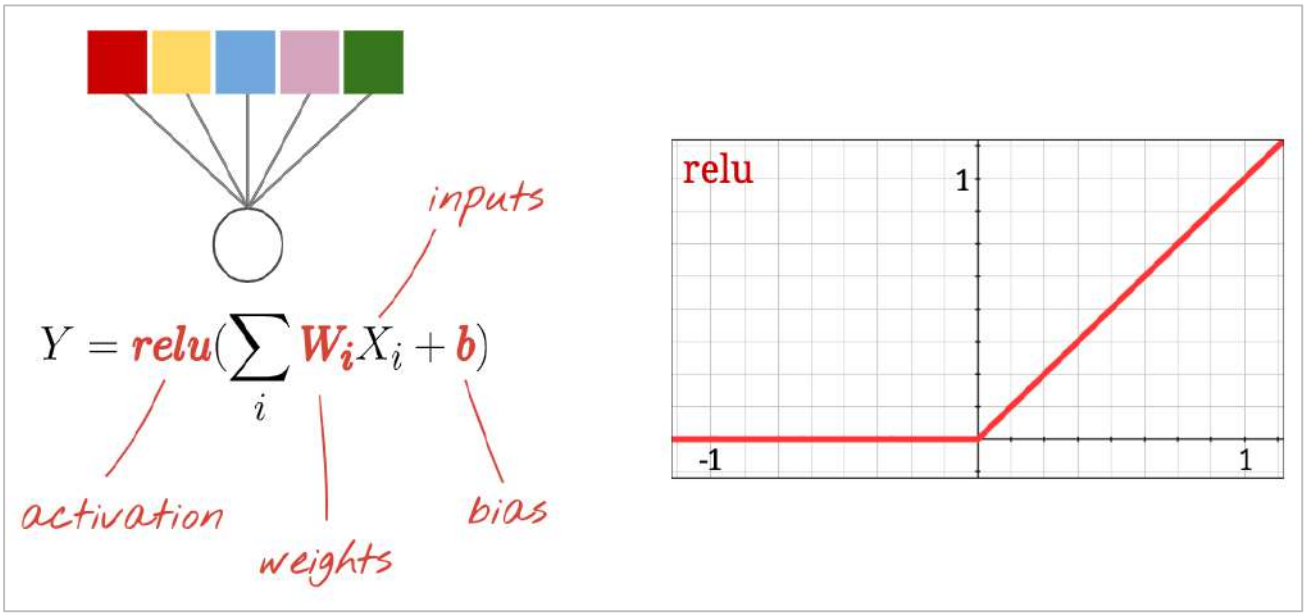


Figure 15. *Principe de fonction ReLu*

3.3 Couche de mise en commun (Pooling)

La couche de pooling est utilisée pour réduire le volume spatial de l'image entrante après la convolution. Elle est utilisée entre deux couches convolutives. Si nous appliquons ***fc*** (Fully Connected) après la couche Convo sans utiliser le pooling ou le pooling maximum, le calcul sera coûteux. En conséquence, le dénominateur commun maximal est le seul moyen de réduire le volume spatial de l'image entrante tout en codant des informations.

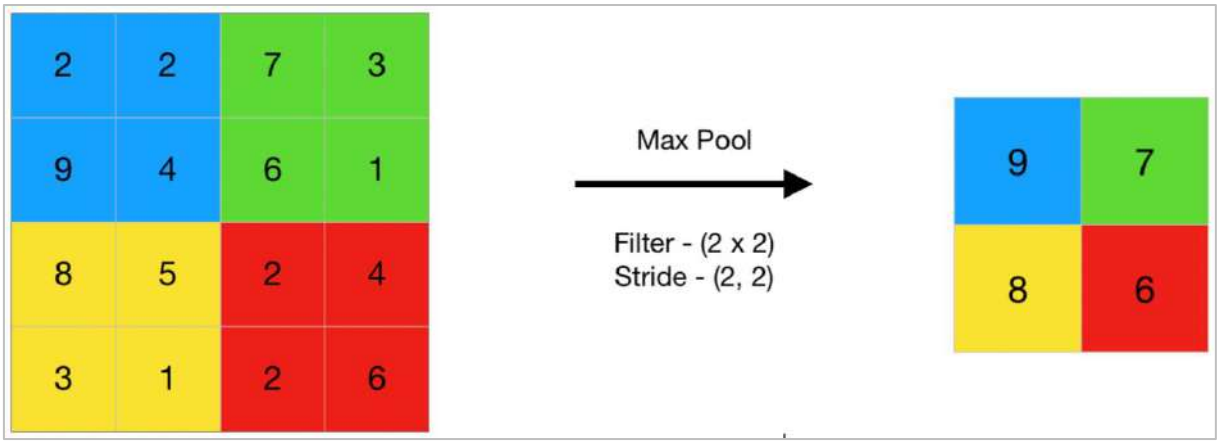


Figure 16. *Exemple de principe de pooling.*

3.4 Couche entièrement connecté (Fully Connected)

La couche entièrement connectée consiste simplement à alimenter les réseaux de neurones. Les couches entièrement connectées forment les dernières couches du réseau. L'entrée de la couche entièrement connectée est la sortie de la couche de mise en commun finale ou de convolution, qui est aplatie puis introduite dans la couche entièrement connectée.

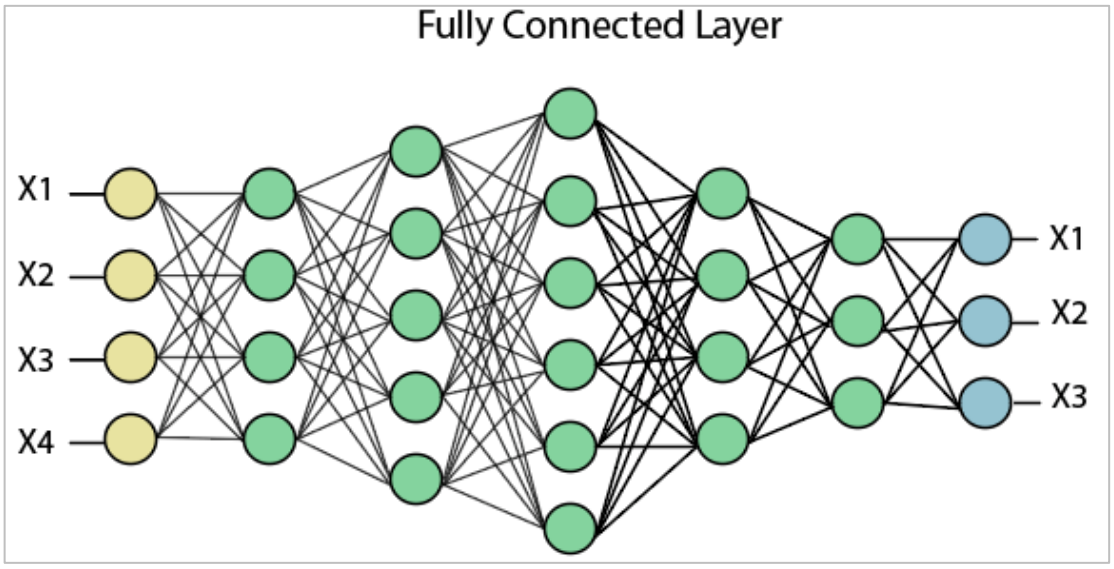


Figure 17. *Principe de la couche entièrement connectée*

3.5 Couche de perte

Elle est la dernière couche du réseau, et diverses fonctions de perte adaptées à différentes tâches peuvent lui être appliquées. La fonction Softmax calcule la distribution de probabilité sur les classes de sortie.

3.6 Couche de sortie (output layer)

La couche de sortie est la dernière couche de neurones qui produit des sorties données pour le programme qui comprend (nombre total de paramètres).

4 Auto-encodeur

Un auto-encodeur est composé de deux composants principaux : un encodeur qui convertit l'entrée en code et un décodeur qui convertit le code en une reconstruction de l'entrée. L'objectif d'un auto-encodeur est d'apprendre une représentation (encodage) d'un ensemble de données, généralement dans le but de réduire la dimension de cet ensemble [28].

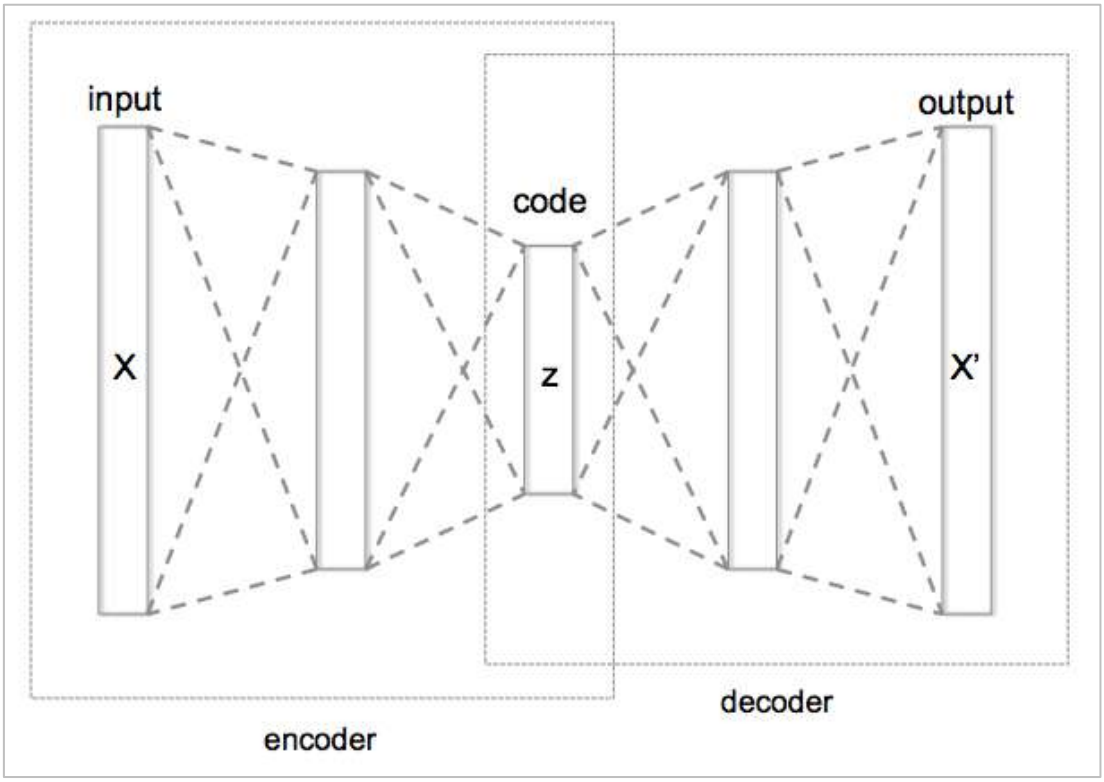


Figure 18. *Architecture d'un autoencodeur*

Le décodeur est fondamentalement l'élément opposé à l'encodeur dans l'autoencodeur et il a également les mêmes couches, mais elles font l'opération inverse.

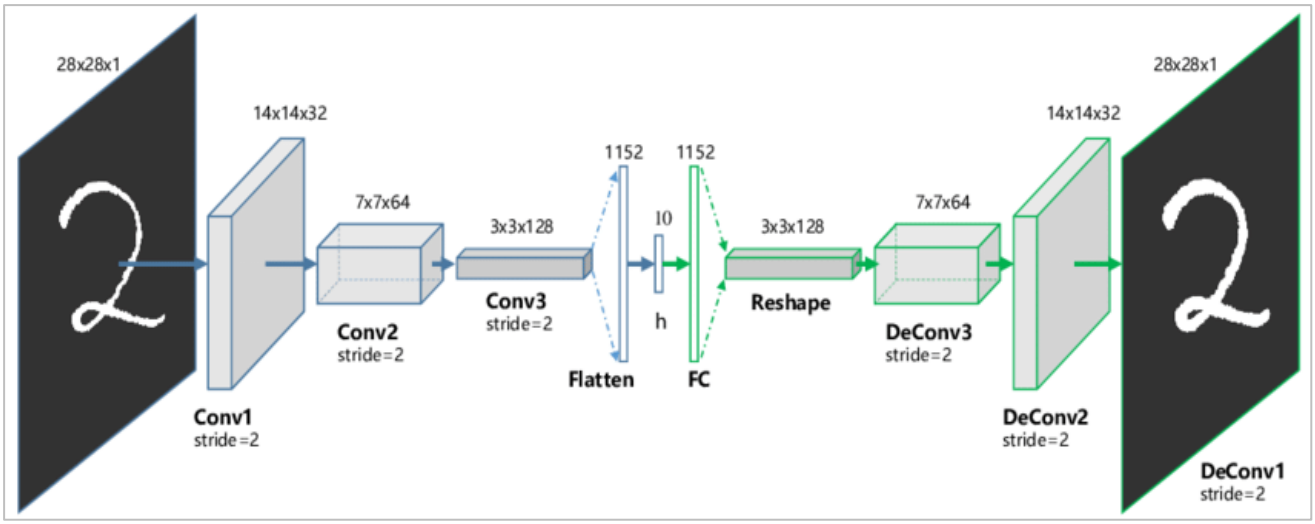


Figure 19. *Architecture d'un autoencodeur*

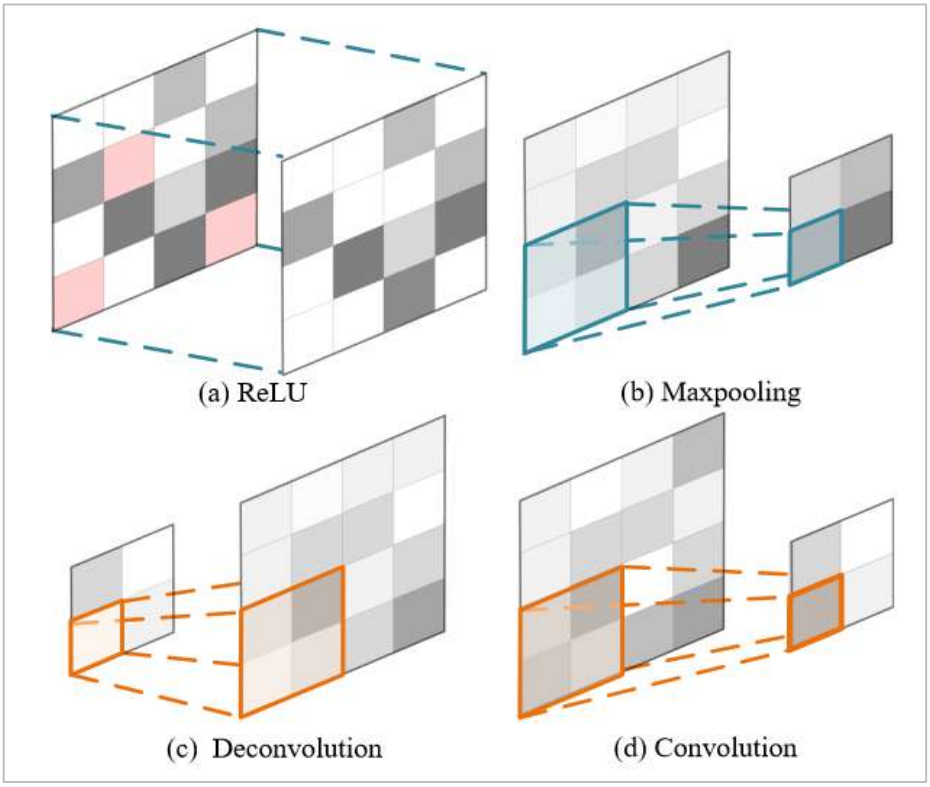


Figure 20. *Opérations Conv/Deconv Relu et Maxpooling*

5 Opérations de couches convolutives

Les couches convolutives appliquent une opération de convolution pour l'apprentissage de la représentation et l'extraction des entités, où les poids de chaque filtre fonctionnent comme un dictionnaire de motifs d'entités. Pour le patch d'entrée X et le noyau de filtre K, l'opération de convolution peut être décrite comme suit :

$$C(m,n) = \sum_{p=0}^{P} \sum_{q=0}^{Q} K(p,q) * X(m+p, n+q) + b \qquad (11)$$

où $*$ est l'opération de convolution, {m, n}, {P, Q}, {p, q} se réfèrent à l'emplacement du patch 2D, à la taille du noyau de convolution et à l'indice du noyau de convolution, respectivement.

6 Logiciel utilisé dans l'apprentissage profond

Pour notre projet, nous avons choisi Python pour la variété de bibliothèques et de fonctions dont il dispose.

6.1 Python

Python est un langage de programmation informatique souvent utilisé pour créer des sites Web et des logiciels, automatiser des tâches et effectuer des analyses de données. Python est un langage à usage général, ce qui signifie qu'il peut être utilisé pour créer une variété de programmes différents et n'est pas spécialisé pour des problèmes spécifiques.

6.1.1 Bibliothèques utilisées dans ce projet (Python)

6.1.1.1.1 OpenCV (Open-Source Computer Vision Library)

OpenCV est une bibliothèque d'apprentissage automatique. OpenCV a été conçu pour fournir une infrastructure commune pour les applications de vision par ordinateur et pour accélérer l'utilisation de la perception de la machine dans les produits commerciaux.

6.1.1.1.2 Matplotlib

Matplotlib est une bibliothèque multiplateforme de visualisation de données et de traçage graphique pour Python et son extension numérique NumPy. En tant que tel, il offre une alternative open source viable à MATLAB. Les développeurs peuvent également utiliser les API de matplotlib (Interfaces de programmation d'applications) pour intégrer des tracés dans des applications GUI.

6.1.1.1.3 Numpy (Numerical Python)

NumPy est une bibliothèque Python utilisée pour travailler avec des tableaux. Il a également des fonctions pour travailler dans le domaine de l'algèbre linéaire, de la transformée de fourier et des matrices. NumPy a été créé en 2005 par Travis Oliphant. C'est un projet open source et vous pouvez l'utiliser librement. NumPy signifie Python numérique.

6.1.1.1.4 Keras

Keras est utilisé pour créer des modèles profonds qui peuvent être produits sur des smartphones. Keras est également utilisé pour la formation distribuée de modèles d'apprentissage profond. Keras est utilisé par des entreprises telles que Netflix, Yelp, Uber, etc.

6.1.1.1.5 Tensorflow

TensorFlow est un framework open-source développé par les chercheurs de Google pour exécuter l'apprentissage automatique, l'apprentissage profond et d'autres charges de travail d'analyse statistique et prédictive.

III. Conclusion

Dans ce chapitre, nous nous sommes familiarisés avec l'apprentissage automatique et comment il est apparu pour la première fois et qui sont les premiers à en adapter l'idée et à l'améliorer au fil des ans. Nous avons discuté d'un domaine plus spécifique de l'apprentissage automatique qui est l'apprentissage en profondeur.

Ensuite, nous avons défini les réseaux de neurones convolutifs et décrit leurs différentes couches et parlé de leur utilité dans les domaines techniques, et à la fin de ce chapitre, nous avons parlé du logiciel le plus utilisé dans l'apprentissage en profondeur qui est python, nous avons indiqué certaines bibliothèques qui sont les plus courantes dans l'apprentissage automatique.

Dans le chapitre suivant, nous passerons à la partie manipulation et discuterons des résultats obtenus.

Chapitre III

Simulation et résultats

I. Introduction

Le débruitage d'image est utile dans une variété d'applications, y compris la restauration d'image, le suivi visuel, l'enregistrement d'image, la segmentation d'image et la classification d'image, où l'obtention du contenu d'image d'origine est essentielle pour des performances élevées.

Donc, dans ce chapitre, nous allons discuter des résultats obtenus lors des simulations que nous avons faites.

II. Plan de travail

Nous avons utilisé un simulateur appelé Anaconda qui a un éditeur python intégré appelé Spyder qui nous a aidés et a rendu notre projet facile

Avantages de l'utilisation de Python Anaconda :

- Il est gratuit et open-source.
- Il contient plus de 1500 packages de science des données Python/R.
- Anaconda simplifie la gestion et le déploiement des packages.
- Il dispose d'outils pour collecter facilement des données à partir de sources utilisant l'apprentissage automatique et l'IA.

Pour parvenir à un système intelligent indépendant spécialisé dans le débruitage d'image, nous devons suivre certaines étapes qui sont présentées dans l'organigramme suivant :

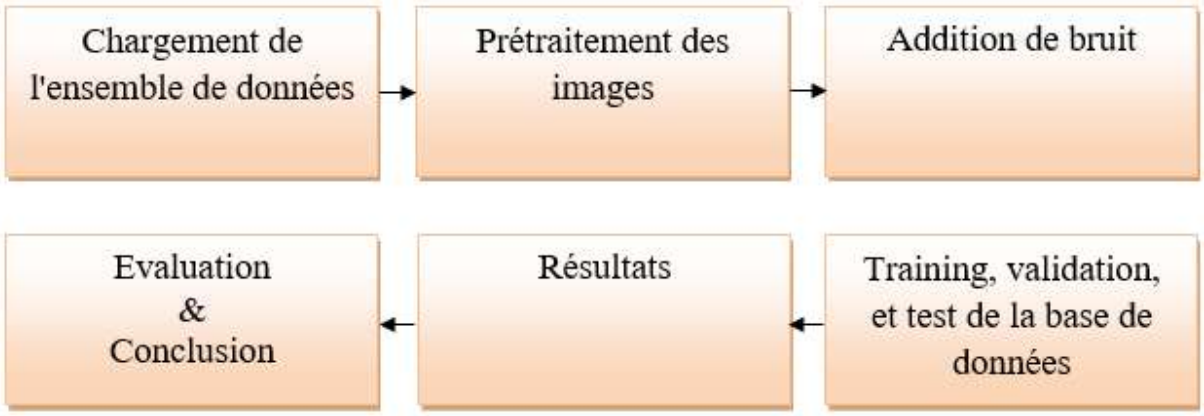


Figure 21. *Un organigramme qui représente le plan de travail pour la simulation*

1. Chargement de l'ensemble de données

Nous utilisons l'ensemble de données publié par l'auteur Pranav Raikote [29] dans les bibliothèques de kaggle, qui vise à aider les passionnés d'apprentissage en profondeur et d'IA à contribuer à améliorer la détection du COVID-19 en utilisant uniquement des radiographies thoraciques, mais dans notre cas, nous utiliserons l'ensemble de données à des fins de débruitage.

A l'aide de ligne de code suivante :

```
train_images = sorted(os.listdir('C:/Users/pc/Desktop/Covid/'))
```

Aperçu de la base de données:

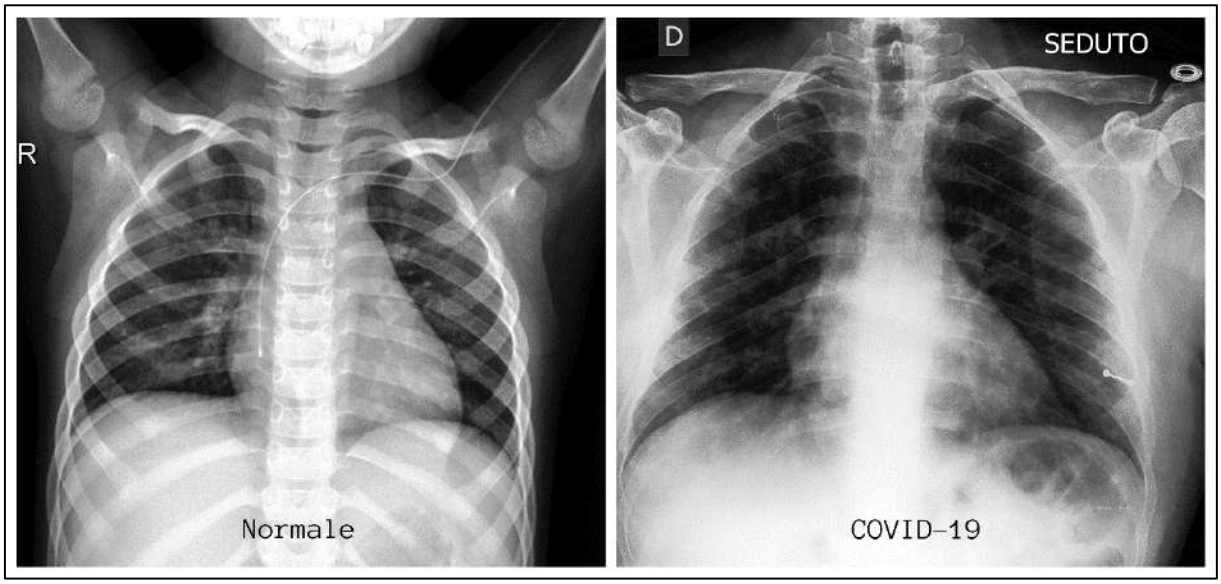


Figure 22. *Aperçu de l'ensemble de données*

2. Prétraitement des images

Dans cette étape, nous effectuons un prétraitement des images de l'ensemble de données que nous avons, tel que le redimensionnement à 128x128 et la conversion en niveaux de gris qui simplifie l'algorithme et réduit les exigences de calcul, nous divisons ensuite les pixels par 255 de cette manière, les nombres seront petits et le calcul devient plus facile et plus rapide. Comme les valeurs de pixels vont de 0 à 256, en dehors de 0, la plage est de 255. Donc, diviser toutes les valeurs par 255 le convertira en une plage de 0 à 1 nous utilisons le code suivant pour cela :

```
train_image = []
for im in train_images:
    img = image.load_img('
C:/Users/pc/Desktop/Covid/'+ im,
target_size=(128,128), color_mode= 'grayscale')
    img = image.img_to_array(img)
    img = img/255
    train_image.append(img)
train_df = np.array(train_image)
```

3. Addition de bruit

Dans cette étape, nous ajoutons du bruit gaussien à l'image, et nous lui attribuons différentes valeurs au cours du processus en faisant varier ses paramètres (sigma et autres) en utilisant le code suivant :

```
def add_noise(image):
    row,col,ch= image.shape
    mean = 0
    sigma = 1
    gauss = np.random.normal(mean,sigma,(row,col,ch))
    gauss = gauss.reshape(row,col,ch)
    noisy = image + gauss*0.07
    return noisy
```

4. Training, validation, et test de la base de données

Dans cette partie, nous construisons notre autoencodeur et définissons les paramètres appropriés pour qu'il fonctionne correctement, nous utiliserons des couches convolutives comme indiqué dans le code suivant :

```
def autoencoder():
input_img = Input(shape=(64,64,1), name='image_input')
#enodeur
x = Conv2D(64, (3,3), activation='relu', padding='same', name='Conv1')(input_img)
x = MaxPooling2D((2,2), padding='same', name='pool1')(x)
x = Conv2D(64, (3,3), activation='relu', padding='same', name='Conv2')(x)
x = MaxPooling2D((2,2), padding='same', name='pool2')(x)
#decodeur
x = Conv2D(64, (3,3), activation='relu', padding='same', name='Conv3')(x)
x = UpSampling2D((2,2), name='upsample1')(x)
x = Conv2D(64, (3,3), activation='relu', padding='same', name='Conv4')(x)
x = UpSampling2D((2,2), name='upsample2')(x)
x = Conv2D(1, (3,3), activation='sigmoid', padding='same', name='Conv5')(x)
return autoencoder
```

Après cela, nous définissons les paramètres appropriés pour entraîner notre modèle tels que : nombre d'itérations, taille du lot, arrêt précoce, patience.

```
with tf.device('/device:GPU:0'):
    early_stopping = EarlyStopping(monitor='val_loss', min_delta=0, patience=10, verbose=1, mode='auto')
    model.fit(xnoised, xnoised, epochs=50, batch_size=128, validation_data=(xtest, xtest), callbacks=[early_stopping])
```

5. Résultats

Dans cette phase, nous compilons après nous être assurés que nous avons les paramètres appropriés et que le code s'exécute sans erreur, nous allons ensuite modifier la valeur du bruit ajouté et mesurer l'efficacité du modèle.

- **Pour : 50 itérations, 128 tailles de lot et une valeur de bruit de 0,07**

```
def add_noice(image):
    row,col,ch= image.shape
    mean = 0
    sigma = 1
    gauss = np.random.normal(mean,sigma,(row,col,ch))
    gauss = gauss.reshape(row,col,ch)
    noisy = image + gauss*0.07
    return noisy
```

```
with tf.device('/device:GPU:0'):
    early_stopping = EarlyStopping(monitor='val_loss',
min_delta=0, patience=10, verbose=1, mode='auto')
    model.fit(xnoised, xnoised, epochs=50, batch_size=128,
validation_data=(xtest, xtest), callbacks=[early_stopping])
```

Tableau 1. *Les valeurs de critère PSNR pour la 1ère manipulation*

Type d'image	**PSNR (dB)**
Filtre médian	55.4471
Auto-encodeur	67.8917

Tableau 2. *Les valeurs de critère MSE pour la 1ère manipulation*

Type d'image	**Erreur moyenne quadratique (MSE)**
Filtre médian	0.131016
Auto-encodeur	0.0496749

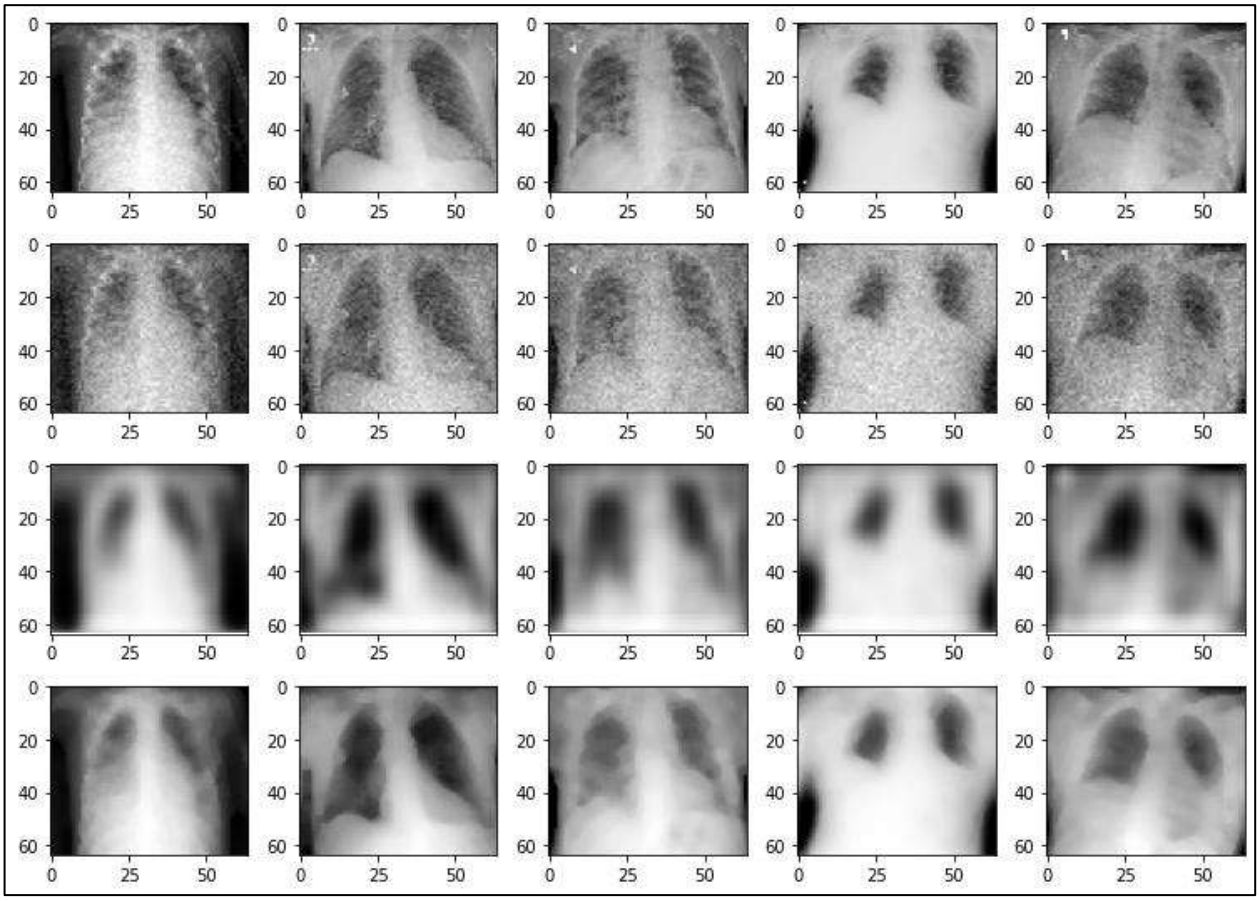


Figure 23. *Images des résultats de manipulation 1 du modèle CNN*

- La première colonne représente les images originales.
- La deuxième colonne représente les images bruitées.
- La troisième colonne représente les images débruitées par filtre médian.
- La quatrième colonne représente les images débruitées par l'auto-encodeur.

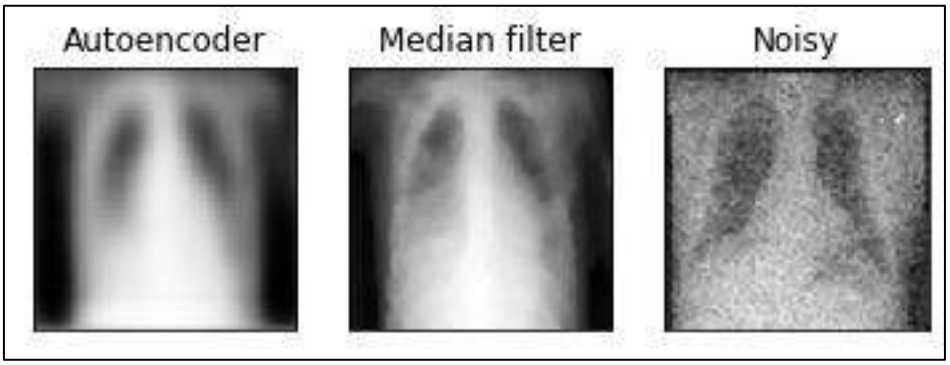


Figure 24. *Un échantillon qui montre les deux méthodes de débruitage utilisées*

- **Pour : 50 itérations, 128 tailles de lot et une valeur de bruit de 0,09**

Tableau 3. *Les valeurs de critère PSNR pour la 2ème manipulation*

Type d'image	**PSNR (dB)**
Filtre médian	55.3695
Autoencodeur	66.9145

Tableau 4. *Les valeurs de critère MSE pour la 2ème manipulation*

Type d'image	**Erreur moyenne quadratique (MSE)**
Filtre médian	0.131016
Autoencodeur	0.0496749

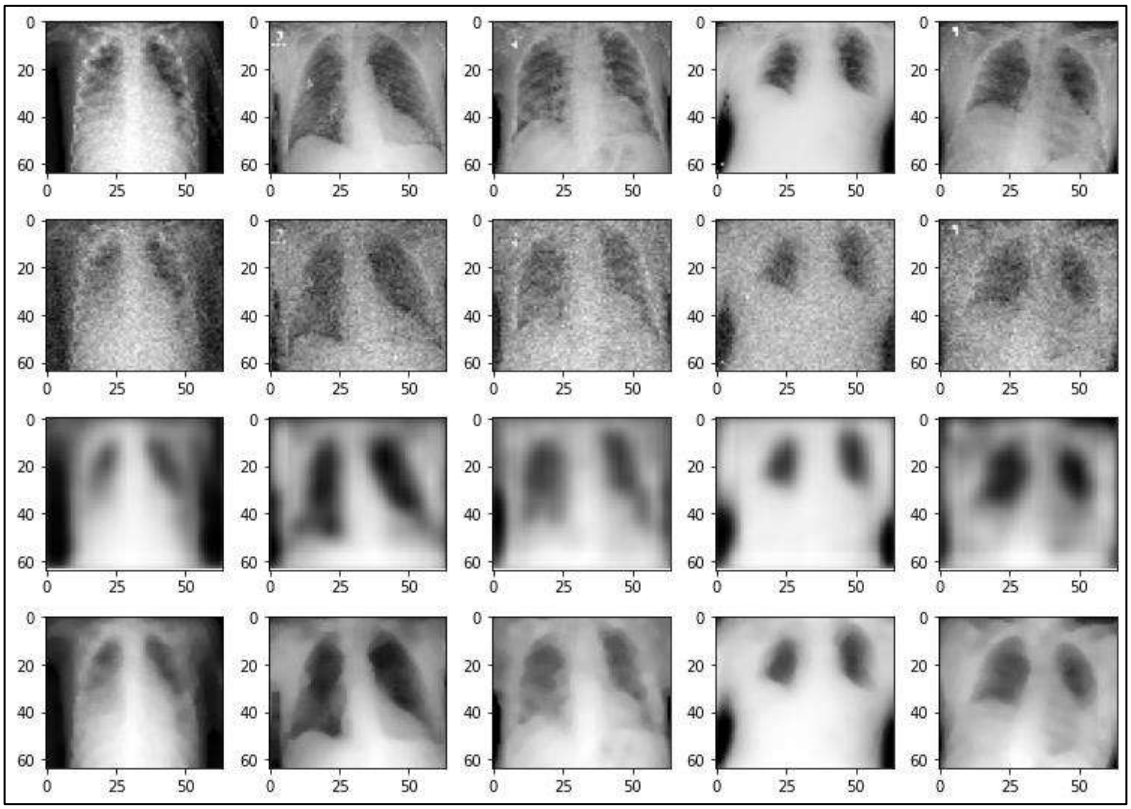


Figure 25. *Images des résultats de manipulation 2 du modèle CNN*

6. Evaluation

6.1 Première manipulation

Nous remarquons que la valeur PSNR dans la première manipulation dans l'auto-encodeur est de 67,89 dB, ce qui est mieux par rapport à celle du filtre médian qui est de 55,44 dB, et bien sûr plus elle est élevée, mieux c'est.

En ce qui concerne la valeur MSE, nous remarquons que l'auto-encodeur a la valeur la plus basse de 0,049, ce qui est un bon signe, mais dans la main opposée, le filtre médian a une valeur de MSE plus élevée de 0,13

6.2 Deuxième manipulation

Dans la deuxième manipulation, nous remarquons un grand écart dans la valeur PSNR entre les deux techniques de débruitage, l'auto-encodeur a marqué une valeur de 66,91 dB, ce qui est remarquablement plus grand que le résultat du filtre médian qui est de 55,36 dB.

Et pour les valeurs MSE, l'auto-encodeur est toujours en tête avec une bonne valeur de 0,05 par rapport au filtre médian avec une valeur de 0,133

D'après les résultats obtenus, on peut dire que l'auto-encodeur a plus d'efficacité de débruitage que le filtre médian car il protège les éléments de l'image d'origine, après le débruitage contrairement au filtre médian, il conduit à perdre la majeure partie de l'image d'origine ce qui entraîne la perte des informations qui pourraient être précieuses pour les médecins faisant leur diagnostic.

Après le débruitage, nous remarquons un flou dans les deux images qui est totalement naturel et pourrait être supprimé avec des techniques de suppression du flou.

Pour plus d'efficacité, il est recommandé d'avoir un ensemble de données plus volumineux et un bon matériel capable de le traiter, notre manipulation sert uniquement à des fins de recherche mais pourrait être améliorée pour aider à servir un objectif plus important.

Conclusion

Dans ce chapitre, nous avons appris que les réseaux de neurones sont efficaces pour le débruitage et pourraient éventuellement être un bon moyen de traiter les images médicales endommagées à l'avenir.

Conclusion générale et perspectives

Conclusion générale et perspectives

Après avoir chargé notre ensemble de données dans le modèle d'apprentissage en profondeur, l'avoir prétraité et l'avoir exécuté à l'intérieur du système convolutif auto encoder et après de nombreuses simulations, l'auto encodeur était bien plus efficace que le filtre médian, ce qui en fait un outil fiable pour le débruitage, mais il pourrait toujours être amélioré et pour cela, nous avons quelques suggestions:

- La taille de l'ensemble de données est importante pour une meilleure formation (Il doit contenir au moins 200 éléments).
- Avoir un bon GPU accélérera certainement le processus d'entraînement (Une meilleure carte graphique accélérera le processus).
- Toutes les bibliothèques importées dans le modèle doivent être à jour (Certains outils ne fonctionneront pas à moins que la bibliothèque ne soit mise à jour).
- L'éditeur python utilisé doit fournir des lignes distinctives du code pour un espace de travail confortable.
- Assurez-vous que les éléments de votre ensemble de données ont à peu près les mêmes caractéristiques.
- Avant de commencer quoi que ce soit, préparez toujours ce dont vous avez besoin pour votre travail, vous ne voulez pas travailler dans un environnement désordonné ou un ordinateur lent, effectuez des routines de nettoyage du stockage, analysez et éliminez les virus...etc.
- Enregistrez toujours une copie de votre code source d'origine et enregistrez-la dans un lecteur flash ainsi que l'ensemble de données

Références bibliographiques

Bibliographie

[1] Pr Guy Frija, Pr Bernard Mazoyer, " L'imagerie médicale ", avril 2002.

[2] Hyunkyung Park and Toshironishimura, "reduced speckle noise on medical ultrasound images using cellular neural network with infinite impulse response filter ", Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE

[3] N. van Doremalen, T. Bushmaker, D.H. Morris, M.G. Holbrook, A. Gamble, B.N. Williamson, et al. Aerosol and surface stability of SARS-CoV-2 as compared with SARS-CoV-1. N Engl J Med, 382 (2020), pp. 1564-1567.

[4] McIntosh K. Coronavirus disease 2019 (COVID-19): Clinical features. In: Hirsch MS, Bloom A, ed. UpToDate. Waltham.

[5] W. Guan, Z. Ni, Y. Hu, W. Liang, C. Ou, J. He, et al. Clinical characteristics of coronavirus disease 2019 in China.

N Engl J Med, 382 (2020), pp. 1708-1720.

[6] Annarumma, Mauro, et al. "Automated triaging of adult chest radiographs with deep artificial neural networks." Radiology 291.1 (2019): 196-202.

[7] Mazurowski, Maciej A., et al. "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI." Journal of magnetic resonance imaging 49.4 (2019) : 939-954.

[8] Pr Guy Frija, Pr Bernard Mazoyer, " L'imagerie médicale ", avril 2002.

[9] M. GOUTAYER :'Nano-émulsions pour la vectorisation d'agents thérapeutiques ou diagnostiques ; étude de la biodistribution par imagerie de fluorescence in vivo', Thèse de doctorat, Spécialité Chimie Physique et Chimie Analytique, Université pierre et marie.curie, Paris, Décembre 2008.

[10] Flushing Hospital Medical Center. 2017. History of Medical Imaging – A Brief Overview. [online] Available at: <https://www.flushinghospital.org/newsletter/history-of-medical-imaging-a-brief-overview/> [Accessed 17 March 2022].

[11] nibib. (n.d.). Retrieved from https://www.nibib.nih.gov/science-education/science-topics/x-rays.

[12] Jean-Michel Morel - ENS Cachan : ‘ Du bruit dans les images’.

[13] S.S. AL-AMRI, N.V. KALYANKA et D A KHAMITKARS: ‘Comparative Study of Removal Noise from Remote Sensing Image’, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue. 1, No. 1, January 2010.

[14] I. SINGH et N. NEERU: ‘Performance Comparison of Various Image Denoising Filters Under Spatial Domain’, International Journal of Computer Applications (0975 – 8887) Volume 96-No.19, June 2014.7475.

[15] A. S. ZIANOU : ‘Evaluation de la qualité d’image’, Université de Mentouri – Constantine, Thèse doctorat ,2012.

[16] Yann LeCun, « L’apprentissage profond, une révolution en intelligence artificielle », La lettre du Collège de France, 41 | 2016, 13.

[17] IBM Cloud Education. (17 August 2020) "Recurrent neural networks." source: https://www.ibm.com/cloud/learn/neural-networks.

[18] McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics 5.4 (1943): 115-133.

[19] Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological review 65.6 (1958): 386.

[20] Werbos, Paul. "Beyond regression:" new tools for prediction and analysis in the behavioral sciences." Ph. D. dissertation, Harvard University (1974).

[21] LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." Neural computation 1.4 (1989): 541-551.

[22] Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation » [archive], sur DeepLearning 0.1, LISA Lab (consulté le 31 août 2013).

[23] Masakazu Matusugu, Katsuhiko Mori, Yusuke Mitari et Yuji Kaneda, « Subject independent facial expression recognition with robust face detection using a convolutional neural network », Neural Networks, vol. 16, no 5, 2003, p. 555–559 (DOI 10.1016/S0893-6080(03)00115-1, lire en ligne [archive], consulté le 17 novembre 2013).

[24] Yann LeCun, « LeNet-5, convolutional neural networks » [archive] (consulté le 16 novembre 2013).

[25] Aaron van den Oord, Sander Dieleman, Benjamin Schrauwen, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani et K. Q. Weinberger, Deep content-based music recommendation, Curran Associates, Inc., 2013, 2643–2651 p.

[26] Ronan Collobert et Jason Weston, « A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning », Proceedings of the 25th International Conference on Machine Learning, New York, NY, USA, ACM, iCML '08, 2008, p. 160–167 (ISBN 978-1-60558-205-4, DOI 10.1145/1390156.1390177, lire en ligne [archive]).

[27] D. H. Hubel et T. N. Wiesel, « Receptive fields and functional architecture of monkey striate cortex », The Journal of Physiology, vol. 195, no 1, 1er mars 1968, p. 215–243 (ISSN 0022-3751, PMID 4966457, PMCID 1557912.

[28] Autoencoder for Words, Liou, C.-Y., Cheng, C.-W., Liou, J.-W., and Liou, D.-R., Neurocomputing, Volume 139, 84–96 (2014).

[29] Dataset (2021) https://www.kaggle.com/pranavraikokte/covid19-image-dataset, COVID-19 Image Dataset, Pranav Raikote.