

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



**Mémoire de Fin d'études Master**

**Filière :** Informatique

**Option :** Systèmes Informatiques

**Thème :**

---

---

**Classification du COVID'19 en utilisant CNN  
dans les images X-ray**

---

---

**Encadré Par :**

**M. HALLACI Samir**

**Présenté par :**

**HAMLAOUI Mohammed  
Yazid**

**SEPTEMBRE 2021**

# Remerciement

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

En tout premier lieu, je remercie Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.

Le travail présenté dans cette mémoire a été réalisé au département d'informatique de la faculté MISM de l'Université de 08 mai 1945 Guelma, sous la direction M. HALLACI SAMIR, je voudrais saisir cette occasion et exprimer ma plus grande gratitude pour mon encadreur pour sa disponibilité et la confiance qu'il m'a accordée.

J'ai profité pendant longtemps du savoir-faire dont j'ai pu bénéficier au cours de nombreuses discussions et j'aimerais aussi le remercier pour l'autonomie qu'il m'a accordée, et ses précieux conseils qui m'ont permis de mener à bien ce travail.

Je tiens également à remercier les membres du jury de m'avoir fait l'honneur de juger mon travail.

Un grand merci également à tous nos enseignants au niveau du Département d'informatique pour leur dévouement et leur patience et leur contribution à notre formation.

Je remercie l'administration du département avec tous ses employés et ingénieurs.

Je remercie tous les employés et les ingénieurs du département d'informatique.

Enfin, je remercie ma famille, mes amis, mes collègues et toute la promotion de 2021.

# Dédicace

Je dédie ce travail à

Mes parents pour encouragements et toute l'aide qu'ils m'ont apportée durant mes études et pour les sacrifices qu'ils ont consentis pour mon instruction et mon bien-être.

À ma chère sœur qui m'a constamment encouragé et qui a cru en moi.

À ma grand-mère pour son souhait constant de Dieu pour la réussite et le succès dans mes études et ma vie.

À tous mes chers oncles et tantes et à toute ma famille.

À mes meilleurs amis et aux meilleurs jours universitaires.

Puisse Dieu leur accorder santé, bonheur, prospérité et longue vie afin que je puisse un jour combler de joie leur vie.

# Résumé

L'image médicale thoracique Rayon-X contient suffisamment d'informations pour déterminer si le patient est infecté par le virus Corona ou non. Les médecins et les experts dans ce domaine ont adopté cette méthode en raison de la difficulté de mener l'examen de réaction en chaîne polymérisée PCR dans la plupart des cas pour diverses raisons.

Le but de ce projet est de concevoir un système intelligent capable de distinguer les images médicales thoraciques afin d'aider les médecins et de classifier les images, comme dans notre cas, il est classifié en une image en deux classes (d'une personne normale ou une image d'une personne infectée par le virus Corona).

Ce système peut fonctionner en deux modes, le premier mode est normal en prenant l'image et en faisant une prédiction dessus et en donnant le résultat, tandis que le deuxième mode fonctionne en temps réel, car une fois que la caméra prend la radiographie, il est traité et donné le résultat.

Ce système fondé principalement sur l'amélioration de la qualité de l'image en équilibrant les couleurs pour extraire plus de détails, et sur les réseaux de neurones convolutifs, et grâce à l'expérience et sur la base des résultats extraits, nous avons pu construire un réseau de neurones convolutifs et l'entraîner à l'aide de la base de données que nous avons construite selon certains critères dont le plus important est d'utiliser la partie qui contient le poumon uniquement à partir de l'image et en abandonnant le reste de ses parties. Nous avons obtenu des résultats encourageants avec une précision de 98,98 % et un taux d'erreur d'environ 0,96 %.

**Mots-clés:** Convolutional neural networks, Models, Covid-19, Deep Learning, Datasets, Tensorflow, Keras, CNN, Coronavirus.

## ملخص:

تحتوي صورة الأشعة السينية الصدرية على معلومات كافية لتحديد ما إذا كان المريض مصاب بفيروس كورونا أو لا، فقد اعتمد الأطباء والخبراء في هذا المجال هذه الطريقة لصعوبة إجراء فحص تفاعل سلسلة البوليميراز PCR في أغلب الحالات لأسباب مختلفة.

الهدف من هذا المشروع هو تصميم نظام ذكي قادر على تمييز الصور الطبية الصدرية من اجل مساعدة الأطباء وتصنيفها، كما في حالتنا هذه يقوم بتصنيفها الى قسمين (صورة لشخص عادي او صورة لشخص مصاب بفيروس كورونا).

يمكن لهذا النظام أن يعمل في وضعيتين، الوضعية الاولى وهي العادية بأخذ الصورة وإجراء التنبؤ عليها وإعطاء النتيجة، أما الوضعية الثانية وهي العمل في الوقت الفعلي حيث انه بمجرد ان تلتقط الكاميرا صورة الأشعة السينية يقوم النظام بمعالجتها وإجراء التنبؤ وإعطاء النتيجة.

يعتمد هذا النظام بشكل أساسي على تحسين جودة الصورة بواسطة موازنة الالوان لاستخراج تفاصيل أكثر، وعلى الشبكات العصبية التلافيفية فمن خلال التجربة وبناء على النتائج المستخلصة فقد تمكنا من بناء شبكة عصبية التلافيفية ودرناها بواسطة قاعدة البيانات التي قمنا ببنائها وفقا لمعايير معينة من أهمها استعمال الجزء الذي يحتوي الرئة فقط من الصورة والتخلي عن باقي اجزائها الأخرى، فقد حصلنا على نتائج مشجعة لهذه الشبكة بدقة تبلغ % 98.98 ومعدل خطأ يقارب % 0.96.

الكلمات المفتاحية: Convolutional neural networks, Models, Covid-19, Deep Learning,

.Datasets, Tensorflow, Keras, CNN, Coronavirus

# Table des matières

<b>Liste des figures .....</b>	<b>1</b>
<b>Liste des tableaux.....</b>	<b>3</b>
<b>Introduction générale .....</b>	<b>4</b>
<b>Chapitre 1 : Généralités .....</b>	<b>6</b>
1. Introduction.....	6
2. Le système respiratoire .....	6
2.1. Les principales parties de système respiratoire.....	7
2.1.1. Le poumon droit.....	7
2.1.2. Le poumon Gauche .....	7
2.1.3. La trachée .....	7
2.1.4. Les alvéoles .....	8
3. Les virus.....	8
3.1. Classification des virus.....	8
3.2. Structure de virus .....	8
3.2.1. Nucléocapside tubulaire ou hélicoïdale.....	9
3.2.2. Nucléocapside polyédrique.....	9
3.2.3. Virus complexes.....	9
3.2.4. Virus enveloppés (Sphériques) .....	9
3.3. Coronavirus .....	10
3.3.1. Définition et propriétés.....	10
3.3.2. La maladie à coronavirus 2019 .....	10
3.3.3. Infection au niveau de la cellule .....	11
3.3.4. Infection au niveau des personnes et les symptômes .....	12
3.3.5. Diagnostique .....	12
4. L'image médicale rayons X.....	12
5. L'intelligence artificielle .....	14
6. L'apprentissage automatique .....	15
6.1. L'apprentissage supervisé.....	15
6.2. L'apprentissage non supervisé.....	16
6.3. L'apprentissage par renforcement .....	17
7. L'apprentissage en profondeur .....	18
7.1. Les neurones biologiques .....	18
7.2. Les réseaux de neurones artificiels (ANN).....	19
7.3. Les types des réseaux de neurones artificiels.....	20
7.4. Réseau de neurones convolutifs (CNN) .....	22

7.4.1.	Évolution architecturale des CNN profonds .....	22
7.4.1.1.	Stagnation de CNN : Début 2000.....	23
7.4.1.2.	Le renouveau de CNN : 2006-2011 .....	23
7.4.1.3.	L'ascension de CNN : 2012-2014 .....	24
7.4.1.4.	Augmentation rapide des innovations architecturales et des applications du CNN : 2015- Aujourd'hui .....	24
7.4.2.	Innovations architecturales dans le CNN .....	24
7.4.2.1.	CNNs basés sur l'exploitation spatiale .....	25
7.4.2.2.	CNNs basés sur la profondeur .....	25
7.4.2.3.	CNNs basés sur les trajets multiples .....	26
7.4.2.4.	CNNs multi-connexions basés sur la largeur.....	27
7.4.2.5.	CNN basés sur l'exploitation des cartes de caractéristiques .....	27
7.4.2.6.	CNNs basés sur l'exploitation des canaux (entrées).....	28
7.4.2.7.	CNN basés sur l'attention .....	28
7.4.3.	Comparaison entre tous les architecture CNNs dans tous les catégories.....	29
7.5.	Réseau de neurones récurrents (RNN).....	31
7.6.	Réseau Long short-term memory (LSTM) .....	31
7.7.	Domaines d'application des réseaux de neurones artificiels .....	32
8.	Les travaux connexes pour la classification de covid-19.....	32
9.	Les défis sur Kaggle .....	33
10.	Les Base de données existant.....	35
11.	Conclusion .....	37
<b>Chapitre 2 : Les réseaux neuronaux convolutifs.....</b>		<b>38</b>
1.	Introduction.....	38
2.	Cortex visuel du cerveau et la vision par ordinateur .....	38
3.	L'architecture d'un réseau de neurones convolutifs .....	40
3.1.	La couche d'entré .....	40
3.2.	La couche de convolution.....	41
3.2.1.	Les filtres .....	42
3.2.2.	Le Pas (Strides) .....	42
3.2.3.	Le Rembourrage (Padding).....	43
3.3.	Fonction d'activation ReLu (unité linéaire rectifiée).....	44
3.4.	L'opération de mise en commun (Pooling) .....	44
3.4.1.	Max pooling.....	45
3.4.2.	Average pooling.....	45
3.4.3.	Global pooling .....	46
3.4.4.	Spatial pyramid pooling .....	46
3.4.5.	Cascade Corner Pooling .....	47

3.5.	La couche entièrement connectée (Fully Connected Layer).....	48
3.6.	La fonction softmax .....	48
3.7.	La couche de sortie.....	49
3.8.	La normalisation .....	49
3.8.1.	Méthode de normalisation .....	49
3.9.	Régularisation .....	50
3.9.1.	La dilution (Dropout) .....	50
3.9.2.	Method de régularisation .....	51
3.10.	Optimisation .....	51
3.10.1.	Méthode d'optimisation.....	52
3.10.2.	Méthodes d'optimisation les plus utilisées .....	53
4.	L'augmentation des données .....	53
5.	Le sur-apprentissage et le sous- apprentissage dans le CNN .....	54
5.1.	Le sur-apprentissage.....	54
5.2.	Méthodes pour éviter le sur-apprentissage .....	54
5.3.	Le sous-apprentissage.....	55
5.4.	Méthodes pour éviter le sous-apprentissage .....	55
6.	Framework plus utilisées pour l'implémentation d'un CNN .....	56
7.	Conclusion .....	57
<b>Chapitre 3 : La conception .....</b>		<b>58</b>
1.	Introduction.....	58
2.	La recherche des bases de données confirmée par PCR et la préparation de notre base de donnée .....	59
2.1.	Le filtrage et nettoyage des bases des donnes.....	59
2.2.	La création d'une base de données .....	60
2.3.	L'application de la balance des couleurs pour l'amélioration de la qualité de l'image. 60	
2.3.1.	L'algorithme de balance des couleurs .....	61
2.4.	La détection des poumons et la sélection des zones requises .....	61
2.5.	L'obtention des images finales et construction de la base de données finale.....	62
2.6.	La division de la base de données .....	62
2.7.	L'augmentation des données .....	63
3.	La recherche d'un modèle adapté.....	64
3.1.	Expérimentation de modèles existants .....	64
3.2.	La création de notre modèle.....	64
4.	L'entraînement et la préparation du modèle .....	67
5.	Estimation de la performance du modèle.....	69
5.1.	La matrice de confusion .....	70



5.2.	La fonction de l'efficacité du récepteur.....	71
5.3.	Classification de Covid-19 en temps réel.....	71
5.4.	Généralisation des résultats sur des bases externes.....	72
5.4.1.	Les tests avec la base de donnée COVID-19 Radiography Database .....	72
5.4.2.	Les tests avec la base de donnée COVID-19 chest xray .....	73
5.4.3.	Les tests avec la base de donnée Covid-19 Image Dataset .....	73
5.4.4.	Les tests avec la base de donnée Chest Xray for covid-19 detection .....	73
5.4.5.	Les tests avec la base de donnée Covid Patients Chest X-Ray .....	74
6.	Conclusion .....	74
<b>Chapitre 4 : L'implémentation .....</b>		<b>75</b>
1.	Introduction.....	75
2.	Environnement de développement et matériel .....	75
2.1.	L'ordinateur et le système .....	75
2.2.	Google colab.....	75
2.2.1.	Les types des GPU disponible dans Google colab.....	76
2.3.	Pycharm.....	76
2.4.	Qt Designer .....	76
3.	Langage de programmation, Framework et bibliothèques.....	77
3.1.	Python.....	77
3.2.	TensorFlow .....	77
3.3.	Keras .....	77
3.4.	NumPy.....	77
3.5.	OpenCv et PIL .....	78
3.6.	Matplotlib .....	78
3.7.	Scikit-learn.....	78
3.8.	FPDF .....	78
4.	Présentation d'application.....	78
5.	Présentation détaillée d'application.....	79
5.1.	La première page.....	80
5.2.	La deuxième page .....	82
5.3.	La troisième page.....	84
5.4.	La quatrième page.....	85
6.	Les points forts de notre système .....	89
7.	Les points faibles de notre système .....	89
8.	Les difficultés que nous avons rencontrées.....	89
9.	Les perspectives pour le développement du système .....	90
10.	Conclusion .....	90

<b>Conclusion générale .....</b>	<b>91</b>
<b>Bibliographie.....</b>	<b>92</b>

# Liste des figures

<b>Chapitre 1 : Généralités .....</b>	<b>6</b>
Figure 1.1 : Poumon humain .....	6
Figure 1.2 : Diagramme schématique de poumon humain .....	7
Figure 1.3 : types des virus .....	10
Figure 1.4 : Images schématique du virus Corona .....	10
Figure 1.5 : Micrographie du virus Corona .....	11
Figure 1.6 : Diagramme schématique de la surface du virus Corona.....	11
Figure 1.7 : Image médicale rayons X .....	13
Figure 1.8 : Représentation schématique d'un assemblage de tube à rayons X .....	13
Figure 1.9 : Croquis d'une cassette de film radiographique .....	13
Figure 1.10 : Schéma d'organisation d'intelligence artificielle .....	14
Figure 1.11 : Schéma d'organisation d'apprentissage automatique "machine learning" .....	15
Figure 1.12 : Schéma d'apprentissage supervisé "supervised learning" .....	16
Figure 1.13 : Schéma d'apprentissage non supervisé "unsupervised learning" .....	17
Figure 1.14 : Schéma d'apprentissage par renforcement "Reinforcement learning" .....	17
Figure 1.15 : Schéma des neurones biologiques .....	19
Figure 1.16 : Schéma des réseaux de neurones artificiels .....	21
Figure 1.17 : Histoire de l'évolution des CNN profonds .....	23
Figure 1.18 : Taxonomie des architectures CNN profondes montrant sept catégories différentes ....	25
<b>Chapitre 2 : Les réseaux neuronaux convolutifs.....</b>	<b>38</b>
Figure 2.1 : Les lobes-cérébraux.....	39
Figure 2.2 : Valeurs des couleurs RGB de certains pixels de l'image de Lena.....	40
Figure 2.3 : L'architecture d'un réseau de neurones convolutifs simple .....	40
Figure 2.4 : Les trois dimensions d'image .....	40
Figure 2.5 : Matrice d'entrée avec le filtre de convolution .....	41
Figure 2.6 : Première étape de convolution .....	41
Figure 2.7 : Deuxième étape de convolution .....	42
Figure 2.8 : Vecteur de caractéristique après l'application de convolution .....	42
Figure 2.9 : Exemple d'application des filtres .....	42
Figure 2.10 : Exemple d'application de pas "stride" .....	43
Figure 2.11 : Exemple d'application de rembourrage.....	43
Figure 2.12 : Graphique de la Fonction d'activation ReLu.....	44
Figure 2.13 : L'application de max polling .....	45
Figure 2.14 : L'application de average polling.....	46
Figure 2.15 : L'application de global average pooling.....	46

Figure 2.16 : L'application de global max pooling .....	46
Figure 2.17 : L'architecture de Spatial pyramid pooling.....	47
Figure 2.18 : Diagramme de Cascade Corner Pooling .....	47
Figure 2.19 : Utilisation dans le temps des méthodes de pooling les plus connues .....	48
Figure 2.20 : L'architecture de la couche entièrement connectée.....	48
Figure 2.21 : la fonction softmax .....	49
Figure 2.22 : L'application de dilution.....	51
Figure 2.23 : utilisation dans le temps des optimiseurs les plus connues .....	52
Figure 2.24 : L'augmentation d'image.....	53
Figure 2.25 : Graphiques de sur-apprentissage et le sous-apprentissage et l'optimale fitting .....	54
<b>Chapitre 3 : La conception .....</b>	<b>58</b>
Figure 3.1 : Diagramme principale de la conception .....	58
Figure 3.2 : Diagramme de collecte de base de données .....	59
Figure 3.3 : Diagramme de vérification et confirmation de la qualité des images.....	60
Figure 3.4 : diagramme global de notre base de données.....	60
Figure 3.5 : Balançage des couleurs.....	61
Figure 3.6 : Les principales étapes d'obtention de l'image finale.....	62
Figure 3.7 : diagramme détaillé de notre base de données .....	63
Figure 3.8 : l'augmentation des données .....	64
Figure 3.9 : diagramme principale de notre CNN.....	66
Figure 3.10 : L'accuracy de model .....	69
Figure 3.11 : loss de model.....	69
Figure 3.12 : matrice de confusion.....	71
Figure 3.13 : fonction d'efficacité du récepteur.....	71
<b>Chapitre 4 : L'implémentation .....</b>	<b>75</b>
Figure 4.1 : L'interface principale de l'application.....	79
Figure 4.2 : La première page d'application.....	80
Figure 4.3 : La première page d'application pendant l'exécution.....	81
Figure 4.4 : La deuxième page d'application .....	82
Figure 4.5 : La deuxième page d'application pendant l'exécution .....	83
Figure 4.6 : La troisième page d'application .....	84
Figure 4.7 : La troisième page d'application pendant l'exécution.....	85
Figure 4.8 : La quatrième page d'application .....	86
Figure 4.9 : La quatrième page d'application pendant l'exécution.....	87
Figure 4.10 : Exemple de compte rendu d'un malade.....	88

## Liste des tableaux

<b>Chapitre 1 : Généralités .....</b>	<b>6</b>
Tableau 1.1: Comparaison des performances des architectures récentes de différentes catégories...	29
Tableau 1.2: les domaines d'application des réseaux de neurones artificiels .....	32
Tableau 1.3: La plupart des défis de covid-19 on Kaggle .....	34
Tableau 1.4: La plupart des datasets de covid-19 on Kaggle .....	36
<b>Chapitre 2 : Les réseaux neuronaux convolutifs.....</b>	<b>38</b>
Tableau 2.1: Les méthodes de normalisation les plus utilisé.....	49
Tableau 2.2: Méthodes de régularisation ordonnée par leur nombre d'utilisation.....	51
Tableau 2.3: Avantages et désavantages des algorithmes d'optimisations populaire .....	52
Tableau 2.4: Méthodes d'optimisation les plus utilisées .....	53
Tableau 2.5: Top huit Framework utilise dans l'apprentissage en profondeur avec leur point fort...	56
<b>Chapitre 3 : La conception .....</b>	<b>58</b>
Tableau 3.1 : pourcentages utilisés pour la division de la base de données .....	63
Tableau 3.2 : résultat de rapport de classification.....	70
Tableau 3.3 : le temps de chaque opération en temps réel .....	72
Tableau 3.4 : Les tests avec la base de donnée COVID-19 Radiography Database .....	72
Tableau 3.5 : Les tests avec la base de donnée COVID-19 chest xray .....	73
Tableau 3.6 : Les tests avec la base de donnée Covid-19 Image Dataset .....	73
Tableau 3.8 : Les tests avec la base de donnée Chest Xray for covid-19 detection .....	74
Tableau 3.8 : Les tests avec la base de donnée Covid Patients Chest X-Ray .....	74

# Introduction générale

## Introduction générale

Le virus Corona ou Covid-19 est un virus qui a évolué et s'est transmis à l'homme par l'intermédiaire d'un animal (chauve-souris), et ce virus est considéré comme beaucoup plus dangereux que le virus de la grippe, et la raison en est qu'il cible le système respiratoire humain et le détruit efficacement et dans un pourcentage élevé qui peut parfois entraîner la mort.

Il existe plusieurs façons de détecter ce virus dans le corps humain, et la plupart de ces méthodes sont biologiques et chimiques et prennent beaucoup de temps, jusqu'à plusieurs jours, pour obtenir le résultat, ce qui entraîne la détérioration de l'état du patient s'il n'est pas traité à temps, pour cette raison.

Les médecins ont adopté une autre méthode de détection, qui est la radiographie-RayonX du thorax, pour prendre une décision d'où le rôle de l'intelligence artificielle et de l'apprentissage profond pour détecter et aider de manière automatisée et plus rapide.

L'objectif principal de ce projet est de construire un système capable de classer les images médicales et de donner des décisions correctes avec un grand pourcentage. Tout cela dépend de la qualité des données existantes, du réseau neuronal convolutif, du nombre de ses couches et de ses coefficients, et des couches utilisées pour traiter les données nécessaires à l'entraînement du réseau.

Tous ces éléments sont traités dans ce projet à travers quatre chapitres principaux :

## **Chapitre 1 : Généralités**

Dans ce chapitre, nous apprendrons les principes de base de la structure des virus et du système respiratoire, en passant par les concepts généraux de l'intelligence artificielle, ses divisions, ses applications, l'historique et les types des CNN, les travaux connexes, les bases de données les plus célèbres dans ce domaine et les défis les plus célèbres.

## **Chapitre 2 : Les réseaux neuronaux convolutifs**

Dans ce chapitre, nous en apprendrons plus sur l'architecture d'un réseau neuronal convolutif et sur toutes ses propriétés telles que les couches, les fonctions, les optimiseurs et tous les paramètres qui contrôlent la qualité du réseau.

## **Chapitre 3 : La conception**

Dans ce chapitre, nous présentons la conception détaillée de ce projet pour un système de haute qualité et toutes les méthodes utilisées et les paramètres sélectionnées.

## **Chapitre 4 : L'implémentation**

Dans ce chapitre, nous avons abordé l'aspect d'implémentation de ce projet et de tous les moyens qui ont été adoptés pour réaliser cette application, depuis l'environnement de développement et les bibliothèques utilisées jusqu'aux détails des tests et des résultats obtenus.



# Chapitre 01

# Généralités

## 1. Introduction

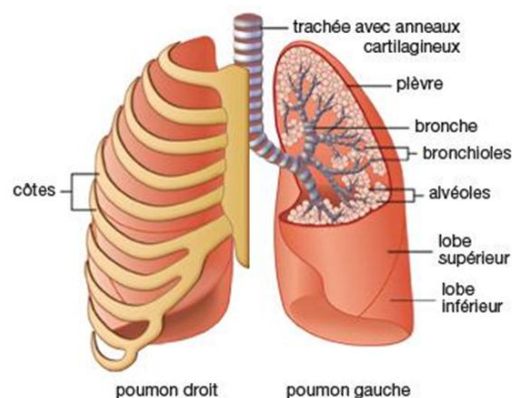
Le corps humain est une chose très complexe de la création de Dieu. Ce corps se compose de nombreux appareils. Ces appareils fonctionnent les uns avec les autres avec une grande précision dans la mesure où lorsqu'un appareil est affecté d'autres organes peuvent être affectés avec lui.

Ces effets peuvent être causés par des causes physiques telles que des accidents et des blessures, ou par des raisons biologiques telles que des virus et des bactéries, provenant de ces appareils sensibles qui sont au centre de nos recherches « le système respiratoire ».

Récemment, un nouveau virus connu sous le nom de COVID-19 est apparu, C'est un virus nocif pour les poumons. Dans ce chapitre nous allons essayer d'identifier ce virus, le système respiratoire, et les défis scientifiques les plus importants dans le domaine du machine Learning et du deep Learning qui tentent de résoudre cette crise et aider les médecins à détecter et classer ce virus.

## 2. Le système respiratoire

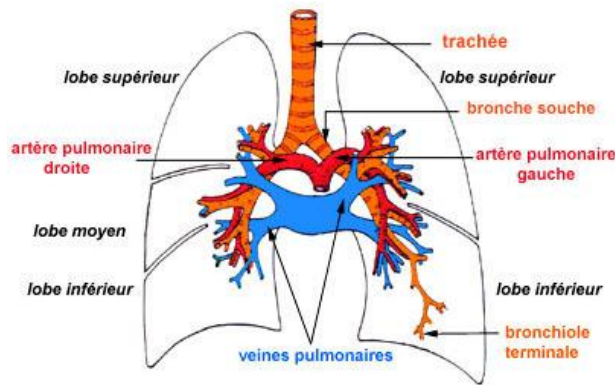
Le système respiratoire (également appareil respiratoire, système ventilatoire) est un système biologique composé d'organes et de structures spécifiques utilisés pour les échanges gazeux chez l'être humain [1].



*Figure 1.1 : Poumon humain*

L'être humain possède deux poumons, logés dans la cage thoracique et protégés par les côtes. Ils reposent sur une vaste nappe musculaire, le diaphragme, qui sépare le thorax de la cavité abdominale [1].

Les poumons sont les organes responsables de la respiration. Ils transmettent l'oxygène dans le sang et éliminent le gaz carbonique. L'air inspiré pénètre par la trachée et s'introduit dans les bronches qui se subdivisent en bronches toujours plus petites puis en bronchioles, débouchant dans les alvéoles [1].



*Figure 1.2 : Diagramme schématisique de poumon humain*

## 2.1. Les principales parties de système respiratoire

### 2.1.1. Le poumon droit

Le poumon droit comprend trois lobes (supérieur, moyen, inférieur) par deux scissures (grande et petite) dont la paroi est tapissée par le feuillet viscéral de la plèvre [2].

### 2.1.2. Le poumon Gauche

Le poumon gauche Divisé en deux lobes (supérieur et inférieur) par une grande scissure, il n'existe pas de segment para cardiaque à gauche. Lobe supérieur et inférieur à chacun desquels correspond une bronche lobaire [2].

### 2.1.3. La trachée

Se divise en deux bronches souches, la bronche souche droite (BSD) plus verticale que bronche souche gauche (BDG), L'angle formé par la réunion des 2 bronches s'appelle la bifurcation trachéale ou la carène.

Les bronches souches ou principales se divisent en bronches lobaires qui vont elles se diviser en bronches segmentaires, sous segmentaires jusqu'aux canaux alvéolaires [2].

#### **2.1.4. Les alvéoles**

Les alvéoles sont de petits « sacs » poreux, en forme de bulle, où se déroulent les échanges gazeux avec le sang [2].

Chaque alvéole est entourée de vaisseaux très fins, appelés capillaires, dont elle est séparée par une membrane. C'est à travers cette membrane que l'oxygène est transféré de l'air vers le sang, et en sens inverse, le gaz carbonique éliminé. Les alvéoles ont une surface totale de 80m<sup>2</sup> environ. Les contractions musculaires du diaphragme et de la cage thoracique permettent aux poumons de se gonfler et de se dégonfler, par augmentation / diminution du volume de la cage thoracique [2].

### **3. Les virus**

Agent infectieux de petite taille et de composition simple occupent une position taxonomique particulière : ce ne sont pas des plantes, des animaux ou des bactéries procaryotes (organismes unicellulaires sans noyau défini), et ils sont généralement placés dans leur propre règne. En fait, les virus ne devraient même pas être considérés comme des organismes, au sens le plus strict, car ils ne peuvent se multiplier que dans une cellule hôte (des cellules vivantes d'animaux, de plantes ou de bactéries) [3].

#### **3.1. Classification des virus**

La classification des virus est basée sur le type de génome (le type d'acide nucléique) des virus (ADN ou ARN) et son mode d'expression.

Les acides nucléiques des virions sont organisés en génomes. Tous les virus à ADN double brin consistent en une seule grosse molécule, alors que la plupart des virus à ARN double brin ont des génomes segmentés, chaque segment représentant généralement un seul gène qui code les informations pour synthétiser une seule protéine. Les virus à ADN génomique simple brin sont généralement de petite taille, avec une information génétique limitée. Certains virus à ADN simple brin sont composés de deux populations de virions, chacune constituée d'ADN simple brin complémentaire de polarité opposée à celle de l'autre [3].

#### **3.2. Structure des virus**

Le génome est emballé dans une structure protéique appelée CAPSIDE, d'un mot grec, capsas, signifiant boîte. La capsidie protège le génome. Elle a une conformation géométrique

qui, selon les virus est, soit tubulaire, soit polyédrique. On appelle nucléocapside la structure compacte formée par l'assemblage de la capsid autour du génome [4].

Il faut retenir que les capsides, tubulaires comme polyédriques, sont faites de protéines virales polymérisées, les virus ayant trop peu de gènes pour s'offrir autant de protéines distinctes qu'il leur en faut pour recouvrir et protéger le génome, et que ces structures ont été sélectionnées dans la nature en raison de leur grande stabilité [4].

### **3.2.1. Nucléocapside tubulaire ou hélicoïdale**

C'est un tube enroulé en peloton (pour ce qui concerne les virus humains ou animaux, ce peloton est lui-même enveloppé dans un 3e élément appelé péplos) [4].

### **3.2.2. Nucléocapside polyédrique**

Ce n'est pas n'importe quel polyèdre mais un ICOSAÈDRE : polyèdre à 20 faces qui sont des triangles équilatéraux, et 12 sommets. Vu sous un certain angle, l'icosaèdre présente un contour hexagonal [4].

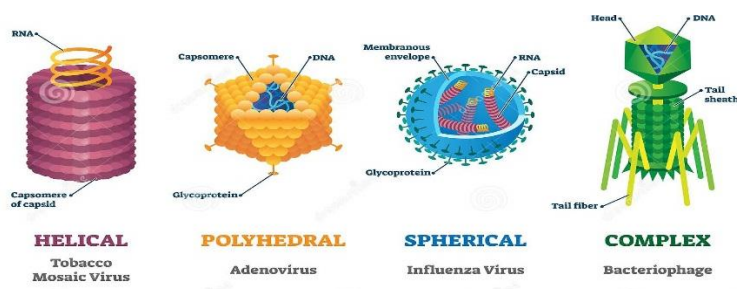
### **3.2.3. Virus complexes**

Ces virus possèdent une capsid symétrique qui n'est ni hélicoïdale, ni d'autre. Les bactériophages sont des virus complexes possédant une tête icosaédrique liée à une queue hélicoïdale à laquelle sont attachés des poils et des fibres caudales [4].

### **3.2.4. Virus enveloppés (Sphériques)**

En plus de la capsid, certains virus sont capables de s'entourer d'une structure membranaire empruntée à la cellule hôte. Cette enveloppe membranaire est composée d'une bicouche lipidique qui peut posséder des protéines codées par le génome viral ou le génome de l'hôte. Cette enveloppe donne quelques avantages aux virions par rapport à ceux composés d'une capsid seule, comme la protection vis-à-vis d'enzymes ou de composés chimiques [4].

Le virus de la grippe et le virus de corona sont des exemples de virus enveloppés.



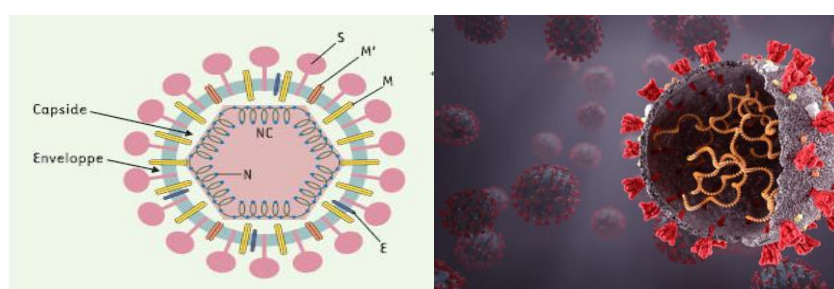
*Figure 1.3 : types des virus.*

### 3.3. Coronavirus

#### 3.3.1. Définition et propriétés

Les coronavirus sont un type de virus issus de la famille des *coronaviridae*, dont la particularité est de posséder un génome à ARN très long (l'acide ribonucléique permet le transport du message génétique et la synthèse des protéines). Ces germes sont entourés d'une capsule de protéines (appelées spicules (Spike, S)) en forme de couronne, d'où leur nom [5].

Plusieurs coronavirus sont déjà connus pour être capables d'infecter les humains : trois coronavirus saisonniers responsables de symptômes hivernaux sans gravité (rhumes), le SARS-CoV responsable du syndrome respiratoire aigu sévère (SRAS) entre novembre 2002 et juillet 2003, le MERS-CoV responsable d'une atteinte respiratoire potentiellement sévère (Middle East Respiratory Syndrome) identifié en 2012 et très récemment le SARS-CoV 2 qui est responsable de la maladie Covid-19[6].

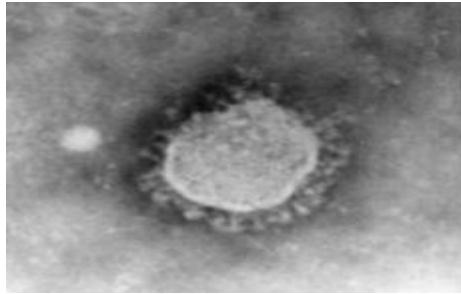


*Figure 1.4 : Images schématique du virus Corona*

#### 3.3.2. La maladie à coronavirus 2019

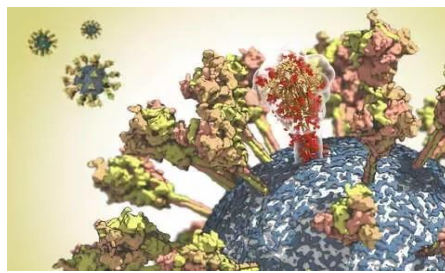
La maladie à coronavirus 2019 (COVID-19) est une maladie infectieuse due au SARS-CoV 2 virus. Son réservoir naturel semble être la chauve-souris [8]. Il se transmet principalement par des gouttelettes produites lorsqu'une personne infectée tousse,

éternue, ou lors d'une expiration. Ces gouttelettes sont trop lourdes pour rester dans l'air et tombent rapidement sur le sol ou sur toute surface proche. [7]



*Figure 1.5 : Micrographie du virus Corona*

Le SARS-CoV 2 est un virus enveloppé à ARN simple-brin linéaire de la famille des bêta coronavirus. Il possède une protéine de surface présentant une différence notable par rapport à d'autres coronavirus génétiquement proches. Cette différence serait importante pour l'évolution et l'émergence du virus [8].



*Figure 1.6 : Diagramme schématique de la surface du virus Corona*

### **3.3.3. Infection au niveau de la cellule**

Le virus est un pathogène intracellulaire obligatoire, et doit pénétrer dans une cellule hôte pour pouvoir se multiplier. La première étape de ce processus est donc l'entrée du matériel viral dans le cytoplasme après avoir franchi la membrane cellulaire. L'étape d'entrée débute par l'attachement de la particule virale à la surface de la cellule. Celle-ci repose sur l'interaction entre les spicules à la surface de la particule virale (protéine S du SARS-CoV-2) et la glycoprotéine angiotensine-convertant enzyme 2 (ACE2) qui agit en tant que récepteur d'entrée [9].

Après fixation à ACE2, le spicule virale (S) est coupé en deux parties par une protéase (enzyme qui coupe les protéines) de la cellule hôte. Cet événement moléculaire est nécessaire pour exposer une partie de la séquence polypeptidique de S. S'ensuit un

rapprochement entre l'enveloppe du virus et la membrane cellulaire, toutes deux formées par une bicouche lipidique qui fusionneront. Donc la molécule TMPRSS2 qui est présente à la surface de la cellule permet la fusion du virus avec la membrane plasmique de la cellule hôte. Le virus peut également entrer par « endocytose » : la fixation de Spike à ACE2 va induire une invagination de la membrane plasmique, englobant le virus qui rentre dans un « endosome » où une protéase, activée par l'acidité de ce compartiment, permettra de déclencher la fusion entre la membrane endosomale et la membrane virale. La fusion entre les membranes cellulaires et virales libère l'ARN viral dans le cytoplasme cellulaire où se met en place la réplication du virus (Figure 1.5) [9].

#### **3.3.4. Infection au niveau des personnes et les symptômes**

Après une période d'incubation d'environ cinq jours, 70 % des patients infectés développent une toux, de la fièvre ou un essoufflement. Ce stade d'invasion virale est suivi, chez certains patients, d'une réaction immunitaire insuffisante caractérisée par une aggravation des symptômes respiratoires, et un syndrome inflammatoire, généralement huit à dix jours après les premiers symptômes. Dans les cas plus graves, l'infection peut provoquer une pneumonie, un syndrome respiratoire aigu sévère, une insuffisance rénale et même la mort [10].

#### **3.3.5. Diagnostique**

Parmi les tests diagnostiques qui nous permettent de confirmer le COVID, la réaction de transcription inverse suivie d'une réaction de polymérisation en chaîne quantitative en temps réel (RT-qPCR), et le test de diagnostic rapide basé sur la détection de l'antigène spécifique du SARS-CoV-2 sont deux méthodes utilisées dans la phase précoce des manifestations infectieuses. Les tests de détection des anticorps sériques (ELISA et test de flux latéral) sont utilisés dans la phase ultérieure et après la guérison [7]. Et on a aussi une autre méthode qui se base sur la lecture des images à rayons X, sur la cage thoracique des patients.

### **4. L'image médicale rayons X**

L'imagerie par rayons X est la plus ancienne modalité d'imagerie médicale, qui a fait son entrée dans la pratique médicale peu après la découverte des rayons X en 1895. L'imagerie radiologique est une technique de projection, et la formation de l'image se fait

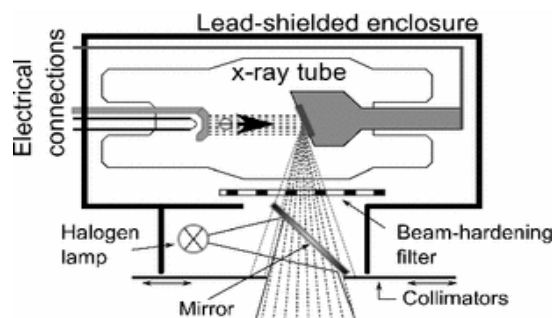


traditionnellement sur un film photosensible, bien que l'imagerie radiologique numérique directe devienne de plus en plus courante [13].

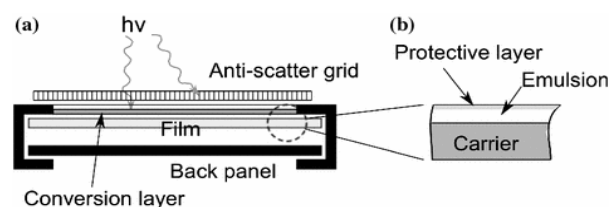


**Figure 1.7 :** Image médicale rayons X

Les rayons X sont générés lorsque des électrons cinétiques interagissent avec une cible solide et subissent une décélération soudaine ("rayonnement de freinage" ou "bremsstrahlung" prononcé en allemand, de bremsen « freiner » et Strahlung « radiation») [13].



**Figure 1.8 :** Représentation schématique d'un assemblage de tube à rayons X avec son boîtier et ses caractéristiques supplémentaires



**Figure 1.9 :** (a) Croquis d'une cassette de film radiographique et (b) film photosensible

Ces dernières années, l'imagerie radiologique a atteint les limites technologiques. La tendance à abandonner les films au profit de l'imagerie radiologique numérique se poursuit.

Des détecteurs améliorés et plus sensibles permettent de réduire encore le temps d'exposition et la dose de rayonnement reçue par le patient. Après que des inquiétudes aient été soulevées quant au fait que l'utilisation accrue des rayons X dans les procédures de diagnostic et d'intervention pourrait entraîner un risque élevé de cancer [18].

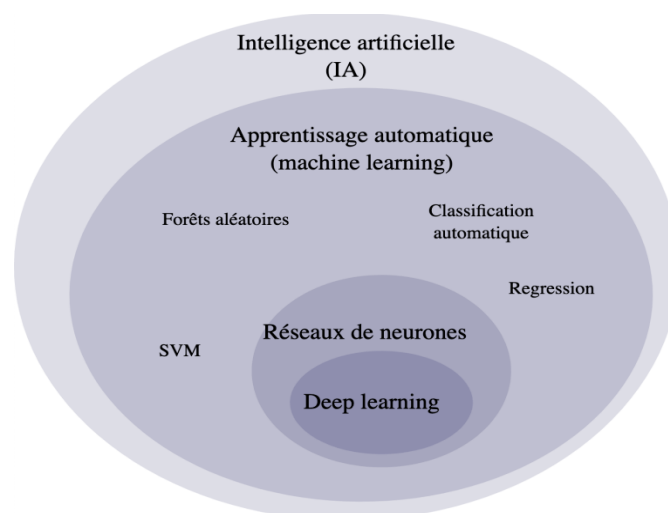
Grâce au développement de techniques d'acquisition d'images radiologiques permettant de recueillir facilement le plus grand nombre d'informations médicales sur le patient, des quantités suffisantes de ces images sont devenues disponibles, ce qui permet d'utiliser l'intelligence artificielle et l'apprentissage profond dans ce domaine.

## 5. L'intelligence artificielle

L'intelligence artificielle est la science qui s'intéresse à la construction des machines et des programmes intelligents capables de simuler l'intelligence humaine en développant des techniques et des théories et des méthodes qui simulent la pensée de l'esprit humain [11].

Elle englobe un ensemble de concepts et de techniques plus qu'une discipline autonome constituée [12].

L'intelligence artificielle est classée dans le groupe des sciences cognitives. Elle fait appel aux neurosciences computationnelles en plus de la logique mathématique et de l'informatique, car elle repose sur l'intervention et la programmation humaines, et utilise des méthodes pour résoudre des problèmes présentant un haut degré de complexité logique et algorithmique.



*Figure 1.10 : Schéma d'organisation d'intelligence artificielle*

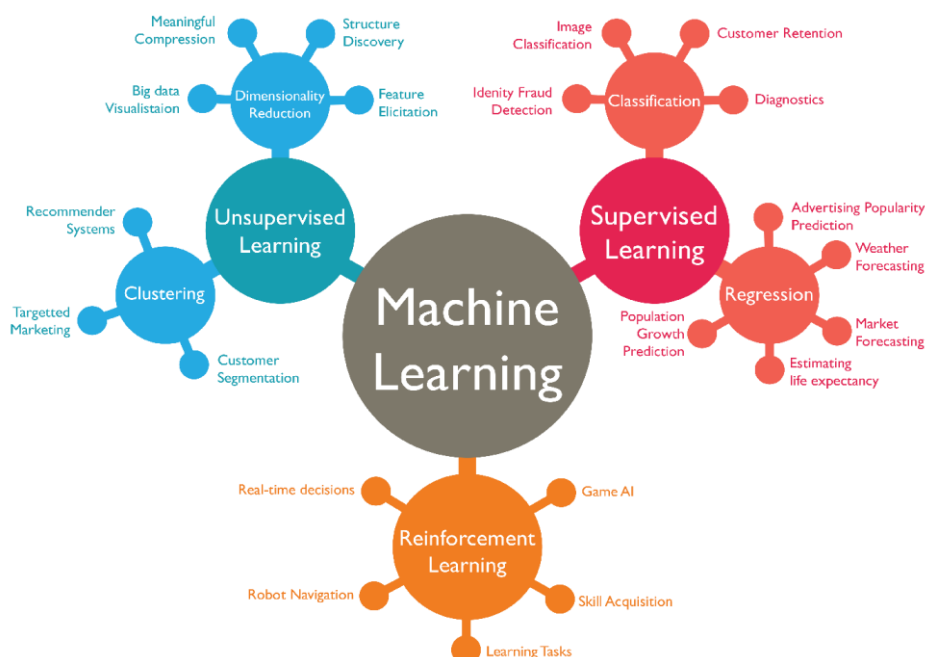
## 6. L'apprentissage automatique

La science de l'apprentissage automatique est établie depuis que le scientifique britannique Alan Turing et son équipe ont inventé la première machine à décoder Unigma.

L'apprentissage automatique est l'étude des algorithmes informatiques qui s'améliorent automatiquement grâce à l'expérience acquise en utilisant des données [14].

L'apprentissage automatique est considéré comme une partie de l'intelligence artificielle. Les algorithmes d'apprentissage automatique adoptent un modèle basé sur des échantillons de données appelés données d'apprentissage, afin de faire des prédictions et de prendre des décisions sans être explicitement programmés pour le faire [15].

Un sous-ensemble de l'apprentissage automatique étroitement lié aux statistiques informatiques, se concentre sur la réalisation de prédictions à l'aide d'ordinateurs. Mais l'apprentissage automatique n'est pas toujours un apprentissage statistique.



*Figure 1.11 : Schéma d'organisation d'apprentissage automatique "machine learning"*

### 6.1. L'apprentissage supervisé

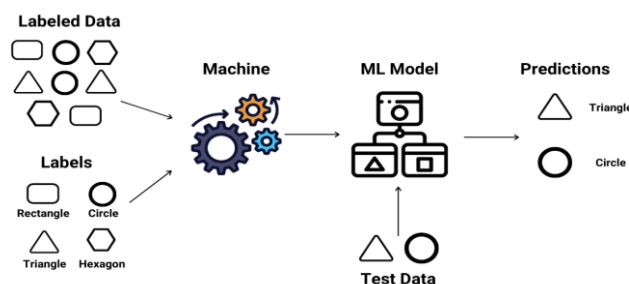
Les algorithmes d'apprentissage supervisé construisent un modèle mathématique d'un ensemble de données qui contient à la fois les entrées et les sorties souhaitées [14].

Les données sont appelées données d'apprentissage et consistent en un ensemble d'exemples d'apprentissage.

Chaque exemple d'apprentissage possède une ou plusieurs entrées et la sortie souhaitée, également appelée signal de supervision. Dans le modèle mathématique, chaque exemple d'apprentissage est représenté par un tableau ou un vecteur, parfois appelé vecteur de caractéristiques, et les données d'apprentissage sont représentées par une matrice.

Grâce à l'optimisation itérative d'une fonction objective, les algorithmes d'apprentissage supervisé apprennent une fonction qui peut être utilisée pour prédire la sortie associée à de nouvelles entrées [16].

Une fonction optimale permettra à l'algorithme de déterminer correctement la sortie pour les entrées qui ne faisaient pas partie des données d'apprentissage. On dit qu'un algorithme qui améliore la précision de ses sorties ou de ses prédictions au fil du temps a appris à effectuer cette tâche [17].

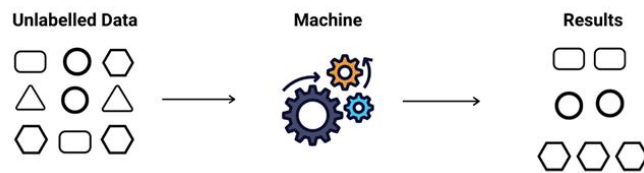


*Figure 1.12 : Schéma d'apprentissage supervisé "supervised learning"*

## 6.2. L'apprentissage non supervisé

Les caractéristiques des algorithmes non supervisés sont qu'ils n'apprennent que des données de test qui n'ont pas été étiquetées, ils prennent un ensemble de données qui ne contient que l'entrée et trouve une structure dans les données comme le regroupement ou le groupement de points de données. Au lieu de répondre aux commentaires, les algorithmes d'apprentissage non supervisé identifient les points communs dans les données et réagissent en fonction de la présence ou de l'absence de ces points communs dans chaque nouvelle donnée. Une application centrale de l'apprentissage non supervisé est dans le domaine de l'estimation de la densité en statistique, comme la recherche de la fonction de densité de probabilité [20].

Bien que l'apprentissage non supervisé englobe d'autres domaines impliquant la synthèse et l'explication des caractéristiques des données.

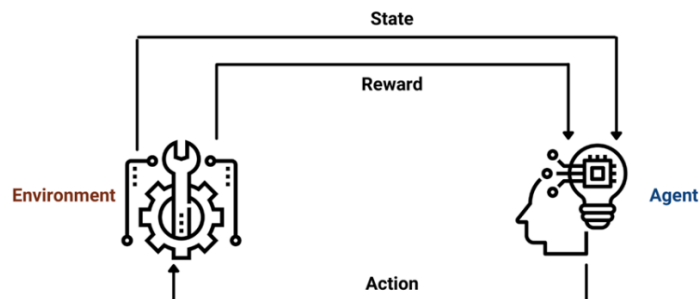


*Figure 1.13 : Schéma d'apprentissage non supervisé "unsupervised learning"*

### 6.3. L'apprentissage par renforcement

L'apprentissage par renforcement est la chose optimale qui permet d'entreprendre des actions dans un environnement afin de maximiser l'idée de récompense cumulative. Il s'agit d'un domaine de l'apprentissage automatique qui étudie de nombreuses spécialités telles que la théorie des jeux, la théorie du contrôle, les processus de recherche et la théorie de l'information., l'optimisation basée sur la simulation, les systèmes multiagents, l'intelligence en essaim, les statistiques et les algorithmes génétiques. Dans l'apprentissage automatique, l'environnement est généralement représenté comme un processus décisionnel de Markov (MDP). De nombreux algorithmes d'apprentissage par renforcement utilisent des techniques de programmation dynamique [21]. Les algorithmes d'apprentissage par renforcement ne supposent pas la connaissance d'un modèle mathématique exact du MDP et sont utilisés lorsque les modèles exacts sont infaisables.

Les algorithmes d'apprentissage par renforcement sont utilisés dans les véhicules à moteur ou pour apprendre à jouer à un jeu contre un adversaire humain.



*Figure 1.14 : Schéma d'apprentissage par renforcement "Reinforcement learning"*

## 7. L'apprentissage en profondeur

L'un des principes de l'apprentissage profond est d'extraire les caractéristiques progressivement, à mesure que l'on s'enfonce dans les couches. Par exemple, dans le traitement des images, les couches inférieures peuvent définir des contours, tandis que les concepts liés à l'être humain, tel que les chiffres, les lettres ou les visages [23] [24] [25]. Nécessitent des couches plus élevées. Il s'agit d'une classe d'algorithmes d'apprentissage automatique dont le principe de fonctionnement repose sur de multiples couches internes [22].

Les architectures d'apprentissage en profondeur ont été appliquées à des domaines tels que la vision par ordinateur, la reconnaissance vocale, le traitement du langage naturel, la traduction automatique, la bio-informatique, la conception de médicaments, la médecine programmée d'analyse d'images, d'inspection des matériaux et de jeux de société, où ils ont produit des résultats comparables et dans certains cas, surpassant les performances des experts humains [26] [27] [28] [29].

### 7.1. Les neurones biologiques

Le neurone où ce qu'on appelle la cellule nerveuse que nous nommerons par la suite l'unité fonctionnelle de base biologique pour le distinguer des neurones artificiels en sciences informatiques. Cellule spécialisée excitable constituant les unités fonctionnelles de base du système nerveux central et périphérique dans notre cerveau il y a 86 milliards parmi les regrouper en réseau. Les neurones donnent naissance à la pensée humaine aussi l'esprit la conscience imagination opinion et à ce que nous nommons intelligence [30].

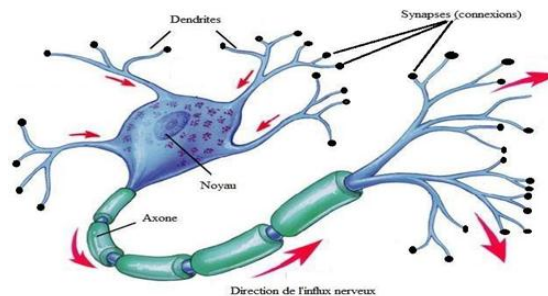
Ils ne sont pas les seules cellules des cerveaux d'autres types cellulaires jouent un rôle fondamental et participent au processus normal. Sans eux les neurones n'assurent pas la transmission de l'influx nerveux.

Les inventaires des neurones artificiels ont inspiré par les neurones biologiques ou plus exactement la propriété électrique en effet ce dernier a les particularités de crier sous certaines conditions un signal électrique que ce fait la diffusion des messages nerveux.

Le neurone biologique contient trois parties principales :

- ✓ Le corps cellulaire a un centre de contrôle traitant les informations reçues par les dendrites.

- ✓ Les dendrites ou les prolongements du corps cellulaire sont les principaux fils conducteurs par lesquels transitent l'information venue de l'extérieur.
- ✓ L'axone, ou fibre nerveuse, est le prolongement du neurone qui conduit le signal électrique du corps cellulaire vers d'autres neurones.



*Figure 1.15 : Schéma des neurones biologiques*

## 7.2. Les réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels (ANN) ou systèmes connexionnistes sont des systèmes informatiques inspirés des réseaux de neurones biologiques qui constituent le cerveau des animaux. De tels systèmes apprennent (améliorent progressivement leur capacité) à effectuer des tâches en considérant des exemples, généralement sans programmation spécifique à la tâche. Par exemple, en reconnaissance d'images, ils peuvent apprendre à identifier des images contenant des chats en analysant des exemples d'images qui ont été manuellement étiquetées comme « chat » ou « non chat » et en utilisant les résultats de l'analyse pour identifier les chats dans d'autres images. Ils ont trouvé la plupart d'utilisation dans des applications difficiles à exprimer avec un algorithme informatique traditionnel utilisant une programmation basée sur des règles [19].

Une ANN est basée sur une collection d'unités connectées appelées neurones artificiels (analogues aux neurones biologiques dans un cerveau biologique). Chaque connexion (synapse) entre neurones peut transmettre un signal à un autre neurone. Le neurone récepteur (postsynaptique) peut traiter les signaux, puis signaler les neurones en aval qui lui sont connectés. Les neurones peuvent avoir un état, généralement représenté par des nombres réels, généralement compris entre 0 et 1. Les neurones et les synapses peuvent également avoir un poids qui varie au fur et à mesure de l'apprentissage, ce qui peut augmenter ou diminuer la force du signal qu'ils envoient en aval [19].

### 7.3. Les types des réseaux de neurones artificiels

Les types de réseau de neurones diffèrent par plusieurs paramètres :

- ✓ La topologie des connexions entre les neurones.
- ✓ La fonction d'agrégation utilisée (somme pondérée, distance pseudo-euclidienne...).
- ✓ La fonction de seuillage utilisée (sigmoïde, échelon, fonction linéaire, fonction de Gauss).
- ✓ L'algorithme d'apprentissage (rétro propagation du gradient, cascade corrélation).

D'autres paramètres, spécifiques à certains types de réseaux de neurones, tels que la méthode de relaxation pour les réseaux de neurones (réseaux de Hopfield par exemple) qui ne sont pas à propagation simple (perceptron multicouche par exemple).



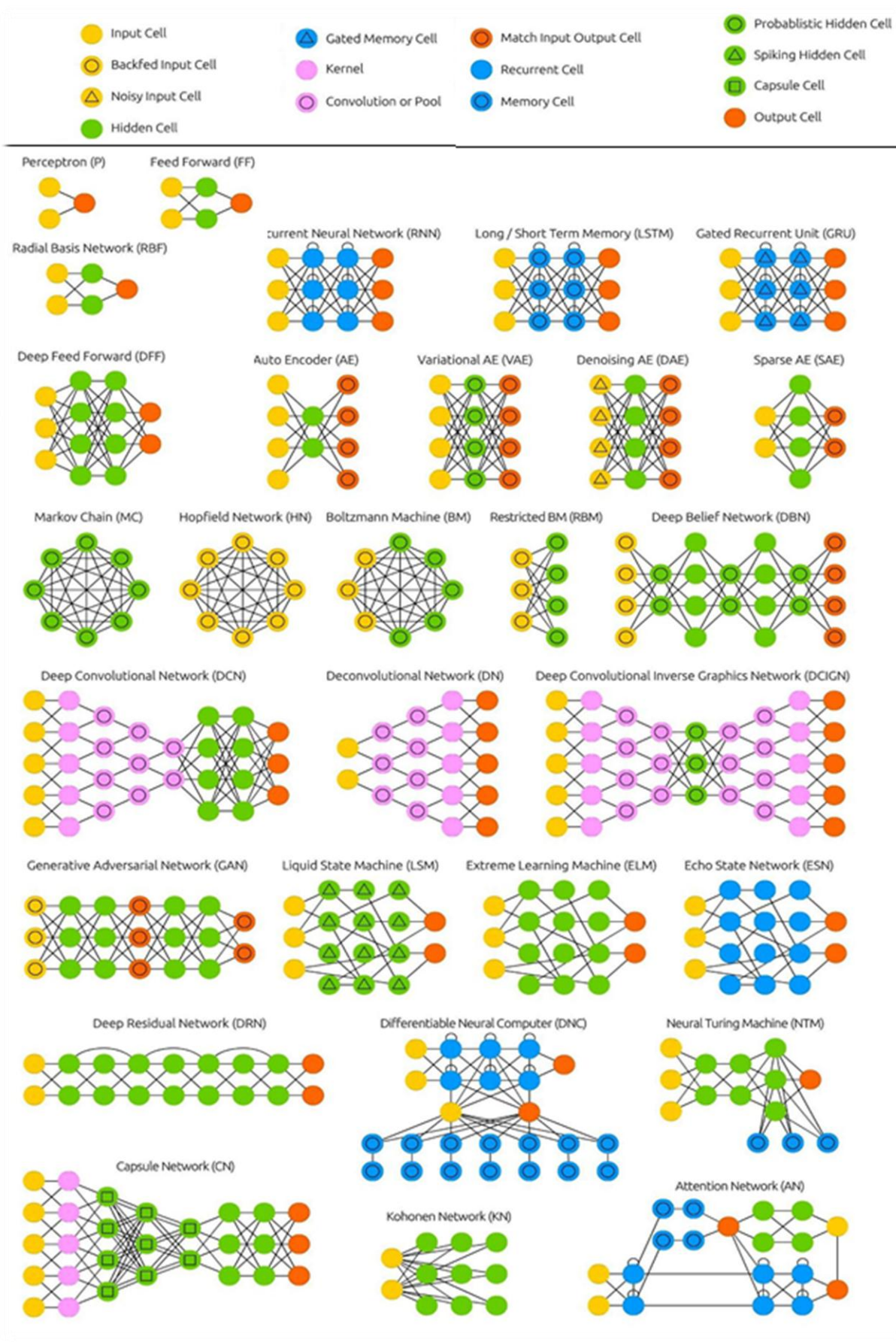


Figure 1.16 : Schéma des réseaux de neurones artificiels [36]

Il existe plusieurs types et architectures de réseaux de neurones (figure 1.16), chacun ayant ses propres forces et faiblesses, cependant, il y en a trois les plus utilisés, à savoir :

#### **7.4. Réseau de neurones convolutifs (CNN)**

CNN ou Convolutional Neural Network est un type de réseau neuronal qui est largement utilisé dans Computer Vision en raison de sa capacité à supprimer le besoin de couches entièrement connectées pendant l'exécution. Concernant la taille des images utilisées, les neurones de la première couche représentent tous les pixels de l'image. Par conséquent, il est nécessaire de trouver un réseau de neurones capable d'extraire les caractéristiques de l'image et de la convertir en dimensions inférieures sans perdre ses caractéristiques. [32]

##### **7.4.1. Évolution architecturale des CNN profondes**

De nos jours, les CNN sont considérées comme les algorithmes les plus utilisés parmi les techniques d'intelligence artificielle (IA) d'inspiration biologique. L'histoire des CNN commence avec les expériences neurobiologiques menées par Hubel et Wiesel [33]. Leurs travaux ont fourni une plate-forme pour de nombreux modèles cognitifs et le CNN a remplacé presque tous ces modèles. Au fil des décennies, différents efforts ont été menés pour améliorer les performances des CNN.

L'histoire de l'évolution des architectures CNN profondes est représentée de manière imagée sur la **Figure 1.17** Les améliorations apportées aux architectures CNN peuvent être classées en cinq époques différentes qui sont présentées ci-dessous :

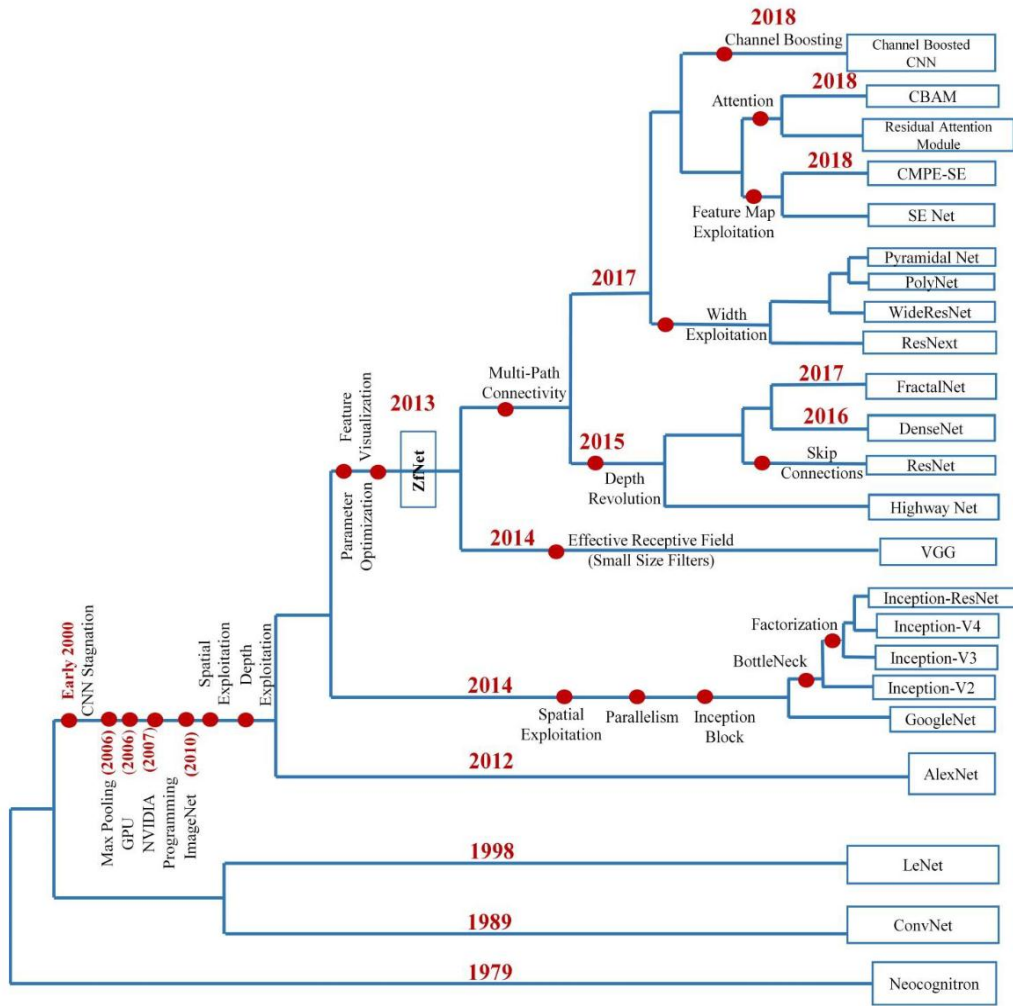


Figure 1.17 : Histoire de l'évolution des CNN profonds montrant les innovations architecturales depuis ConvNet jusqu'aux architectures actuelles [37].

### 7.4.1.1. Stagnation de CNN : Début 2000

Les chercheurs n'avaient pas beaucoup de connaissances sur les réseaux de neurones convolutifs entre 1990 et 2000, ce qui était quelque chose de difficile à comprendre en raison de la conception non simplifiée et du traitement lourd et difficile à l'époque [35].

### 7.4.1.2. Le renouveau de CNN : 2006-2011

Les réseaux CNN profonds ont généralement une architecture complexe et une phase d'apprentissage longue qui peut parfois s'étendre sur plusieurs semaines. Au début de 2000, il existait peu de techniques de traitement parallèle et des ressources matérielles limitées pour l'apprentissage des réseaux profonds. En 2006, Hinton a

présenté le concept de préformation gourmande par couche, qui a relancé la recherche sur l'apprentissage profond [38].

#### **7.4.1.3. L'ascension de CNN : 2012-2014**

À cette époque, plusieurs tentatives ont été faites pour améliorer les performances du CNN, des stratégies d'optimisation de la profondeur et des paramètres ont été explorées avec une réduction significative du coût de calcul. De mêmes, différentes conceptions architecturales ont été proposées, chaque nouvelle architecture essayant de surmonter les lacunes des architectures proposées précédemment en combinaison avec de nouvelles reformulations structurelles.

#### **7.4.1.4. Augmentation rapide des innovations architecturales et des applications du CNN : 2015- Aujourd'hui**

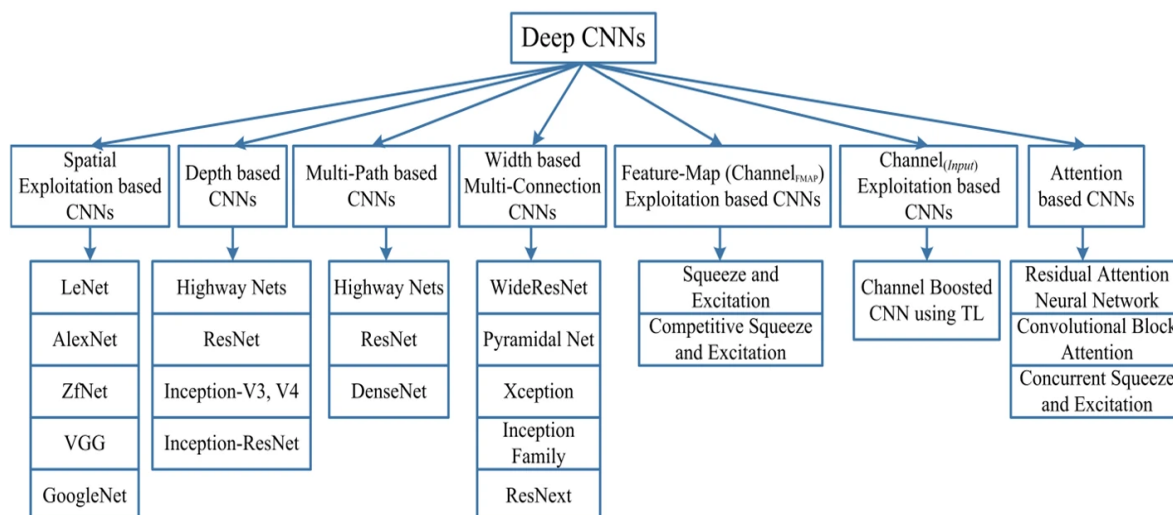
La recherche sur les CNN est toujours en cours et présente un potentiel d'amélioration important. On observe généralement que les améliorations significatives des performances des CNN ont eu lieu entre 2015 et 2019. La capacité de représentation d'un CNN dépend généralement de sa profondeur et, en un sens, un ensemble de caractéristiques enrichi allant d'abstractions simples à complexes peut aider à l'apprentissage de problèmes complexes. Cependant, le principal défi auquel sont confrontées les architectures profondes est celui du gradient décroissant.

À cet égard, différentes idées telles que le mécanisme de déclenchement de l'information à travers plusieurs couches, les connexions de saut et la connectivité des canaux entre les couches ont été introduites [45] [46] [47].

### **7.4.2. Innovations architecturales dans le CNN**

Différentes améliorations de l'architecture CNN ont été apportées de 1989 à aujourd'hui. Ces améliorations peuvent être classées comme l'optimisation des paramètres, la régularisation, la reformulation structurelle, etc. Cependant, on observe que l'amélioration des performances des CNN est principalement due à la restructuration des unités de traitement et à la conception de nouveaux blocs.

Les CNN peuvent être classés en sept classes différentes. La taxonomie des architectures CNN est représentée de manière imagée à la Figure 1.18.



**Figure 1.18** : Taxonomie des architectures CNN profondes montrant sept catégories différentes[37]

#### 7.4.2.1. CNNs basés sur l'exploitation spatiale

Les CNNs possèdent un grand nombre de paramètres et d'hyper-paramètres, tels que les poids, les biais, le nombre de couches et d'unités de traitement (neurones), la taille du filtre, le pas, la fonction d'activation, le taux d'apprentissage, etc. [48] [49]. Comme l'opération convolutif tient compte du voisinage (localité) des pixels d'entrée, différents niveaux de corrélation peuvent être explorés en utilisant différentes tailles de filtres. Différentes tailles de filtres encapsulent différents niveaux de granularité, généralement les filtres de petite taille extraient des informations à grain fin et ceux de grande taille des informations à gros grains.

Par conséquent, au début des années 2000, les chercheurs ont exploité les filtres spatiaux pour améliorer les performances et ont exploré la relation d'un filtre spatial avec l'apprentissage du réseau. Différentes études menées à cette époque ont suggéré que, grâce à l'ajustement des filtres, le CNN peut obtenir de bonnes performances tant sur les détails à gros grain que sur ceux à grain fin [18].

#### 7.4.2.2. CNNs basés sur la profondeur

Les architectures CNN profondes sont basées sur l'hypothèse qu'avec l'augmentation de la profondeur, le réseau peut mieux approcher la fonction cible avec un certain nombre de mappings non linéaires et des hiérarchies de caractéristiques plus enrichies [50]. La profondeur du réseau a joué un rôle essentiel dans le succès de la l'entraînement supervisée. Des études théoriques ont montré que

les réseaux profonds peuvent représenter certaines classes de fonctions plus efficacement que les architectures peu profondes [51].

Csáji a représenté un théorème d'approximation universel en 2001, qui stipule qu'une seule couche cachée est suffisante pour approximer n'importe quelle fonction. Cependant, cela se fait au prix d'un nombre exponentiel de neurones ; ainsi, cela rend souvent l'approximation non réaliste sur le plan informatique [52]. À cet égard, Bengio et Delalleau [53] ont suggéré que les réseaux plus profonds peuvent maintenir le pouvoir expressif du réseau à un coût réduit [54] [55].

Ont montré empiriquement que les réseaux profonds sont plus efficaces sur le plan informatique pour les tâches complexes [55] [56]. Inception et VGG, qui ont affiché les meilleures performances lors du concours 2014-ILSVRC, renforcent l'idée que la profondeur est une dimension essentielle dans la régulation de la capacité d'apprentissage des réseaux [54] [57] [58] [59].

#### **7.4.2.3. CNNs basés sur les trajets multiples**

L'entraînement de réseaux profonds est une tâche difficile, ce qui a fait l'objet de recherches récentes sur les réseaux profonds. Les CNN profondes sont généralement performantes pour les tâches complexes. Cependant, ils peuvent souffrir d'une dégradation des performances, d'une disparition du gradient ou de problèmes d'explosion, qui ne sont pas causés par un sur-ajustement mais plutôt par une augmentation de la profondeur [61] [62].

Le problème du gradient évanescent entraîne non seulement une erreur de test plus élevée, mais aussi une erreur d'entraînement plus élevée [61] [62] [63]. Pour l'apprentissage des réseaux profonds, le concept de connectivité à chemins multiples ou à couches croisées a été proposé. [45] [47] [64] [67].

Les chemins multiples ou les connexions de raccourci peuvent connecter systématiquement une couche à une autre en sautant certaines couches intermédiaires pour permettre le flux spécialisé d'informations entre les couches [66] [67]. La connectivité entre couches partitionne le réseau en plusieurs blocs. Ces chemins tentent également de résoudre le problème du gradient évanescent en rendant le gradient accessible aux couches inférieures. À cette fin, différents types de connexions de raccourci sont utilisés, tels que les connexions à remplissage nul,

basées sur la projection, le décrochage, les connexions de saut et les connexions 1x1, etc.

#### **7.4.2.4. CNNs multi-connexions basées sur la largeur**

Au cours de la période 2012-2015, l'accent a été principalement mis sur l'exploitation de la puissance de la profondeur, ainsi que sur l'efficacité des connexions régulatrices multi-passages dans la régularisation des réseaux [45] [46].

Toutefois, Kawaguchi. Ont signalé que la largeur du réseau est également importante [68]. Le perceptron multicouche a gagné l'avantage de cartographier des fonctions complexes par rapport au perceptron en faisant une utilisation parallèle de plusieurs unités de traitement dans une couche. Cela suggère que la largeur est un paramètre essentiel pour définir les principes d'apprentissage avec la profondeur [69]. Ainsi que Hanin et Sellke ont récemment montré que les NN avec la fonction d'activation ReLU doivent être suffisamment larges pour conserver la propriété d'approximation universelle avec une augmentation de la profondeur [70].

De plus, une classe de fonctions continues sur un ensemble compact ne peut pas être arbitrairement bien approximée par un réseau arbitrairement profond, si la largeur maximale du réseau n'est pas plus grande que la dimension d'entrée [71] [56]. Bien que l'empilement de plusieurs couches (augmentation de la profondeur) puisse apprendre diverses représentations de caractéristiques, il n'augmente pas nécessairement le pouvoir d'apprentissage du NN.

Un problème majeur lié aux architectures profondes est que certaines couches ou unités de traitement peuvent ne pas apprendre de caractéristiques utiles. Pour résoudre ce problème, la recherche s'est déplacée des architectures profondes et étroites vers les architectures fines et larges.

#### **7.4.2.5. CNNs basés sur l'exploitation des cartes de caractéristiques (Channel FMap)**

Les CNN sont devenues populaires pour les tâches de MV en raison de leur apprentissage hiérarchique et de leur capacité d'extraction automatique de caractéristiques [72]. La sélection des caractéristiques joue un rôle essentiel dans la détermination de la performance des modules de classification, de segmentation et

de détection. Dans le CNN, les caractéristiques sont sélectionnées dynamiquement en ajustant les poids associés à un noyau, également appelé masque. De plus, plusieurs étapes d'extraction de caractéristiques sont utilisées, ce qui permet d'extraire divers types de caractéristiques (appelées cartes de caractéristiques ou canaux dans CNN). Cependant, certaines des cartes de caractéristiques ne jouent qu'un rôle limité ou nul dans la discrimination des objets [73]. Des ensembles de caractéristiques énormes peuvent créer un effet de bruit et conduire ainsi à un surajustement du réseau.

#### **7.4.2.6. CNNs basés sur l'exploitation des canaux (entrées)**

La représentation de l'image joue un rôle important dans la détermination de la performance des algorithmes de traitement d'image, y compris les algorithmes conventionnels et d'apprentissage profond. Une bonne représentation de l'image est celle qui peut définir les caractéristiques saillantes d'une image à partir d'un code compact.

Dans les tâches de MV, divers types de filtres conventionnels sont appliqués pour extraire différents niveaux d'information pour un seul type d'image [74] [75]. Ces différentes représentations sont ensuite utilisées comme entrée du modèle pour améliorer les performances [76] [77]. Aujourd'hui, le CNN est un apprenant de caractéristiques convaincant qui peut extraire automatiquement des caractéristiques discriminantes en fonction du problème [78]. Cependant l'apprentissage du CNN repose sur la représentation de l'entrée.

#### **7.4.2.7. CNNs basés sur l'attention**

Les différents niveaux d'abstractions jouent un rôle important dans la définition du pouvoir de discrimination des CNN. En plus de l'apprentissage de multiples hiérarchies d'abstractions, la concentration sur les caractéristiques pertinentes pour le contexte joue également un rôle important dans la localisation et la reconnaissance d'images. Dans le système visuel humain, ce phénomène est appelé attention. L'être humain observe la scène en une succession d'aperçus partiels et prête attention aux parties pertinentes pour le contexte. Ce processus sert non seulement à focaliser les régions sélectionnées, mais aussi à déduire différentes interprétations des objets à cet



endroit, ce qui permet de mieux saisir la structure visuelle. Un type d'interprétabilité plus ou moins similaire est ajouté aux RNN et LSTM [79] [80].

Les réseaux RNN et LSTM exploitent les modules d'attention pour la génération de données séquentielles, et les nouveaux échantillons sont pondérés en fonction de leur occurrence dans les itérations précédentes. Le concept d'attention a été incorporé dans les réseaux CNN par divers chercheurs afin d'améliorer la représentation et de surmonter les limites de calcul. Cette idée d'attention aide également à rendre le CNN suffisamment intelligent pour reconnaître des objets même dans des arrière-plans encombrés et des scènes complexes.

### 7.4.3. Comparaison entre tous les architecture CNNs dans tous les catégories

N°	Nom de l'architecture	Année	Contribution principale	Paramètres	Taux d'erreur	Profondeur	Catégorie	Réf
1	LeNet	1998	- Première architecture CNN populaire	0.060 M	MNIST: 0.95	5	Spatial Exploitation	[81]
2	AlexNet	2012	- Plus profond et plus large que LeNet - Utilise Relu, dropout et chevauchement Pooling - GPU NVIDIA GTX 580	60 M	ImageNet: 16.4	8	Spatial Exploitation	[43]
3	ZfNet	2014	- Visualisation des couches intermédiaires	60 M	ImageNet: 11.7	8	Spatial Exploitation	[82]
4	VGG	2014	- Topologie homogène - Utilise des noyaux de petite taille	138 M	ImageNet: 7.3	19	Spatial Exploitation	[83]
5	GoogLeNet	2015	- Introduction du concept de bloc - Diviser la transformation et fusionner l'idée	4 M	ImageNet: 6.7	22	Spatial Exploitation	[44]
6	Inception-V3	2015	- Gère le problème d'un goulot d'étranglement représentationnel - Remplacez les filtres de grande taille par de petits filtres	23.6 M	ImageNet : 3.5 Multi-Crop: 3.58 Single-Crop: 5.6	159	Depth + Width	[59]
7	Highway Networks	2015	- Introduction d'une idée de multi-chemins	2.3 M	CIFAR-10: 7.76	19	Depth + Multi-Path	[84]
8	Inception-V4	2016	- Diviser la transformation et l'idée de fusion Utilise des filtres asymétriques	35 M	ImageNet: 4.01	70	Depth +Width	[58]
9	Inception-ResNet	2016	- Utilise l'idée de fusion de transformation fractionnée et les liens résiduels	55.8M	ImageNet: 3.52	572	Depth + Width + Multi-Path	[58]
10	ResNet	2016	- Residual learning - Identity mapping based skip connections	25.6 M 1.7 M	ImageNet : 3.6 CIFAR-10 : 6.43	152 110	Depth + Multi-Path	[46]
11	DelugeNet	2016	- Permet le flux d'informations	20.2 M	CIFAR-10 : 3.76	146	Multi-path	[64]

			inter-couches dans les réseaux profonds		CIFAR-100 : 19.02			
12	FractalNet	2016	- Différentes longueurs de chemin interagissent les unes avec les autres sans aucune connexion résiduelle	38.6 M	CIFAR-10 : 7.27 CIFAR-10+ : 4.60 CIFAR-10++ : 4.59 CIFAR-100 : 28.20 CIFAR-100+ : 22.49 CIFAR100++ : 21.49	20 40	Multi-Path	[65]
13	WideResNet	2016	- La largeur est augmentée et la profondeur est diminuée	36.5 M	CIFAR-10 : 3.89 CIFAR-100 : 18.85	28 -	Width	[85]
14	Xception	2017	- Convolution en profondeur suivie d'une convolution par points convolution par points	22.8 M	ImageNet: 0.055	126	Width	[86]
15	Residual Attention Neural Network	2017	- Introduction d'un mécanisme d'attention	8.6 M	CIFAR-10 : 3.90 CIFAR-100 : 20.4 ImageNet : 4.8	452	Attention	[87]
16	ResNeXt	2017	- Cardinalité - Topologie homogène - Convolution groupée	68.1 M	CIFAR-10 : 3.58 CIFAR-100 : 17.31 ImageNet : 4.4	29 - 101	Width	[88]
17	Squeeze & Excitation Networks	2017	- Modélisation des interdépendances entre cartes de caractéristiques	27.5 M	ImageNet: 2.3	152	Feature-Map Exploitation	[73]
18	DenseNet	2017	- Flux d'informations entre couches	25.6 M 25.6 M 15.3 M 15.3 M	CIFAR-10+ : 3.46 CIFAR100+ :17.18 CIFAR-10 : 5.19 CIFAR-100 : 19.64	190 190 250 250	Multi-Path	[47]
19	PolyNet	2017	- Expérimenter la diversité structurelle - Introduit le module Poly Inception - Généraliser l'unité résiduelle en utilisant Compositions polynomiales	92 M	ImageNet : Single :4.25 Multi :3.45	-	Width	[89]
20	PyramidalNet	2017	- Augmente progressivement la largeur par unité	116.4 M 27.0 M 27.0 M	ImageNet : 4.7 CIFAR-10 : 3.48 CIFAR-100 : 17.01	200 164 164	Width	[90]
21	Convolutional Block Attention Module (ResNeXt101 (32x4d) + CBAM)	2018	- Exploite à la fois les informations spatiales et les cartes de caractéristiques	48.96 M	ImageNet: 5.59	101	Attention	[91]
22	Concurrent Spatial & Channel Excitation Mechanism	2018	- L'attention spatiale - L'attention sur la carte des caractéristiques - Placement simultané de l'attention spatiale et canal d'attention	-	MALC : 0.12 Visceral : 0.09	-	Attention	[92]

23	Channel Boosted CNN	2018	- Renforcement des canaux originaux avec Des canaux artificiels générés riches en informations supplémentaires	-	-	-	Channel Boosting	[93]
24	Competitive Squeeze & Excitation Network CMPE-SE-WRN-28	2018	- Les mappages résiduels et d'identité sont tous deux utilisées pour redimensionner la carte des caractéristiques	36.92 M 36.90 M	CIFAR-10 : 3.58 CIFAR-100 : 18.47	152 152	Feature-Map Exploitation	[94]

*Tableau 1.1: Comparaison des performances des architectures récentes de différentes catégories [37].*

### 7.5. Réseau de neurones récurrents (RNN)

Un réseau de neurones récurrents (RNN) est une classe de réseaux de neurones artificiels où les connexions entre les nœuds forment un graphe orienté le long d'une séquence temporelle. Cela lui permet de présenter un comportement dynamique temporel. Dérivés des réseaux de neurones à action directe, les RNN peuvent utiliser leur état interne (mémoire) pour traiter des séquences d'entrées de longueur variable [95] [96] [97]. Cela les rend applicables à des tâches telles que la reconnaissance d'écriture manuscrite non segmentée et connectée ou la reconnaissance vocale [98] [99].

### 7.6. Réseau Long short-term memory (LSTM)

Le réseau Long short-term memory (LSTM) est une architecture de réseau neuronal récurrent (RNN) artificielle, utilisée dans le domaine de l'apprentissage profond. Contrairement aux réseaux neuronaux à anticipation standard, le LSTM possède des connexions de rétroaction. Il peut non seulement traiter des points de données uniques (comme des images), mais aussi des séquences entières de données (comme la parole ou la vidéo) [100].

Les réseaux LSTM sont bien adaptés à la classification, au traitement et aux prédictions basés sur des données de séries temporelles, car il peut y avoir des décalages de durée inconnue entre des événements importants dans une série temporelle. Les LSTM ont été développés pour résoudre le problème du gradient de fuite qui peut être rencontré lors de l'entraînement des RNN traditionnels. L'insensibilité relative à la longueur du décalage est un avantage des LSTM par rapport aux RNN, aux modèles de Markov cachés et à d'autres méthodes d'apprentissage de séquences dans de nombreuses applications [100].

### 7.7. Domaines d'application des réseaux de neurones artificiels

CNN	RNN	LSTM
<ul style="list-style-type: none"> <li>▪ Reconnaissance d'images</li> <li>▪ Reconnaissance par électromyographie (EMG)</li> <li>▪ Analyse vidéo</li> <li>▪ Traitement du langage naturel</li> <li>▪ Détection d'anomalies</li> <li>▪ Découverte de médicaments</li> <li>▪ Évaluation des risques pour la santé et découverte des biomarqueurs du vieillissement</li> <li>▪ Jeu de dames</li> <li>▪ Allez le jeu</li> <li>▪ Prévion de séries chronologiques</li> </ul>	<ul style="list-style-type: none"> <li>▪ Contrôle des robots</li> <li>▪ Prévion de séries temporelles</li> <li>▪ Reconnaissance vocale</li> <li>▪ Synthèse vocale</li> <li>▪ Détection d'anomalies de séries temporelles</li> <li>▪ Apprentissage du rythme</li> <li>▪ Composition musicale</li> <li>▪ Apprentissage de la grammaire</li> <li>▪ Reconnaissance de l'écriture manuscrite</li> <li>▪ Reconnaissance de l'action humaine</li> <li>▪ Détection d'homologie de protéines</li> <li>▪ Prédiction de la localisation subcellulaire des protéines</li> <li>▪ Plusieurs tâches de prédiction dans le domaine de la gestion des processus d'affaires.</li> <li>▪ Prédiction dans les parcours de soins</li> </ul>	<ul style="list-style-type: none"> <li>▪ Contrôle des robots</li> <li>▪ Prévion de séries temporelles</li> <li>▪ Reconnaissance vocale</li> <li>▪ Apprentissage du rythme</li> <li>▪ Composition musicale</li> <li>▪ Apprentissage de la grammaire</li> <li>▪ Reconnaissance de l'écriture manuscrite</li> <li>▪ Reconnaissance de l'action humaine</li> <li>▪ Traduction en langue des signes</li> <li>▪ Détection d'homologie de protéines</li> <li>▪ Prédiction de la localisation subcellulaire des protéines</li> <li>▪ Détection d'anomalies de séries temporelles</li> <li>▪ Plusieurs tâches de prédiction dans le domaine de               <ul style="list-style-type: none"> <li>▪ gestion des processus d'affaires</li> <li>▪ Prédiction dans les parcours de soins</li> <li>▪ Analyse sémantique</li> <li>▪ Co-segmentation des objets</li> <li>▪ Gestion des passagers de l'aéroport</li> <li>▪ Prévion de trafic à court terme</li> </ul> </li> </ul>

*Tableau 1.2: les domaines d'application des réseaux de neurones artificiels*

## 8. Les travaux connexes pour la classification de covid-19

Le besoin d'identification automatique des maladies thoraciques et pulmonaires par radiographie x-ray est devenu urgent et peut être réalisé par l'intelligence artificielle, surtout ces derniers temps avec l'émergence de la maladie de Covid-19. De nombreux scientifiques ont utilisé des images radiologiques pour diagnostiquer cette maladie [229].

Dans l'article de la référence numéro [230], Naren et al ont utilisé l'apprentissage par transfert pour une classification binaire à quatre classes, cette enquête a examiné cinq modèles pré-entraînés sur des réseaux de neurones convolutifs (InceptionV3, ResNet50, ResNet152, Inception-ResNetV2 et ResNet101) pour distinguer les patients atteints de COVID et de pneumonie à l'aide d'images de radiographies thoraciques Parmi ces résultats, le modèle pré-entraîné Resnet-50 fournit les meilleurs résultats dans cette catégorie.

Dans le recherche de la référence numéro [231], l'auteur a mené des expériences sur différents modèles d'apprentissage profond DenseNet121, DenseNet169 et DenseNet201. L'auteur a utilisé un total de 746 images pour entraîner les modèles et les tester par le biais d'expériences. L'auteur montre que DenseNet201 a d'excellentes performances par rapport aux autres modèles d'apprentissage profond.

Maghdid et al construisent un réseau simple et apportent des modifications au modèle AlexNet qui a déjà été entraîné à l'aide d'un ensemble de données comprenant des radiographies et CTs. Les auteurs ont utilisé des techniques d'apprentissage par transfert. Cette étude indique que l'utilisation de l'apprentissage par transfert peut fournir au modèle utilisé une précision allant jusqu'à 98 % et 94,1 % en utilisant le CNN modifié [232].

Apostolopoulos et al. Ont utilisé deux ensembles de données sur l'apprentissage par transfert, l'un contenant 224 images de Covid-19 et l'autre 224 images de la maladie de Covid-19. Les meilleurs scores de spécificité, d'exactitude et de sensibilité obtenus étaient respectivement de 96,46 %, 98,66 % et 96,78 % [233].

Dans l'article de la référence numéro [234], Majeed et al. Ont proposé un CNN simple à 4 couches, comparée à 12 CNN pré-entraînés pour la classification et la détection de Covid-19 (Vgg19, ResNet101, GoogleNet, InceptionResNetV2, ResNet50, SqueezeNet, Vgg16, Densenet201, Xception, ResNet18, Alexnet, InceptionV3). Les auteurs ont justifié les résultats à l'aide du CAM (Class Activation Mapping). Cet article était basé sur une classification binaire et une classification multiclasse comportant 4 catégories : normal ou sans infection, Covid-19, pneumonie virale et pneumonie bactérienne. Pour le premier type de classification (normal vs Covid-19), leur modèle a obtenu une sensibilité de 93,15 % et une spécificité de 97,86 %, pour le deuxième type de classification (normal vs Covid-19 vs viral), leur modèle a obtenu une sensibilité de 91,78 % et 99,05 % (pour les cas Covid-19 uniquement) et pour la troisième classification (Covid-19 vs viral vs pneumonie bactérienne), le modèle a donné une sensibilité de 94,52 % et une spécificité de 99,35 % (pour les cas Covid-19 uniquement).

## **9. Les défis sur Kaggle**

Le virus Covid-19 est un virus mortel pour l'humanité, car il cible directement les poumons et les détruit, allant parfois jusqu'à tuer des humains, car il est cinq fois plus dangereux que la grippe.

Les symptômes de ce virus peuvent être diagnostiqués biologiquement par PCR "réaction en chaîne par polymérase", mais ce processus prend beaucoup de temps, ce qui fait souffrir davantage le patient et son état peut se détériorer, mais il existe une autre solution, qui est de réaliser des images médicales et connaître l'état des poumons et l'étendue des dommages.

Pour faciliter ce processus pour les médecins, ils peuvent attribuer ce travail aux ordinateurs au moyen de l'intelligence artificielle et leur faire prendre en charge cette tâche. Ainsi, pour encourager cette idée, de nombreux défis sont apparus sur de nombreux sites. Y compris le site Web de Kaggle, qui offrent des prix et des cadeaux pour les meilleurs projets, et nous verrons le plus important de ces défis dans ce tableau :

N°	nom de challenge	description	prix	date de début	date de fin
1	SIIM-FISABIO-RSNA COVID-19 Detection [101]	Identifier et localiser les anomalies COVID-19 sur les radiographies pulmonaires	1st Place - \$30,000 2nd Place - \$20,000 3rd Place - \$10,000 4th Place - \$8,000 5th Place - \$7,000 6th - 10th Places - \$5,000 each	18-05-2021	10-08-2021
2	COVID-19 Hospital Stay Prediction [102]	Le 31e concours d'analyse de données 1056Lab	1st place - médaille d'or et 10 points 2nd place - médaille d'argent et 8 points 3rd place - médaille de Bronze et 6 points 4th place - 4 points 5th place - 2 points sur kaggle	31/05/2021	07/06/2021
3	Covid-19 Risk [103]	Prédire les résultats des patients dans les cas confirmés de Covid-19 à partir de facteurs de risque, en utilisant des données du Mexique		25/01/2021	15/02/2021
4	ECE657AW21-Assignment4-COVID [104]	Tâche de classification de réseau de neurones profonds		25/03/2021	26/04/2021
5	Small Covid Data [105]	Pour l'ensemble de données fourni, recherchez toutes les lignes aberrantes/anomalies		26/01/2021	24/02/2021
6	COMS 4771 COVID Challenge [106]	Développer des méthodes de diagnostic des cas de COVID-19 et de pneumonie à partir de radiographies pulmonaires		27/03/2020	10/05/2020
7	COVID19 Local US-CA Forecasting (Week 1) [107]	Prévision quotidienne de la propagation du COVID-19 en Californie, aux États-Unis		19/03/2020	26/03/2020
8	COVID-19 detection from coughs [108]	Prédire qu'un patient est infecté par COVID-19 à partir d'enregistrements de toux		08/05/2021	13/01/2021
9	ML2021Spring-hw1 [109]	Prévision des cas de COVID-19		05/03/2021	26/03/2021
10	COVID-19 CT Images Segmentation [110]	Résultats radiologiques de segment sur des coupes axiales de poumons		15/05/2020	02/01/2030
11	COVID-19 Hospital Stay Prediction [111]	Le 31e concours d'analyse de données 1056Lab		31/05/2021	07/06/2021
12	COVID-19 Segmentation [112]	Segmentation des régions de consolidation sur les scanners pulmonaires		26/02/2021	02/04/2021

13	COVID-19 Cases Prediction [113]	Machine Learning IT KMITL 2020		20/04/2021	10/05/2021
14	COVID-19 X-ray image classification [114]	Classification des cas de COVID-19 à partir d'images de radiographie pulmonaire		30/01/2021	15/02/2021
15	COVID-19 Chest x-ray challenge [115]	CSC532 Classe d'apprentissage machine Hackathon 2021		05/03/2021	22/03/2021
16	ProtonX - Covid 19 Classification [116]	Classement pour l'attribution de la classification		05/09/2021	28/10/2021
17	COVID19 Retweet Prediction Challenge 2020 [117]	Donnez une collection de tweets liés à covid19, prédisez le nombre de retweets		09/11/2020	11/12/2020
18	CT Scan COVID Prediction [118]	COVID Prediction from CT Scan Images	Des prix seront décernés aux meilleures candidatures par le professeur Mehul Motani, professeur associé, membre de l'IEEE, Université nationale de Singapour, Département de génie électrique et informatique.	13/05/2020	19/05/2022
19	Multi-class COVID-19 Chest x-ray challenge [119]	CSC532 Cours d'apprentissage machine Hackathon 2 (2021)		29/04/2021	08/05/2021
20	Covid-19 detection using transfer learning [120]	Sur la base d'une radiographie pulmonaire, prédire si le patient est infecté par le covid-19		03/12/2020	06/12/2020
21	Hack-D-Covid'20 Track-1 [121]	Challenge to detect Covid-19 from scanned X-Ray images		23/10/2020	24/10/2020
22	Prediction of COVID-19 diagnosis based on symptoms [122]	Un dépistage efficace du SRAS-CoV-2 permet un diagnostic rapide et efficace du COVID-19		18/06/2021	03/07/2021
23	Diagnóstico de COVID19 según síntomas del paciente [123]	Un dépistage efficace du SRAS-CoV-2 permet un diagnostic rapide et efficace du COVID-19		23/01/2021	31/01/2021
24	AIAT Hackathon: COVID Image [124]	Dans ce concours, nous aimerions que vous aidiez au dépistage des patients COVID à partir d'images radiographiques pulmonaires.		16/06/2020	15/06/2020
25	2020_UCLA_CS145_Covid19_Prediction [125]	Predict daily #cases and # deaths for each state in the U.S.		07/10/2020	11/10/2020

*Tableau 1.3: La plupart des défis de covid-19 on Kaggle*

## 10. Les Base de données existant

Afin de faciliter la recherche de solutions. Il a été créé des Datasets qui contiennent de nombreuses images médicales des poumons pour les personnes normales, les patients atteints de covid-19 et pour les patients atteints d'autres maladies. Parmi ces Datasets figurent les suivants :

N°	nom de Dataset	description	Nombre total des images	nombre d'image normal	nombre d'image covid	nombre d'image pneumonie
1	COVID-19 Radiography Database [126]**	Base de données de radiographie pulmonaire COVID-19	21200	10192	1616	1345
2	COVID-19 chest xray [127]**	Collecte de données d'images COVID-19	357	0	357	0
3	Covid-19 Image Dataset [128]**	Classification à 3 voies - COVID-19, pneumonie virale, normale	317	90	137	90
4	COVIDx CXR-2 [129]	Radiographies thoraciques pour la détection du COVID-19	16400	0	16400	0
5	COVID-19 Xray Dataset (Train & Test Sets) [130]	avec le détecteur de pneumonie COVID-19 CNN	188	94	0	94
6	COVID-19 Patients Lungs X Ray Images 10000 [131]	Ensemble de données sur le virus Corona	98	28	70	0
7	Chest X-ray (Covid-19 & Pneumonia)[132]	L'ensemble de données contient des images radiographiques thoraciques de Covid-19, de pneumonie et de patients normaux.	6432	1583	576	4273
8	COVID-19 Chest X-ray Image Dataset [133]	images de radiographie pulmonaire normales et affectées par covid-19	94	25	69	0
9	Chest Xray for covid-19 detection [134]**	Ensemble de données de radiographie pulmonaire pour la détection de covid-19	371	176	174	0
10	CoronaHack -Chest X-Ray-Dataset [135]	Classer l'image X Ray qui a Corona	5935	0	5935	0
11	COVID19_Pneumonia_Normal_Chest_Xray_PA_Dataset [136]	Ensemble de données COVID19 avec pneumonie et radiographie pulmonaire normale (PA)	6939	2313	2313	2313
12	COVID-19 X rays [137]	Radiographies et clichés tomodensitométriques des patients CONVID-19	97	0	97	0
13	Covid Patients Chest X-Ray [138]**	Téléchargez 162 images de radiographies pulmonaires de patients covid et de patients normaux.	342	162	162	0
14	Covid-GAN and Covid-Net mini Chest X-ray [139]	Radiographie pulmonaire pour les patients atteints de covid et de pneumonie avec GAN généré	7544	2083	972	4489
15	Chest X-Ray Images (Pneumonia) [140]*	5,863 images, 2 catégories	5856	1583	0	4273
16	covid-chestxray-dataset [141]*	ensemble de données d'images radiographiques thoraciques et tomodensitométriques de patients positifs ou suspectés de COVID-19 ou d'autres pneumonies virales et bactériennes (MERS, SRAS et ARDS.)	930	0	930	0

**Tableau 1.4:** La plupart des datasets de covid-19 on Kaggle

\* : les base de donnes utiliser pour la collection de notre dataset

\*\* : les base de donnes utiliser pour des tests externes



## 11. Conclusion

À la fin de ce chapitre, nous avons pris connaissance de la plupart des éléments abordés dans ce projet, tant d'un point de vue biologique que technique. Du point de vue biologique, nous avons une idée du virus Corona et de sa structure, ainsi que du système respiratoire et de la structure de la synapse nerveuse. et du point de vue technique, nous avons vu un aperçu historique du développement des réseaux neuronaux convolutifs depuis leurs débuts et leurs divisions en termes de structure interne.

Nous avons également vu des bases de données qui aident à développer l'entraînement des réseaux neuronaux convolutifs pour la classification du virus Covid aux radiographies thoraciques et nous avons également vu la plupart des défis sur un site Kaggle dans ce domaine.

Nous essaierons d'en savoir plus sur le côté technique des réseaux neuronaux convolutifs dans le prochain chapitre.

# Chapitre 02

# Les réseaux neuronaux convolutifs

## 1. Introduction

Les réseaux neuronaux convolutifs (CNN ou ConvNet) sont un type de réseau neuronal artificiel dans lequel le modèle de communication entre les neurones est inspiré des réseaux neuronaux biologiques.

Ce réseau est constitué de nœuds ou de soi-disant neurones ou unités de traitement connectés entre eux pour former un réseau de neurones, et chaque connexion entre ces nœuds à un ensemble de valeurs appelées poids, ces poids permettent de déterminer les valeurs résultantes pour chacun élément de traitement basé sur les valeurs internes de cet élément [156].

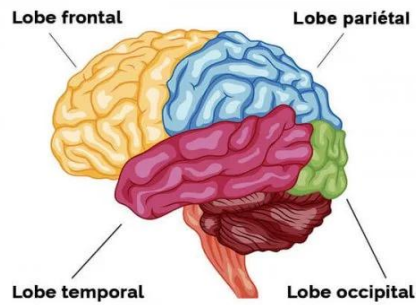
Dans ce chapitre, nous étudierons les principes et les bases des réseaux de neurones convolutifs, et nous nous concentrerons sur :

- Les méthodes de construction de CNN les plus importantes telles que la convolution et pooling.
- Les différentes couches présentées dans l'architecture CNN, de la couche d'entrée à la couche de connectivité complète et les sorties avec leurs détails spécifiques du type de données aux fonctions qui leur sont appliquées.
- Les techniques utilisées dans l'amélioration des CNN, telles que la normalisation, l'optimisation et la régularisation, et comment éviter le sur-apprentissage ou le sous-apprentissage.
- Les Framework et les fonctions utilisés dans la construction d'un CNN.

## 2. Cortex visuel du cerveau et la vision par ordinateur

Le cortex cérébral est le responsable du traitement des données visuelles et est situé dans le lobe occipital à l'arrière du cerveau. Il joue un grand rôle dans le traitement des informations visuelles provenant de l'œil [157] [158].

Le nombre moyen de neurones dans le cortex visuel primaire d'un humain adulte dans chaque hémisphère a été estimé à 140 millions [159].



*Figure 2.1 : Les lobes-cérébraux*

D'où le terme de vision par ordinateur, qui permet de simuler ce processus qui se produit chez l'humain.

Il est devenu le plus important et efficace ces dernières années en raison de ses applications étendues dans divers domaines, notamment la santé et la médecine, la surveillance, la robotique et les drones.

Les développements récents dans les réseaux de neurones convolutifs (CNN) ont abouti à des performances exceptionnelles dans ces applications visuelles modernes, et par conséquent les CNN sont désormais au cœur des algorithmes d'apprentissage en profondeur en vision par ordinateur.

Et en d'autre côté une image est essentiellement un tableau bidimensionnel de pixels sur les axes x et y. Chaque pixel possède un ensemble de qualités et de caractéristiques qui déterminent son type, son identité et sa fonction dans l'image, représentés par un seuil de couleur spécifique [160].

Par conséquent, lorsque l'œil humain traite une image spécifique, il la voit sous sa forme naturelle, tandis que l'ordinateur la voit sous la forme d'un tableau numérique de valeurs, car chaque pixel a une valeur sous une forme figurative proche de celle-ci image.



*Figure 2.2 : Valeurs des couleurs RGB de certains pixels de l'image de Lena*

### 3. L'architecture d'un réseau de neurones convolutifs

L'architecture d'un réseau de neurones convolutifs contient de nombreuses couches allant de la couche d'entrée à la couche de sortie en passant par de nombreuses étapes nécessaires et importantes qui peuvent être répétées, chacune de ces étapes à un rôle dans la réussite de la fonction de CNN.

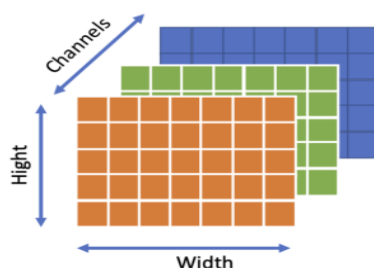
Dans cette image, nous verrons les couches et les fonctions formées pour une structure CNN simple et nous essaierons d'expliquer chacune séparément.



*Figure 2.3 : L'architecture d'un réseau de neurones convolutifs simple*

#### 3.1. La couche d'entrée

C'est l'entrée du réseau CNN qu'il s'agisse de texte, d'audio ou d'image. Dans un réseau de neurones de traitement d'images, il s'agit généralement de la matrice de pixels d'une image. Cette matrice est généralement représentée par trois dimensions qui sont la longueur, la largeur et la profondeur (RGB couleurs) [161] [162].



*Figure 2.4 : Les trois dimensions d'image*

### 3.2. La couche de convolution

L'une des choses principales et importantes dans la construction d'un réseau CNN est la couche convolutifs. Lorsque nous présentée les couches convolutifs, nous devons expliquer ce quoi la convolution. La convolution est une opération arithmétique permettant de combiner deux ensembles d'informations. La convolution est appliquée aux données d'entrée à l'aide d'un filtre de convolution pour produire un vecteur des caractéristiques.

Sur le côté gauche de la figure 2.5 se trouve l'entrée de la couche de convolution, par exemple l'image d'entrée. Et à droite se trouve le filtre de convolution, également appelé noyau ou kernel, nous utiliserons ces termes de manière interchangeable. C'est ce qu'on appelle une convolution 3x3 en raison de la forme du filtre [163].

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

**Figure 2.5 :** Matrice d'entrée avec le filtre de convolution

L'opération de convolution est effectuée plusieurs fois en déplaçant ce filtre sur toute la matrice ou l'image d'entrée, on multiplie la matrice avec le filtre et on additionne le résultat. Ce résultat va à vecteur des caractéristiques. Nous faisons ce processus pour couvrir toutes les parties de l'image.

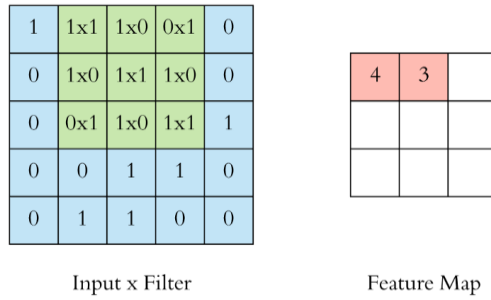
1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

Input x Filter

4		

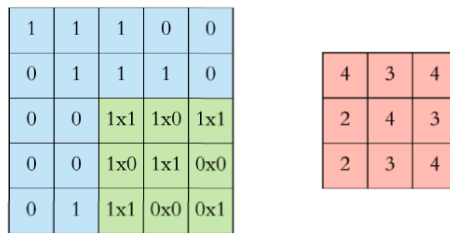
Feature Map

**Figure 2.6 :** Première étape de convolution



**Figure 2.7 :** Deuxième étape de convolution

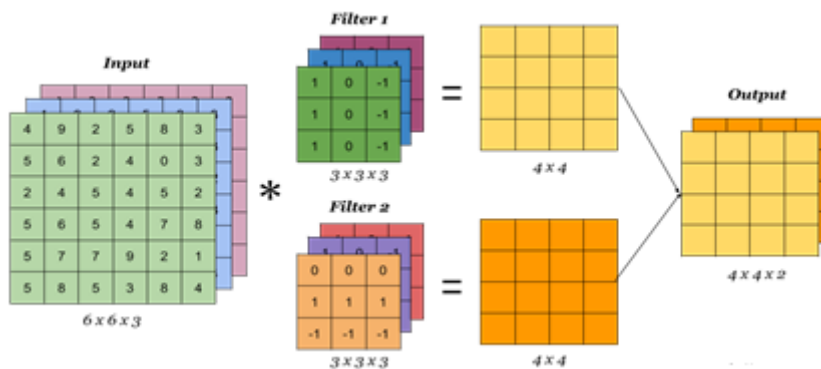
Encore et encore nous répétons le processus jusqu'à ce que nous couvrons toute l'image.



**Figure 2.8 :** Vecteur de caractéristique après l'application de convolution

### 3.2.1. Les filtres

Les filtres jouent un rôle important et efficace dans l'extraction des propriétés des images au cours du processus de convolution. Ce sont des formes carrées avec certaines dimensions. Ces formes sont appelées Kernel ou noyaux et sont appliquées dans toute l'image.



**Figure 2.9 :** Exemple d'application des filtres

### 3.2.2. Les Pas (Stride)

Le pas est le nombre de décalages de pixels sur l'ensemble de la matrice d'entrée. Lorsque le pas est de 1, nous déplaçons les filtres sur 1 pixel à la fois. Lorsque le pas est de 2, nous déplaçons les filtres sur 2 pixels à la fois et ainsi de suite [164] [165].

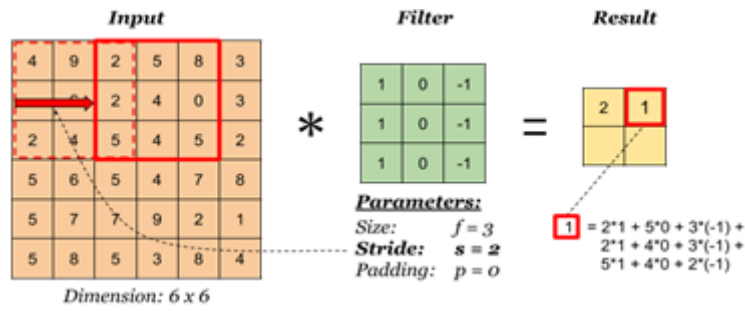


Figure 2.10 : Exemple d'application de pas "stride"

### 3.2.3. Rembourrage (padding)

Le rembourrage permet d'utiliser une couche convolution sans nécessairement rétrécir la hauteur et la largeur des volumes. Ceci est important pour la construction de réseaux plus profonds, car sinon la hauteur/largeur diminuerait au fur et à mesure que nous allons vers des couches plus profondes [166].

Cela nous aide à garder plus d'informations à la frontière d'une image. Sans rembourrage, très peu de valeurs du calque suivant seraient affectées par les pixels en tant que bords d'une image.

- Quelques terminologies de rembourrage :

- ✓ Rembourrage « valid » : pas de rembourrage.
- ✓ Rembourrage « same » : rembourrage afin que la dimension de sortie soit la même que l'entrée.
- ✓ Rembourrage « full » : introduis des zéros tels que tous les pixels sont visités par le même nombre de carreaux par le filtre, ce qui entraîne une augmentation de la taille de la sortie.

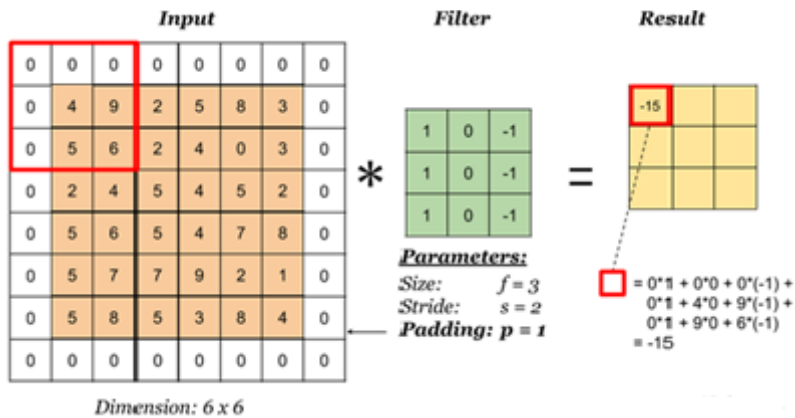


Figure 2.11 : Exemple d'application de rembourrage



### 3.3. Fonction d'activation ReLu (unité linéaire rectifiée)

ReLU est la fonction d'activation la plus utilisée actuellement par rapport au sigmoïde et au TanH, car tous les inconvénients tels que le problème de gradient de fuite sont complètement supprimés dans cette fonction d'activation, ce qui rend cette fonction d'activation plus avancée par rapport aux autres fonctions d'activation [167].

Intervalle : 0 à l'infini.

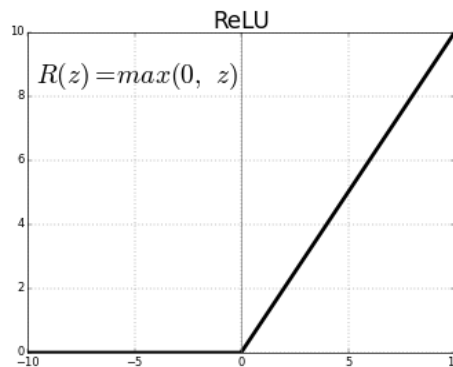


Figure 2.12 : Graphique de la Fonction d'activation ReLu

### 3.4. L'opération de pooling

L'opération de pooling consiste à faire glisser un filtre bidimensionnel sur chaque canal du vecteur de caractéristiques et à résumer les caractéristiques se trouvant dans la région couverte par le filtre [168].

Pour un vecteur de caractéristiques ayant des dimensions  $nh * nw * nc$   
Les dimensions de sortie obtenues après une couche de pooling sont [168] :

$$\frac{(nh - f + 1)}{s} * \frac{(nw - f + 1)}{s} * nc$$

- $nh$  : hauteur de la carte des caractéristiques.
- $nw$  : largeur de la carte des caractéristiques.
- $nc$  : nombre de canaux dans le vecteur des caractéristiques.
- $f$  : taille du filtre.
- $s$  : longueur de pas.

Les couches de pooling sont utilisées pour réduire les dimensions des vecteurs de caractéristiques. Ainsi, il réduit le nombre de paramètres à apprendre et la quantité de calculs effectués dans le réseau.

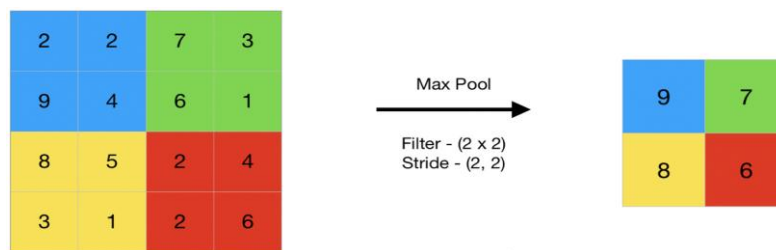
La couche de pooling résume les caractéristiques présentes dans une région du vecteur des caractéristiques générées par une couche de convolution. Ainsi, d'autres opérations sont effectuées sur les entités résumées au lieu des entités positionnées avec précision générées par la couche de convolution. Cela rend le modèle plus robuste aux variations de la position des caractéristiques dans l'image d'entrée.

Et ça les plus types courants de pooling :

### 3.4.1. Max pooling

Le max pooling est une opération de regroupement qui sélectionne l'élément maximal dans la région du vecteur des caractéristiques couvertes par le filtre. Ainsi, la sortie après la couche max-pooling serait un vecteur de caractéristiques contenant les caractéristiques les plus importantes du vecteur de caractéristiques précédentes [168].

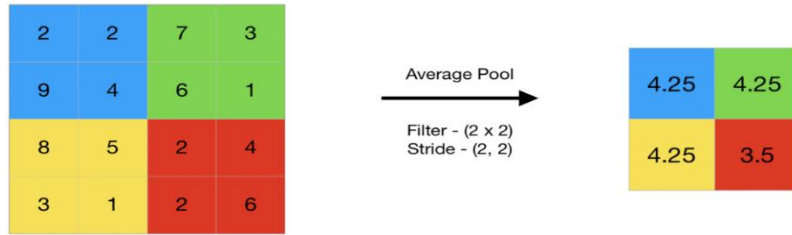
Et il est caractérisé par il est le meilleur pour extraire des caractéristiques plus prononcées comme les bords [169].



*Figure 2.13 : L'application de max pooling*

### 3.4.2. Average Pooling

Average Pooling calcule la moyenne des éléments présents dans la région du vecteur des caractéristiques couvertes par le filtre. Ainsi, alors que le max pooling donne la caractéristique la plus importante dans un patch particulier du vecteur des caractéristiques [168], l'average pooling donne la moyenne des caractéristiques présentes dans un patch. Et il est caractérisé par des fonctionnalités d'extraction plus fluides que le max pooling [170].



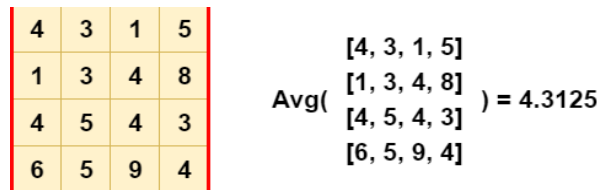
**Figure 2.14 :** L'application de average polling

Il existe également de nombreux pooling, y compris Global pooling, spatial pyramide pooling, cascade corner pooling.

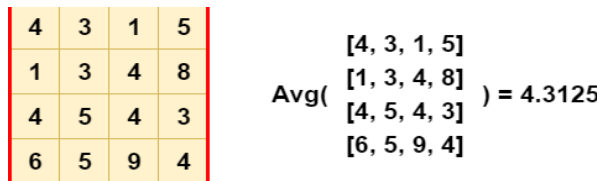
### 3.4.3. Global pooling

Global pooling réduit chaque canal du vecteur des caractéristiques à une valeur unique. Ainsi, une carte de caractéristiques  $nh * nw * nc$  est réduite à  $1 * 1 * nc$  vecteur de caractéristiques. Cela équivaut à utiliser un filtre de dimensions  $nh * nw$ , c'est-à-dire les dimensions de la carte des caractéristiques[168].

Caractérisé par réduit la dimensionnalité de 3d à 1d et élimine le besoin d'appliquer un aplatissement [171].



**Figure 2.15 :** L'application de global average pooling

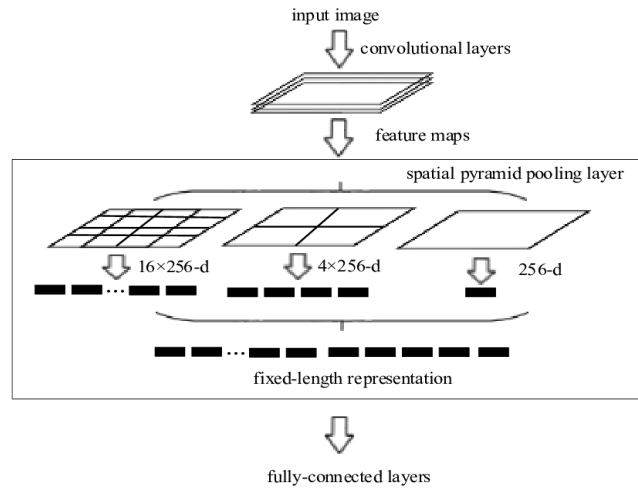


**Figure 2.16 :** L'application de global max pooling

### 3.4.4. Spatial pyramid pooling

SPP fonctionne en divisant les vecteurs de caractéristiques produites par la dernière couche convolutifs en un certain nombre de bacs spatiaux avec des tailles proportionnelles à la taille de l'image, de sorte que le nombre de bacs est fixe quelle que soit la taille de l'image. Les bacs sont capturés à différents niveaux de granularité [172].

La couche spatiale pyramid pooling peut traiter n'importe quelle image de taille et produire un vecteur de taille fixe, sans recadrer ni déformer l'image, ce qui améliore considérablement la précision [172].

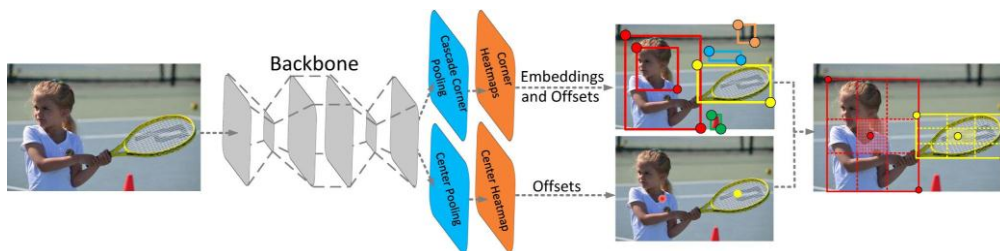


*Figure 2.17 : L'architecture de Spatial pyramid pooling*

### 3.4.5. Cascade Corner Pooling

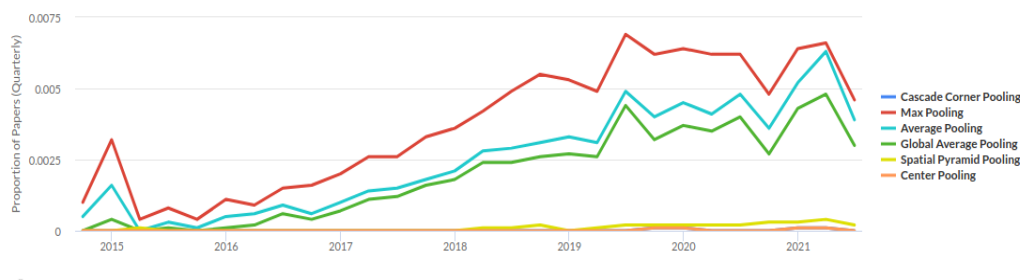
Cascade Corner Pooling est une couche de pooling pour la détection d'objets qui s'appuie sur l'opération de pooling des coins. Les coins sont souvent à l'extérieur des objets, ce qui manque de caractéristiques d'apparence locale [173].

Caractérisé par une meilleure localisation des coins en encodant des connaissances préalables explicites [173]



*Figure 2.18 : Diagramme de Cascade Corner Pooling*

Le backbone génère une carte de caractéristiques, et pour déterminer si un pixel Cap dans le vecteur de caractéristiques est un point clé central, nous devons trouver la valeur maximale dans ses directions horizontale et verticale et les additionner. Ce faisant, le center pooling permet une meilleure détection des points-clés centraux [173].

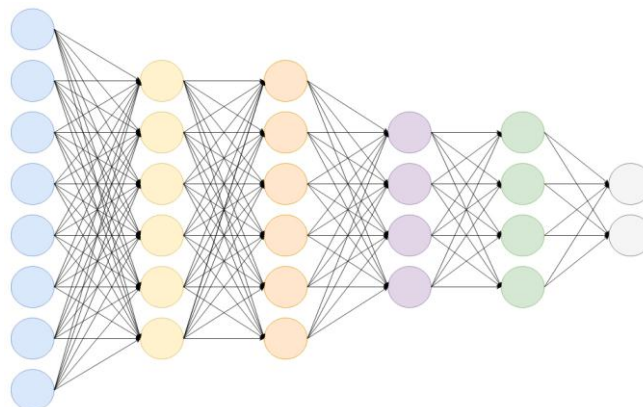


**Figure 2.19 :** Utilisation dans le temps des méthodes de pooling les plus connues

### 3.5. La couche entièrement connectée (Fully Connected Layer)

La couche entièrement connectée consiste simplement à alimenter en avant les réseaux de neurones. Et forment les dernières couches du réseau [174].

L'entrée de la couche entièrement connectée est la sortie de la couche finale de pooling de convolution, qui est aplatie puis introduite dans la couche entièrement connectée.



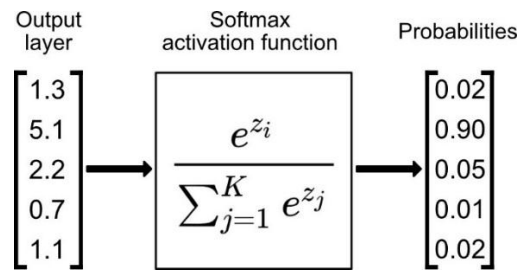
**Figure 2.20 :** L'architecture de la couche entièrement connectée

### 3.6. La fonction softmax

La fonction *softmax*, également connue sous le nom de *softargmax* ou fonction exponentielle normalisée est une généralisation de la fonction logistique à plusieurs dimensions. Il est utilisé dans la régression logistique multinomiale et est souvent utilisé comme dernière fonction d'activation d'un réseau de neurones pour normaliser la sortie d'un réseau à une distribution de probabilité sur les classes de sortie prédites, sur la base de l'axiome de choix de Luce [175].

La fonction softmax prend en entrée un vecteur  $z$  de  $K$  nombres réels et le normalise en une distribution de probabilité constituée de  $K$  probabilités proportionnelles aux exponentielles des nombres d'entrées. C'est-à-dire qu'avant d'appliquer softmax, certaines composantes vectorielles pourraient être négatives ou supérieures à un, et peut ne pas être égal à 1, mais après application de softmax, chaque composant sera dans l'intervalle  $[0,1]$ , et les composants s'additionneront jusqu'à 1, afin qu'ils puissent être interprétés comme des

probabilités. De plus, les composantes d'entrée plus grandes correspondront à des probabilités plus grandes [176].



*Figure 2.21 : la fonction softmax*

### 3.7. La couche de sortie

La couche de sortie dans une CNN, comme présentée précédemment, est une couche entièrement connectée, où l'entrée des autres couches est aplatie et envoyée de manière à transformer la sortie en le nombre de classes souhaité par le réseau.

La couche de sortie contient une étiquette qui se présente sous la forme d'un chiffre indiquant le résultat attendu.

### 3.8. La normalisation

La normalisation est une approche qui est appliquée lors de la préparation des données afin de modifier les valeurs des colonnes numériques dans un jeu de données pour utiliser une échelle commune lorsque les entités des données ont des intervalles différents. Par conséquent, il est utilisé pour faciliter l'optimisation en lissant la surface de perte du réseau. Il normalise chaque canal d'entrée sur un mini-lot. Pour accélérer l'apprentissage des réseaux de neurones convolutifs et réduire la sensibilité à l'initialisation du réseau, il est utilisé entre les couches convolutifs et les non-linéarités [177].

#### 3.8.1. Les méthodes de normalisation

N°	La méthode	La référence	L'année
1	Layer Normalization	Ba, J.L et al [142]	2016
2	Batch Normalization	Ioffe, S et al [143]	2015
3	Local Response Normalization	Krizhevsky et al [144]	2012
4	Instance Normalization	Ulyanov D et al [145]	2016
5	Spectral Normalization	Miyato T et al [146]	2018
6	Adaptive Instance Normalization	Ogasawara E et al [147]	2017
7	Weight Normalization	Salimans T et al [148]	2016
8	Conditional Batch Normalization	De Vries H et al [149]	2017
9	Group Normalization	Wu T et al [150]	2018
10	Activation Normalization	Kingma D.P et al [151]	2018

<b>11</b>	Weight Demodulation	Karras T et al [152]	2019
<b>12</b>	Switchable Normalization	Luo P et al [153]	2018
<b>13</b>	Local Contrast Normalization	Foracchia M et al [154]	2009
<b>14</b>	Weight Standardization	Qiao et al [155]	2019
<b>15</b>	Conditional Instance Normalization	Huang X et al [178]	2016
<b>16</b>	SyncBN	He K et al [179]	2018
<b>17</b>	Attentive Normalization	Li X et al [180]	2019
<b>18</b>	Decorrelated Batch Normalization	Huang L et al [181]	2018

**Tableau 2.1:** Les méthodes de normalisation les plus utilisées ordonnées par leur nombre d'utilisation [156]

### 3.9. Régularisation

Régularisation est le processus d'ajout d'informations afin de résoudre un problème défectueux ou d'éviter le sur-apprentissage dans des problèmes tels que la classification, le processus est courant en mathématiques, en statistiques et en informatique, en particulier en apprentissage automatique [176].

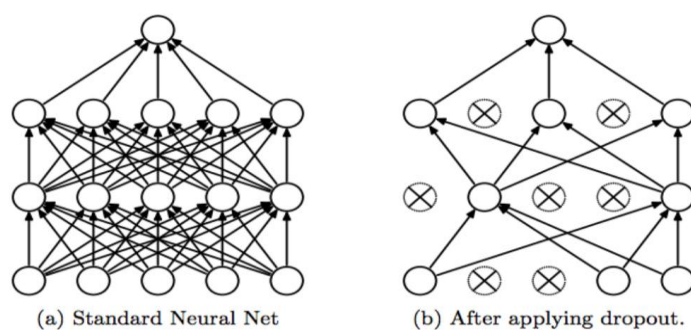
Les stratégies régularisation sont conçues pour réduire l'erreur de test des algorithmes d'apprentissage automatique, éventuellement au détriment de l'erreur d'apprentissage. Il existe de nombreuses formes régularisation dans le domaine de l'apprentissage en profondeur [176].

#### 3.9.1. La dilution (Dropout)

La dilution (également appelée Dropout) est une technique de régularisation pour réduire le sur-apprentissage dans les réseaux de neurones artificiels en empêchant les coadaptations complexes sur les données d'entraînement. C'est un moyen efficace d'effectuer un modèle avec des réseaux de neurones [183].

- Le terme dilution fait référence à l'amincissement des poids [184].
- Le terme dropout fait référence au "dropping out" aléatoire ou à l'omission d'unités (à la fois cachées et visibles) pendant le processus d'apprentissage d'un réseau de neurones [183] [185] [186].

L'amincissement des poids et les unités d'abandon déclenchent le même type de régularisation, et le terme dropout est souvent utilisé pour désigner la dilution des poids.



**Figure 2.22 :** L'application de dilution

La dilution force un réseau de neurones à apprendre des fonctionnalités plus robustes qui sont utiles en conjonction avec de nombreux sous-ensembles aléatoires différents des autres neurones.

La dilution double à peu près le nombre d'itérations nécessaires pour converger. Cependant, le temps d'entraînement pour chaque itération est moindre.

Avec  $H$  unités cachées, chacune pouvant être supprimée, nous avons  $2^H$  modèles possibles. En phase de test, l'ensemble du réseau est considéré et chaque activation est réduite d'un facteur  $p$  [187].

### 3.9.2. Méthodes de régularisation

N°	La Reference	La méthode
1	Srivastava N et al [235]	Dropout
2	Loshchilov I et al [236]	Weight Decay
3	Choe J et al [237]	Attention Dropout
4	Pereyra G et al [238]	Label Smoothing
5	Grandvalet Y et al [239]	Entropy Regularization
6	Yao Y et al [240]	Early Stopping
7	Kingma D.P et al [241]	Variational Dropout
8	Wan L et al [242]	DropConnect
9	Elden L et al [243]	R1 Regularization
10	Park M.Y et al [244]	L1 Regularization
11	Gal Y et al [245]	Embedding Dropout
12	Kim S.J et al [246]	Off-Diagonal Orthogonal
13	Oster H.S et al [247]	Temporal Activation Regularization

**Tableau 2.2:** Méthodes de régularisation ordonnée par leur nombre d'utilisation [156]

### 3.10. Optimisation

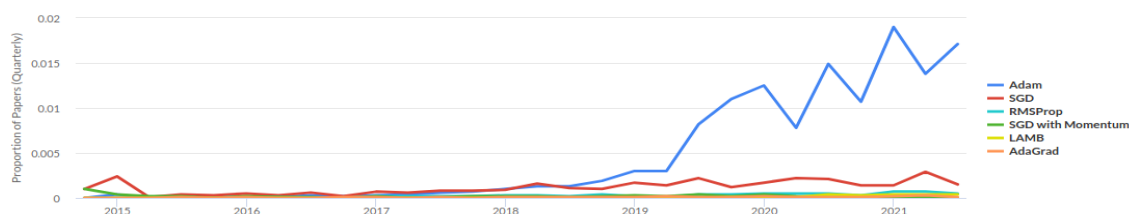
Les optimiseurs sont des algorithmes ou des méthodes utilisés pour modifier les attributs de votre réseau de neurones tels que les poids et le taux d'apprentissage afin de réduire les pertes.



### 3.10.1. Méthodes d'Optimisation

Méthodes	Avantages	Désavantages
<b>Gradient Descent</b>	<ul style="list-style-type: none"> <li>Calcul facile.</li> <li>Facile à mettre en œuvre.</li> <li>Facile à comprendre.</li> </ul>	<ul style="list-style-type: none"> <li>Peut piéger aux minima locaux.</li> <li>Les poids sont modifiés après le calcul du gradient sur l'ensemble de données. Ainsi, si l'ensemble de données est trop volumineux, cela peut prendre des années pour converger vers les minima.</li> <li>Nécessite une grande mémoire pour calculer le gradient sur l'ensemble de données.</li> </ul>
<b>Stochastic Gradient Descent</b>	<ul style="list-style-type: none"> <li>Les mises à jour fréquentes des paramètres du modèle convergent donc en moins de temps.</li> <li>Nécessite moins de mémoire car il n'est pas nécessaire de stocker les valeurs des fonctions de perte.</li> <li>Peut obtenir de nouveaux minima.</li> </ul>	<ul style="list-style-type: none"> <li>Variance élevée des paramètres du modèle.</li> <li>Peut tirer même après avoir atteint le minima global.</li> <li>Pour obtenir la même convergence que la descente de gradient, il faut réduire lentement la valeur du taux d'apprentissage.</li> </ul>
<b>Mini-Batch Gradient Descent</b>	<ul style="list-style-type: none"> <li>Met à jour fréquemment les paramètres du modèle et a également moins de variance.</li> <li>Nécessite une quantité moyenne de mémoire.</li> </ul>	
<b>Momentum</b>	<ul style="list-style-type: none"> <li>Réduit les oscillations et la forte variance des paramètres.</li> <li>Converge plus vite que la descente de pente.</li> </ul>	<ul style="list-style-type: none"> <li>Un autre hyper-paramètre est ajouté qui doit être sélectionné manuellement et avec précision.</li> </ul>
<b>Nesterov Accelerated Gradient</b>	<ul style="list-style-type: none"> <li>Ne manque pas les minima locaux.</li> <li>Ralentit si des minima se produisent.</li> </ul>	<ul style="list-style-type: none"> <li>Néanmoins, l'hyper-paramètre doit être sélectionné manuellement.</li> </ul>
<b>Adagrad</b>	<ul style="list-style-type: none"> <li>Le taux d'apprentissage change pour chaque paramètre d'entraînement.</li> <li>Pas besoin de régler manuellement le taux d'apprentissage.</li> <li>Capable de s'entraîner sur des données éparées.</li> </ul>	<ul style="list-style-type: none"> <li>Coûteux en calcul car il faut calculer la dérivée du second ordre.</li> <li>Le taux d'apprentissage est toujours en baisse, ce qui entraîne un entraînement lent.</li> </ul>
<b>AdaDelta</b>	<ul style="list-style-type: none"> <li>Maintenant, le taux d'apprentissage ne diminue pas et l'entraînement ne s'arrête pas.</li> </ul>	<ul style="list-style-type: none"> <li>Coûteux en calcul.</li> </ul>
<b>Adam</b>	<ul style="list-style-type: none"> <li>La méthode est trop rapide et converge rapidement.</li> <li>Rectifie le taux d'apprentissage en voie de disparition, la variance élevée.</li> </ul>	<ul style="list-style-type: none"> <li>Coûteux en calcul .</li> </ul>

**Tableau 2.3:** Avantages et désavantages des algorithmes d'optimisations populaire



**Figure 2.23 :** utilisation dans le temps des optimiseurs les plus connus

### 3.10.2. Méthodes d'optimisation les plus utilisées

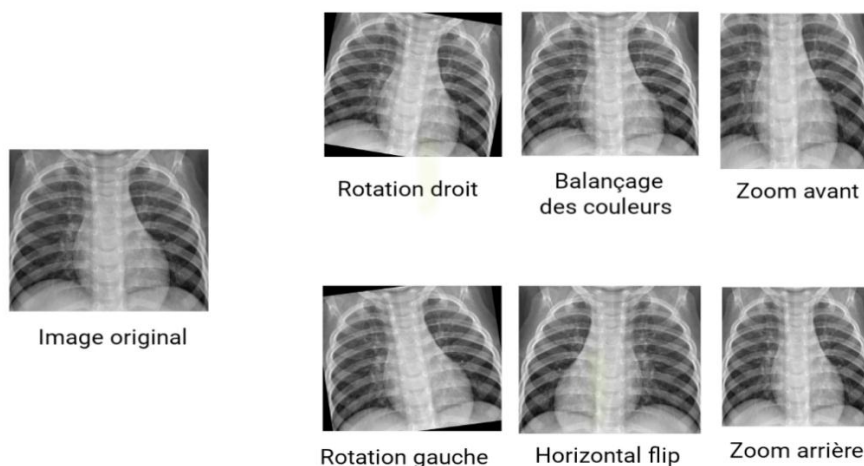
	nom	Reference	année
1	Adam	Kingma D.P et al [188]	2014
2	SGD	Robbins H and Monro S [189]	1951
3	ADMM	Boyd S et al [190]	2000
4	RMSProp	Hinton G [191]	2013
5	SGD with Momentum	Ning Q et al [192]	1999
6	AdaGrad	Duchi J et al [193]	2011
7	TTUR	Heusel M et al [194]	2017
8	Gradient Clipping	Pascanu et al [195]	2000
9	LAMB	You Y et al [196]	2019
10	AMSGrad	Reddi et al [197]	2019
11	Nesterov Accelerated Gradient	Nesterov Y [198]	1983
12	Adafactor	Shazeer M et al [199]	2018
13	Natural Gradient Descent	Ratray et al [200]	1998
14	LARS	You Y et al [201]	2017
15	Population Based Training	Jaderberg M et al [202]	2017

*Tableau 2.4:* Méthodes d'optimisation les plus utilisées [156]

## 4. L'augmentation des données

L'augmentation des données dans l'analyse des données est une technique utilisée pour augmenter la quantité de données en ajoutant des copies légèrement modifiées de données déjà existantes ou de données synthétiques nouvellement créées à partir de données existantes.

Il agit comme un régulateur et aide à réduire le sur-apprentissage lors de l'entraînement d'un modèle d'apprentissage automatique [203]. Elle est étroitement liée au sur échantillonnage dans l'analyse des données.

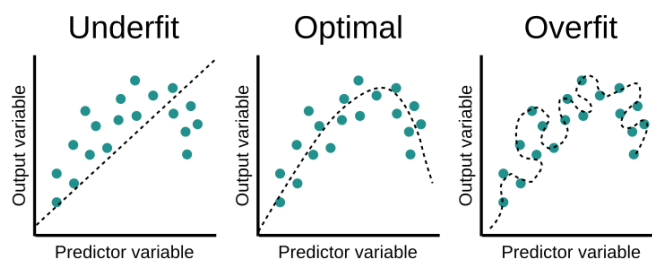


*Figure 2.24 :* L'augmentation d'image

## 5. Le sur-apprentissage et le sous-apprentissage dans le CNN (Overfitting and Underfitting)

Le sous-apprentissage signifie que l'équation mathématique utilisée est plus faible et ne couvre pas tous les points présents [204].

Le sur-apprentissage signifie que l'équation mathématique utilisée ajuste complètement tous les points. Cela provoque des erreurs de prédiction [204].



*Figure 2.25 : Graphiques de sur-apprentissage et le sous-apprentissage et l'optimale fitting*

### 5.1. Le sur-apprentissage [204]

- ✓ Lorsque l'équation passe sur tous les points exactement et exagérément.
- ✓ La forme est belle, mais le problème est de s'attendre à une nouvelle valeur.
- ✓ Cela vient souvent d'une augmentation inutile des données (soit le nombre d'échantillons  $m$  ou le nombre d'éléments  $n$ ), ou le choix de la mauvaise équation.

### 5.2. Méthodes pour éviter le sur-apprentissage

- **Réseaux de neurones de recyclage (Retraining neural networks)**

Exécuter le même modèle sur le même ensemble d'entraînement mais avec des poids initiaux différents, et sélectionner le réseau avec les meilleures performances.

- **Réseaux de neurones multiples (Multiple neural networks)**

Entraîner plusieurs modèles de réseaux de neurones en parallèle, avec la même structure mais des poids différents, et faire la moyenne de leurs sorties.

- **Arrêt anticipé (Early stopping)**

L'entraînement du réseau et surveillance de l'erreur sur l'ensemble de validation après chaque itération, et arrêt de l'entraînement lorsque le réseau commence à overfit les données.

- **Régularisation (Regularization)**

L'ajout d'un terme à l'équation de la fonction d'erreur, destiné à réduire les poids et les biais, à lisser les sorties et à rendre le réseau moins susceptible de over-fit.

- **Rapport de performance de réglage (Tuning performance ratio)**

Similaire à la régularisation, mais en utilisant un paramètre qui définit de combien le réseau doit être régularisé.

### 5.3. Sous-apprentissage [204]

- ✓ L'équation est simple, petite et incapable de comprendre et de passer par la plupart des points existants
- ✓ Il a un gros problème pour prédire les valeurs à venir et le taux d'erreur est grand
- ✓ Cela provient souvent de points de données (soit le nombre d'échantillons  $m$  ou le nombre d'éléments  $n$ ), ou le choix de la mauvaise équation

### 5.4. Méthodes pour éviter le sous-apprentissage

- **Ajout de couches de neurones ou d'entrées (Adding neuron layers or inputs)**

L'ajout de couches de neurones ou l'augmentation du nombre d'entrées et de neurones dans chaque couche peuvent générer des prédictions plus complexes et améliorer l'ajustement du modèle.

- **Ajout d'échantillons de l'entraînement supplémentaires ou amélioration de la qualité (Adding more training samples or improving quality)**

Plus vous introduisez d'échantillons d'apprentissage dans le réseau et mieux ils représentent la variance de la population réelle, meilleures sont les performances du réseau.

- **La dilution (Dropout)**




Supprimer au hasard un certain pourcentage de neurones à chaque itération d'entraînement. Cela garantit que certaines informations apprises sont supprimées de manière aléatoire, réduisant ainsi le risque de sur apprentissage.

- **Paramètre de régularisation décroissant (Decreasing regularization parameter)**

La régularisation peut être exagérée. En utilisant un paramètre de performance de régularisation, vous pouvez connaître le degré optimal de régularisation, ce qui peut aider le modèle à mieux s'adapter aux données.

## 6. Framework plus utilisées pour l'implémentation d'un CNN

Framework	Points forts	Lugo
<b>TensorFlow [205].</b>	<ul style="list-style-type: none"> <li>• Prise en charge robuste de plusieurs GPU</li> <li>• Visualisation de graphiques et files d'attente à l'aide de TensorBoard.</li> <li>• Connu pour être complexe et a une courbe d'apprentissage abrupte.</li> <li>• Excellente documentation et soutien de la communauté.</li> </ul>	
<b>PyTorch [206].</b>	<ul style="list-style-type: none"> <li>• Excellent en prototypage rapide.</li> <li>• Prise en charge solide des GPU car des programmes parallèles peuvent être implémentés sur plusieurs GPU.</li> <li>• Fournit une interface plus propre et est plus facile à utiliser.</li> <li>• Facilite l'échange de données avec des bibliothèques externes.</li> </ul>	
<b>DeepLearningG4J[207].</b>	<ul style="list-style-type: none"> <li>• Rassemble l'ensemble de l'écosystème Java pour exécuter le deep learning.</li> <li>• Peut traiter rapidement des quantités massives de données.</li> <li>• Comprend à la fois des frameworks d'apprentissage en profondeur multithread et monothread.</li> <li>• Peut être administré en plus de Hadoop et Spark.</li> </ul>	
<b>The Microsoft Cognitive Toolkit[208].</b>	<ul style="list-style-type: none"> <li>• Très efficace et évolutif pour plusieurs machines.</li> <li>• Pris en charge par des interfaces telles que Python, C++ et Command Line.</li> <li>• Convient aux cas d'utilisation d'image, d'écriture manuscrite et de reconnaissance vocale.</li> <li>• Prend en charge les réseaux de neurones de type RNN et CNN.</li> </ul>	
<b>Keras [209].</b>	<ul style="list-style-type: none"> <li>• API faciles à comprendre et cohérentes</li> <li>• S'intègre parfaitement au flux de travail TensorFlow.</li> <li>• Prend en charge plusieurs backends d'apprentissage en profondeur.</li> <li>• Prise en charge intégrée de l'entraînement distribuée et du parallélisme multi-GPU.</li> </ul>	
<b>ONNX [210].</b>	<ul style="list-style-type: none"> <li>• Fournit l'interopérabilité et la flexibilité.</li> <li>• Fournit des environnements d'exécution et des bibliothèques compatibles.</li> <li>• Liberté d'utiliser le framework préféré avec un moteur d'inférence sélectionné.</li> </ul>	

	<ul style="list-style-type: none"> <li>• Maximise les performances sur l'ensemble du matériel.</li> </ul>	
<b>MXNet [211].</b>	<ul style="list-style-type: none"> <li>• Programmation hybride qui offre le meilleur de la programmation impérative et symbolique.</li> <li>• Fournit un entraînement distribué.</li> <li>• Prend en charge le déploiement dans différents langages tels que Java, Scala, R, Julia, C++, Perl et Clojure</li> <li>• Presque linéaire sur les clusters GPU qui offre une excellente évolutivité</li> </ul>	
<b>Caffe [212].</b>	<ul style="list-style-type: none"> <li>• La bibliothèque C++ est livrée avec une interface Python.</li> <li>• La configuration définit des modèles sans codage en dur.</li> <li>• Plus facile à installer et à former, sans avoir à s'appuyer sur le réseau</li> <li>• La prise en charge des réseaux de neurones récurrents est assez faible</li> </ul>	

*Tableau 2.5: Top huit Framework utilise dans l'apprentissage en profondeur avec leur point fort*

## 7. Conclusion

Dans ce chapitre, nous avons présenté les bases de CNN, en commençant par ses principales bases telles que la convolution et le pooling, puis en passant à ses types de couches et à ses fonctions détaillées.

Ensuite, nous avons présenté les différentes façons d'améliorer le travail de CNN en utilisant des méthodes de normalisation, de régularisation et d'optimisation.

Ensuite, nous avons fait une comparaison des Framework les plus couramment utilisés. En réalisant tout ce que nous avons appris dans ce chapitre, nous pouvons construire un réseau CNN spécialisé qui classe et divulgue le meilleur possible.

Dans ce projet, nous essaierons de construire CNN, comme nous l'avons appris, et comment cela fonctionnera et la conception dans le prochain chapitre.

## Chapitre 03

# La conception

## 1. Introduction

Dans ce Chapitre nous allons expliquer comment ce système fonctionne en utilisant les meilleures et les plus récentes techniques d'apprentissage profond et d'intelligence artificielle qui sont appropriées et compatibles avec le domaine de l'imagerie médicale et de la classification des maladies pulmonaires on suivant ces principales étapes :

- ✓ La recherche des datasets confirmés par PCR puis la préparation de notre dataset.
- ✓ La recherche et la réalisation d'un modèle en rapport avec le domaine de la classification des maladies pulmonaires.
- ✓ L'entraînement et la préparation du modèle pour la prédiction.
- ✓ Estimation du succès et tester la performance du modèle.

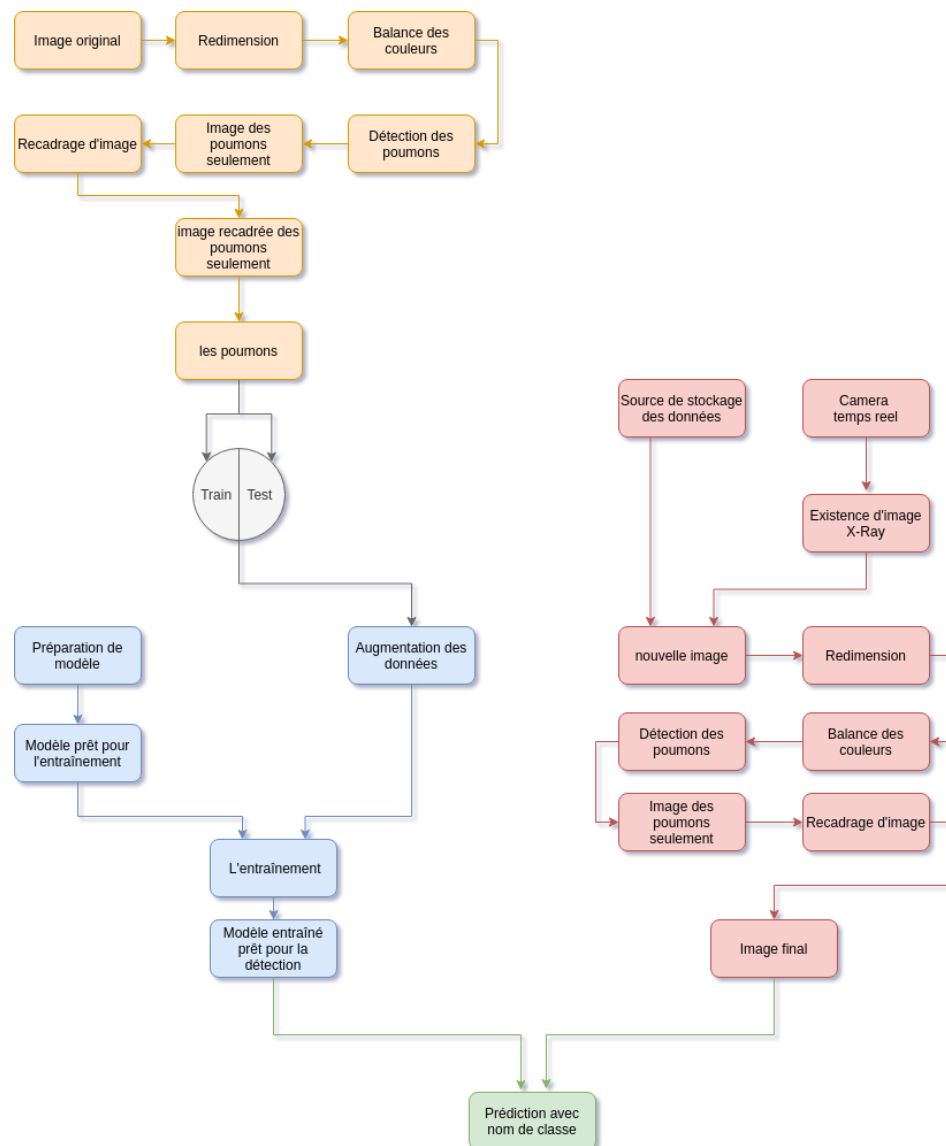
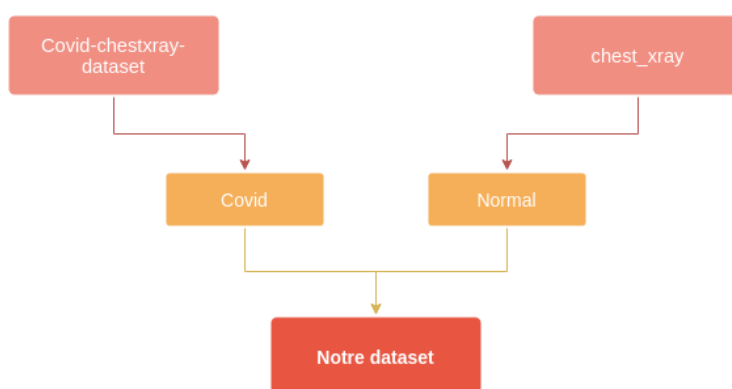


Figure 3.1 : Diagramme principale de la conception



## 2. La recherche des bases de données confirmées par PCR et la préparation de notre base de donnée

nous avons utilisée comme base de donnée dans ce projet des parties d'autres bases de données connexes de sorte que nous prenons les parties qui nous intéressent seulement, cette base de données se compose de deux classes principales, la première classe est la partie des images médicales thoracique des patients de Covid elle est pris de cette base **covid-chestxray-dataset** [141], Quant à la deuxième classe est la partie des images médicales thoraciques des patientes normales, qui était prise de cette base **chestxray** [140].



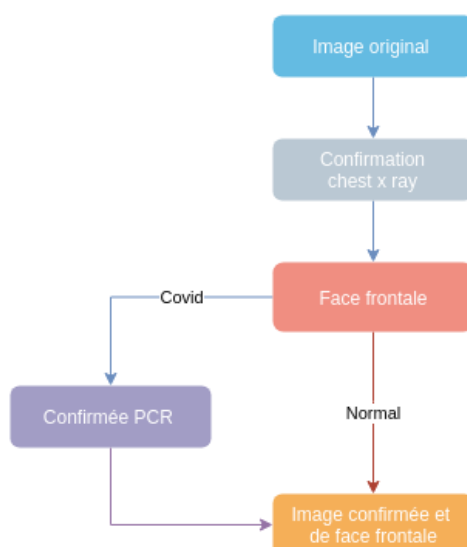
*Figure 3.2 : Diagramme de collecte de base de données*

Cette étape peut être détaillée et divisée en plusieurs sous étapes, qui sont les suivantes :

### 2.1. Le Filtrage et nettoyage des bases des données

Dans la partie consacrée aux images médicales thoraciques des patients Covid, nous avons extrait juste les données confirmées par le test de réaction en chaîne par polymérase (PCR) et également de face frontale, quant aux images médicales des patientes Non atteint de maladies pulmonaires, nous avons pris un ensemble des données claires et de haute qualité.

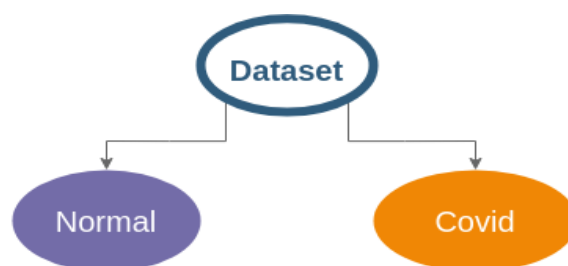
Considérant que les données prises pour les deux classes sont d'abord confirmées comme des radiographies Rayon-X.



*Figure 3.3 : Diagramme de vérification et confirmation de la qualité des images*

## 2.2. La création d'une base de données

Pour créer une nouvelle base de donnée fiable et de données confirmées nous avons utilisé des images provenant des deux bases de données précédentes, cette base est une nouvelle dataset composée de deux classes principales, une classe nommée "Covid" pour les images médicales thoraciques des patients Covid et la seconde nommée "Normal" pour les images médicales thoracique des patients normaux. Chaque classe est composée d'environ 200 images confirmées originales et non augmentée.



*Figure 3.4 : diagramme global de notre base de données*

## 2.3. L'application de la balance des couleurs pour l'amélioration de la qualité de l'image

Ces images sont des images naturelles dont la qualité doit être améliorée en fonction du domaine dans lequel elles sont utilisées.

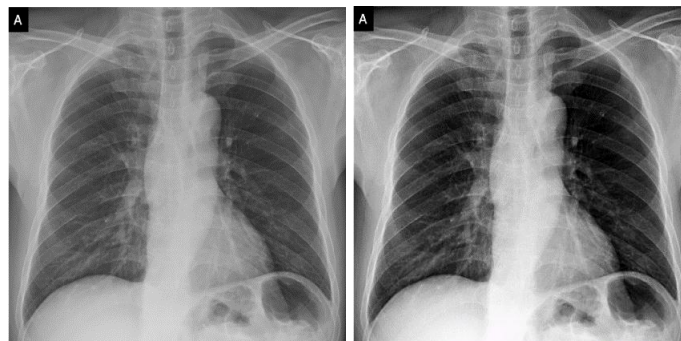
Dans le domaine des images médicales, nous avons utilisé le Balançage des couleurs pour faire apparaître les détails fins, l'algorithme est appelé Simplest Color Balance.

### 2.3.1. L'algorithme de balance des couleurs

L'algorithme de balance des couleurs (simplest color balance) [19], part de l'hypothèse que les valeurs les plus élevées de R, V, B observées dans l'image doivent correspondre au blanc et les valeurs les plus faibles à l'obscurité. L'algorithme étire simplement, autant qu'il le peut, les valeurs des trois canaux Rouge, Vert, Bleu (R, V, B), afin qu'ils occupent la plage maximale possible [0, 255] en appliquant une transformée affine  $ax + b$  à chaque canal.

Étant donné que de nombreuses images contiennent quelques pixels aberrants qui occupent déjà les valeurs 0 et 255, la méthode proposée sature un petit pourcentage des pixels avec les valeurs les plus élevées à 255 et un petit pourcentage des pixels avec les valeurs les plus faibles à 0, avant d'appliquer la transformée affine.

Le plus gros problème dans la plupart de ce type d'images (X-Ray) est que les couleurs ne sont pas équilibrées, c'est ce que nous appelons une color cast, car tout dans l'image semble avoir été décalé vers une couleur ou une autre. L'équilibrage des couleurs peut être pensé en termes de suppression de cette color cast. L'équilibrage des couleurs d'une image affecte non seulement les neutres, mais également les autres couleurs. L'idée est que dans une photo bien équilibrée, la couleur la plus lumineuse doit être le blanc et le noir le plus foncé.

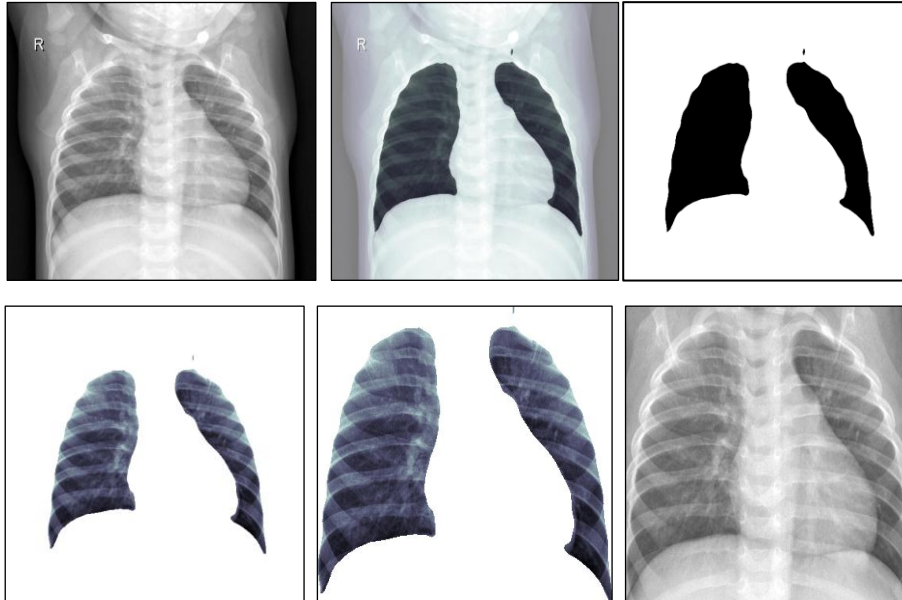


*Figure 3.5 : Balançage des couleurs*

### 2.4. La détection des poumons et la sélection des zones requises

Pour un bon modèle et pour des résultats performants, il faut travailler seulement sur les poumons et non pas sur l'image Rayon-X complète, car si nous donnons des images complètes sans spécifier les zones d'intérêt dans lesquelles le modèle doit s'entraîner, cela pourrait affecter les résultats de la prédiction dans le futur.

D'où l'idée de sélectionner uniquement la zone sur laquelle se trouve le poumon dans l'image, pour détecter cette zone, nous avons utilisé un **modèle pré-entraîné** [228] pour la détection des poumons et pu couper uniquement la zone souhaitée. Afin de détecter les poumons, le modèle de détection des poumons utilisé est un modèle qui accepte des images d'une résolution de 512 \* 512 pixels.



*Figure 3.6 : Les principales étapes d'obtention de l'image finale*

## 2.5. L'obtention des images finales et construction de la base de données finale

Après avoir effectué le processus d'amélioration et de découpage et le recadrage, nous arrivons à l'étape de la construction d'une base de données finale, composée de deux classes principales, la classe covid et la classe normale.

Les images de cette base sont des images ne contenant que la zone d'intérêt et chaque image traitée chromatiquement et de source radiographique Rayon-X et de face frontale, et pour la classe covid, elle est confirmée par le PCR.

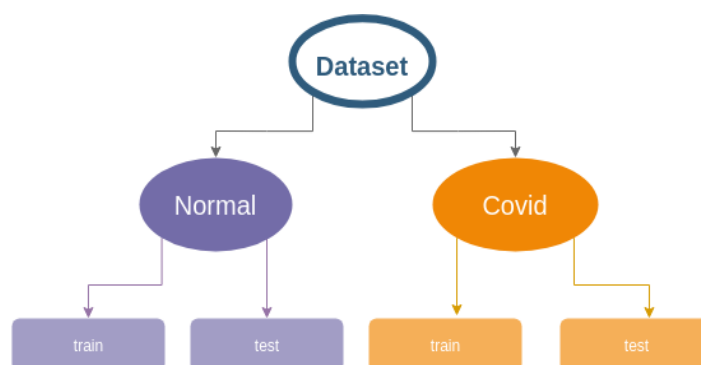
## 2.6. La division de la base de données

À cette étape, nous divisons chaque classe en deux parties, une partie pour l'apprentissage et une autre pour le test, en adoptant certains pourcentages comme il sont présentés dans le tableau suivant.

	L'apprentissage	Le test
<b>Pourcentage</b>	75%	25%
	66.7%	33.3%
	50%	50%

**Tableau 3.1** : pourcentages utilisés pour la division de la base de données

Et pour la division qui donne le meilleur résultat pour notre modèle de classification est la première (75%, 25%).

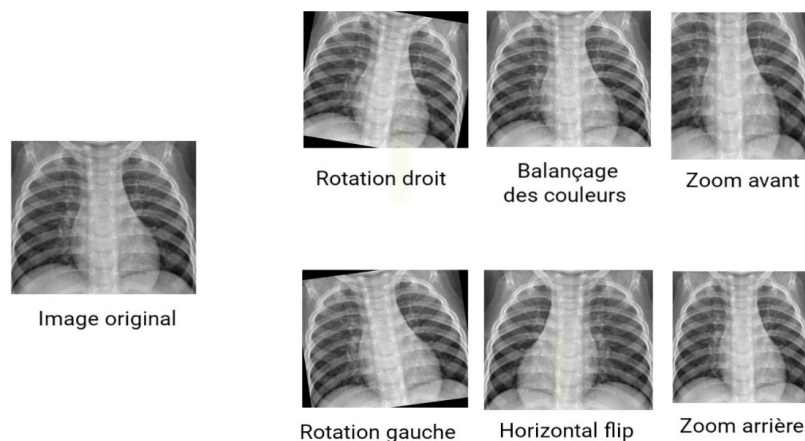


**Figure 3.7** : diagramme détaillé de notre base de données

## 2.7. L'augmentation des données

Les images médicales ne peuvent pas toujours être verticales dans une grande proportion ou être claires, et puisque notre application travaille aussi en temps réel aussi, nous devons prendre des mesures pour cette situation, et ces mesures sont des ajustements morphologiques ou aussi appelés l'augmentation.

C'est pourquoi nous devons faire ces augmentations des images, ces augmentations incluent la rotation à droite et à gauche d'un certain angle, ainsi que le zoom avant et arrière d'un certain pourcentage, ainsi que le retournement de l'image horizontalement. Tout ceci permet à notre modèle d'apprendre plus profondément et contribue également à réduire le taux de perte de manière très significative.



*Figure 3.8 : l'augmentation des données*

### 3. La recherche d'un modèle adapté

#### 3.1. Expérimentation des modèles existants

Dans ce projet, nous avons testé de nombreux modèles existants tels que inceptionV3, vgg16, vgg19 en remplaçant et modifiant la dernière couche du modèle selon les besoins et en effectuant un nouvel apprentissage dans ce qui est connu comme l'apprentissage par transfert.

Les résultats de ces entraînements sont toujours en sur-apprentissage malgré de nombreuses tentatives pour corriger la situation en changeant les valeurs des coefficients en fonction de ce qui convient aux images médicales, mais toutes les modifications n'ont pas donné le résultat souhaité, et la raison en est l'architecture interne du modèle qui est très profonde ou large ainsi que les classes ne demandent pas tout cette profondeur, qui entraîne le sur-apprentissage et augmenter l'erreur lors de la prise prédiction.

Cela nous a fait penser à d'autres solutions en essayant de faire un modèle CNN simple et non compliqué afin qu'il n'y ait pas de problèmes comme avec les modèles précédents.

#### 3.2. La création de notre modèle

En raison des problèmes des modèles qui ne donnent pas les résultats souhaités dans certains domaines car leur structure interne profonde et large qui ne correspond pas à la qualité des données, en particulier dans le domaine des images médicales thoraciques, les informations qu'ils contiennent sont très proches et il est même difficile de les distinguer à l'œil humain, donc la solution optimale après plus de 40 expériences (plusieurs cas de différents paramètres) est de créer un modèle correspondre les conditions de profondeur et

de largeur nécessaires, ainsi que le nombre approprié de couches, en tenant compte de la taille de l'image pour maintenir la qualité des données et pour entraîner ce modèle sur la base de données finale.

En travaillant avec cette idée, les résultats de l'entraînement et des tests étaient très bons, et le taux d'erreur était faible et presque inexistant, ce qui nous a permis de juger du succès la performance de ce modèle.

Par conséquent, la structure du modèle peut être déterminée en sélectionnant les hyper paramètres du modèle, qui sont :

- ✓ Le nombre de couches convolutifs.
- ✓ Le type de fonctions d'activation pour chaque couche.
- ✓ Le nombre d'unités cachées par couche.

Ces hyper paramètres peuvent être sélectionnés en copiant des recherches/études existantes ou en effectuant un apprentissage par transfert ou dans notre cas, nous construisons une nouvelle expérience.

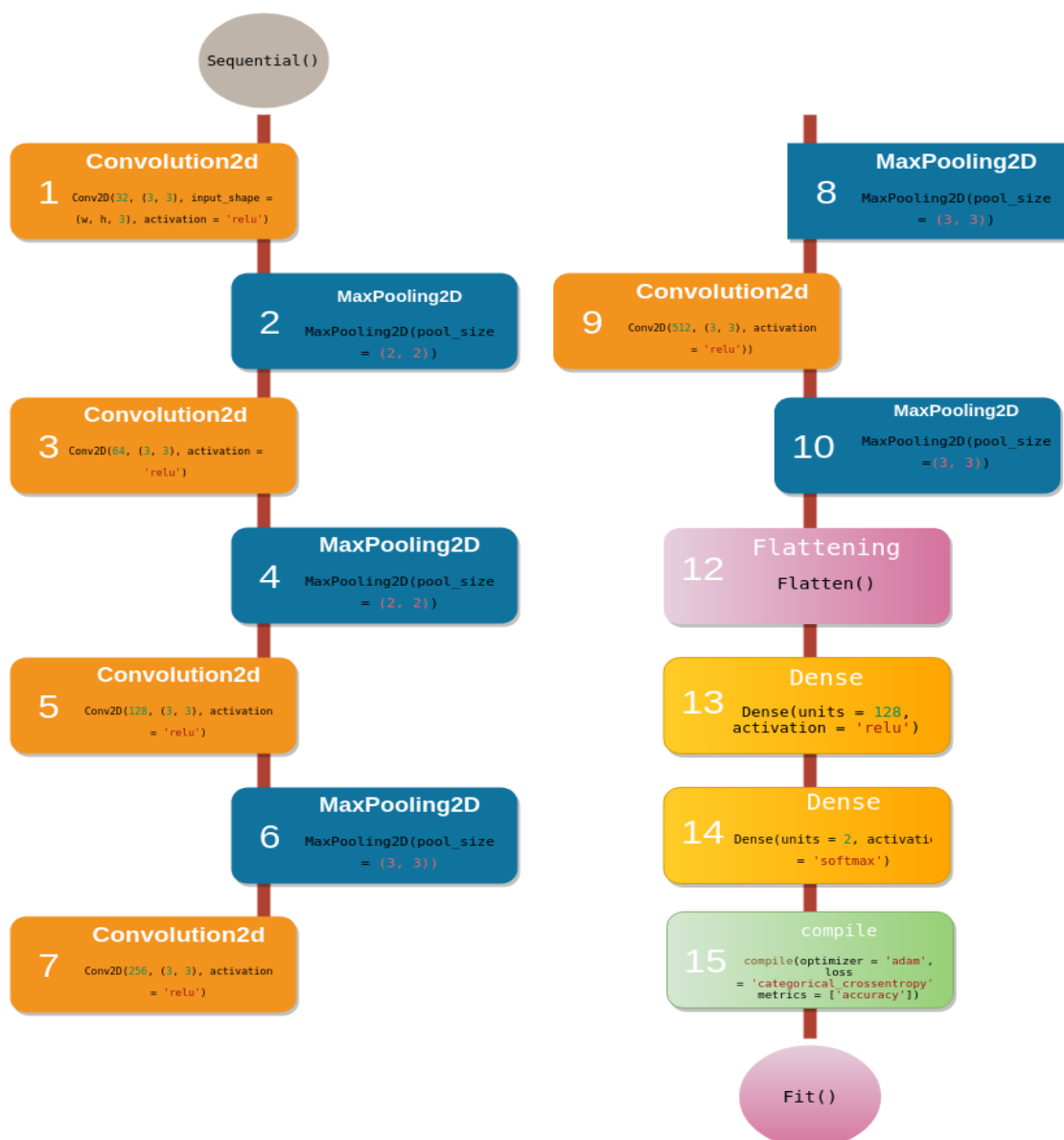


Figure 3.9 : diagramme principale de notre CNN

Dans ce modèle, qui contient de nombreuses couches, et grâce à de nombreuses expérimentations, le résultat du choix de ces hyper paramètres était le meilleur parmi de nombreuses autres options.

Qui est choisi dans :

- ✓ Cinq couches de convolution2D qui diffèrent par le nombre de filtres, de sorte qu'il double au fur et à mesure que l'on s'enfonce, en commençant par la première couche, qui contient 32 filtres, et en terminant par la dernière couche, qui contient 512 filtres, tout cela afin d'extraire de nombreuses caractéristiques.



- ✓ Cinq couches de max pooling afin d'extraire les informations importantes de la couche de convolution2D précédente.
- ✓ Couche de flattening afin de rendre l'information en une seule dimension.
- ✓ Puis une couche de Dense avec 128 unités.

Dans toutes ces couches, la fonction d'activation relu a été utilisée car elle est la plus avancée par rapport aux autres fonctions.

- ✓ Enfin, la couche de dense avec deux unités, en raison du nombre de classes finales a été utilisée dans laquelle la fonction SoftMax a été utilisée car c'est la fonction la plus utilisée dans les modèles multi-classes.

#### 4. L'entraînement et la préparation du modèle

Afin d'entraîner le modèle, il faut des données d'entraînement et pour la validation aussi il faut des données de test.

Une fois que nous sommes satisfaits du résultat de test du modèle, nous pouvons l'utiliser pour faire des prédictions sur des nouvelles données (*dans notre cas, on a utilisé pour la généralisation de résultats cinq autres bases que le modèle n'a jamais vu « voir section 5.4 »*).

Et pour les propriétés et paramètres du modèle commençant par le nombre d'*epochs* il est non déterminé, il est lié à la fonction d'arrêt **Early Stopping** Pour éviter l'**overfitting** et à la base de cette fonction le modèle détermine et calcul le nombre d'*epochs* (les *epochs* sont le nombre de répétitions typiques de l'entraînement) est généralement effectué avec des permutations et des sélections aléatoires des éléments de l'ensemble de données. Ensuite, nous avons enregistré le modèle pour une utilisation ultérieure.

Pour configurer notre modèle d'entraînement, nous devons appeler la méthode compilation (compile) avec la fonction de perte que nous voulons utiliser, le type d'optimisation et les métriques que notre modèle doit évaluer pendant l'entraînement et les tests nous utiliserons :

- ✓ **L'optimiser adam** est dû au fait que, comme mentionné précédemment, il s'agit de l'un des optimiseurs les plus utilisés car c'est la méthode la plus rapide, très rapide et qui converge également rapidement pour corriger la latence du taux d'apprentissage et le contraste élevé.
- ✓ **La fonction de perte** Categorical Cross entropy parce que cette fonction de perte est utilisée lorsqu'il existe deux ou plusieurs classes d'étiquettes.
- ✓ **Metrics** est une fonction utilisée pour évaluer les performances de votre modèle.

Les fonctions métriques sont similaires aux fonctions de perte, sauf que les résultats de l'évaluation d'une métrique ne sont pas utilisés lors de l'apprentissage du modèle. On peut utiliser n'importe quelle fonction de perte comme métrique.

Nous utilisons l'accuracy, cette métrique crée deux variables locales, total et count, qui sont utilisées pour calculer la fréquence à laquelle Y-pred (la classe prédit) correspond à Y\_true (la classe correcte). Cette fréquence est finalement renvoyée sous forme de précision binaire : une opération qui divise simplement le total par le nombre.

Et pour faire le fitting de notre modèle, nous avons sélectionné et choisi quelque paramètre pour le meilleur résultat :

- ✓ **Target\_size** = (512, 512) : pour garder le maximum d'information
- ✓ **Batch\_size** = 8 : Parce que la base de données n'est pas très volumineuse et qu'il s'agit d'apprendre par petits lots, nous avons donc utilisé ce lot afin de travailler avec moins d'effort et de mémoire et pour qu'il y ait une fluctuation modérée de l'entraînement.
- ✓ **Class\_mode** = 'categorical' parce qu'on a deux classes, la classe normale et la classe covid.

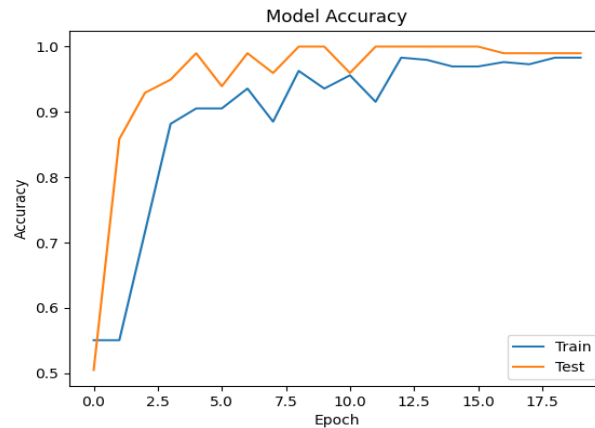
Pour un meilleur entraînement et pour éviter l'overfitting nous avons utilisé une technique Early Stopping, en apprentissage automatique, l'arrêt précoce ou Early Stopping est une forme de régularisation utilisée pour éviter le sur apprentissage lors de l'entraînement d'un apprenant avec une méthode itérative, telle que la descente de gradient. De telles méthodes mettent à jour l'apprenant afin qu'il s'adapte mieux aux données d'apprentissage à chaque itération. Jusqu'à un certain point, cela améliore les performances de l'apprenant sur des données en dehors de l'ensemble d'apprentissages et pour les paramètres de cette technique nous avons choisi :

- ✓ **Le monitor** avec le paramètre 'val\_loss'.
- ✓ **La patience** égale 7.
- ✓ **Le mode** automatique.

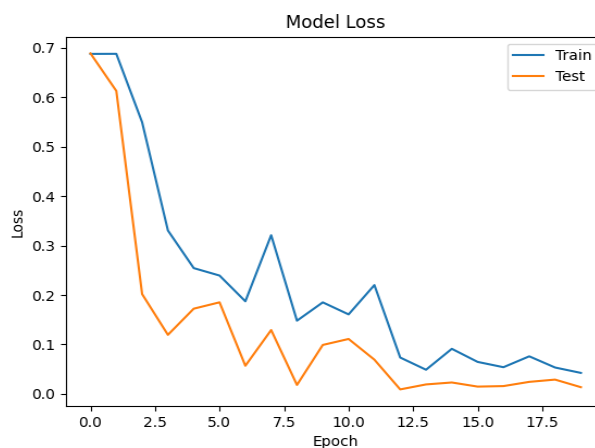
Et enfin nous devons faire le fitting avec ces paramètres sur l'architecture du modèle.

Nous validons le modèle selon le taux de précision et le taux d'erreur, et comme la base est divisé en deux parties d'apprentissage et de test alors pour chaque itération d'apprentissage, il y a le taux de précision de l'apprentissage (accuracy) et le test (val-accuracy) et le taux d'erreur pour l'apprentissage (loss) et le test (val-loss).

Les figures montrent les résultats obtenus pour le modèle où nous avons obtenu l'accuracy suivante : 98.98% (à partir des valeurs sue la table 3.2) pour le taux de précision et 0.0096 (à partir des valeurs sue la table 3.2) pour le taux d'erreur.



**Figure 3.10** : L'accuracy de model



**Figure 3.11** : loss de model

## 5. Estimation de la performance du modèle

Enfin, nous chargeons les données de test (images) qui sont passées par l'étape de prétraitement aussi.

Nous prédisons ensuite les classes pour ces images en utilisant le modèle entraîné et nous estimons la performance du modèle en comparant le nombre de prédictions correctes et d'erreurs.

Ce modèle dans ce domaine de classification dans les images radiologiques médicales atteignant un taux de accuracy comme nous avons dit précédemment de 98.98 % et un taux d'erreur de 0.96%.

	précision	recall	f1-score
<b>covid</b>	1.00	0.98	0.99
<b>normal</b>	0.98	1.00	0.99
<b>accuracy</b>			0.99
<b>macro avg</b>	0.99	0.99	0.99
<b>weighted avg</b>	0.99	0.99	0.99

*Tableau 3.2 : résultat de rapport de classification*

De ce tableau, nous concluons :

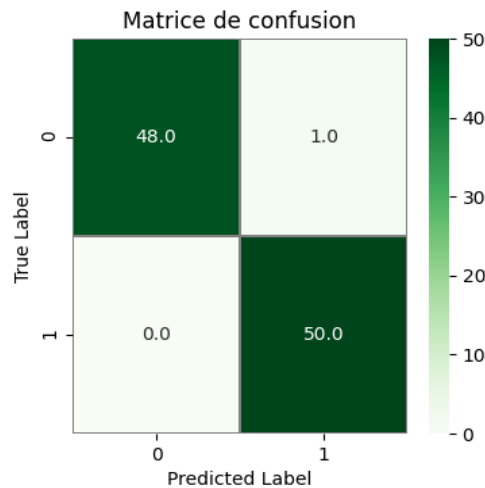
- ✓ Accuracy de model =  $\frac{TruePos+TrueNeg}{(TruePos+FalsePos)+(TrueNeg+FalseNeg)}$  : 98.98989898989899
- ✓ Precision =  $\frac{TruePos}{(TruePos+FalsePos)}$  : 98.0392156862745
- ✓ Recall =  $\frac{TruePos}{(TruePos+FalseNeg)}$  : 100.0
- ✓ F1\_score =  $2 * \frac{Precision * Recall}{Precision + Recall}$  : 99.009900990099
- ✓ Accuracy score : 0.98989898989899
- ✓ Log loss score : 0.009661452789872429

### 5.1. La matrice de confusion

La matrice de confusion est une matrice qui mesure la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée. La cellule ligne L, colonne C contient le nombre d'éléments de la classe réelle L qui ont été estimés comme appartenant à la classe C.

Un des intérêts de la matrice de confusion est qu'elle montre rapidement si un système de classification parvient à classifier correctement. Dans notre cas, nous avons deux classes réelles, chacune étant la classe attribuée à test à partir des données d'origine.

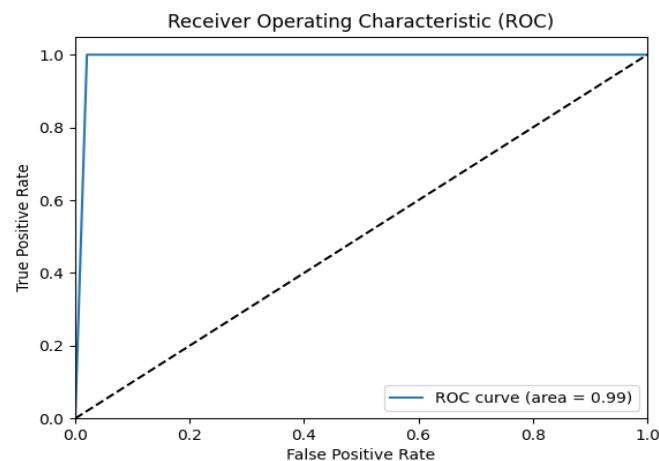
- 0 pour la classe « Covid »
- 1 pour la classe « Normal »



**Figure 3.12** : matrice de confusion

## 5.2. La fonction de l'efficacité du récepteur

La courbe de la fonction de l'efficacité du récepteur est plus couramment appelée la courbe ROC pour l'abréviation de l'anglais **Receiver Operating Characteristics**. Une simple représentation graphique des probabilités de DC (détection correcte en anglais True Positive) en fonction des probabilités des FA (fausse alarme ou en anglais False Positives) permet de visualiser les évolutions de la stratégie de l'observateur.



**Figure 3.13** : fonction d'efficacité du récepteur

## 5.3. Classification de Covid-19 en temps réel

Nous avons pu faire fonctionner ce projet également en temps réel, et afin de prévoir en temps réel cela peut prendre un certain temps, ce tableau représente la moyenne de temps pris pour chaque opération pendant le cycle de classification de Covid-19 :

Opération	Temps « S »
Détection d'objet	[0.00044 , 0.00083]
Balançage des couleurs	[0.026 , 0.028]
Détection des poumon	[2.26 , 2.43]
La classification	[0.1980 , 0.2181]
Temps total pure	[2.49 , 2.70]

**Tableau 3.3 :** le temps de chaque opération en temps réel

#### 5.4. Généralisation des résultats sur des bases externes

Après avoir terminé la phase d'entraînement et montré les résultats représentés dans les images graphiques et les courbes, et dans le but de vérifier la validité des résultats obtenus, nous avons testé ce modèle sur plusieurs bases de données les plus utilisées et les plus courantes, et nous avons obtenu de très bons résultats.

Tandis que, beaucoup d'images dans ces bases ne sont pas soumises aux conditions de de confirmation comme le PCR et la face frontale et les radiographies.

Cela indique que la plupart des mauvaises décisions de ce modèle sont dues à l'image utilisée qui ne remplit pas les conditions stipulées.

Ces tableaux nous montrent les résultats obtenus :

##### 5.4.1. Les tests avec la base de donnée COVID-19 Radiography Database [126]

Nous avons pris deux classes de cette base de données, la classe Covid contient 1212 photos de personnes infectées par la maladie Covid-19, et la seconde classe des personnes normales, qui contient 1380 photos Ces photos ne sont pas confirmées par la PCR.

les noms des classes	Covid		Normal	
	la total des image a la classe	1212		1380
la prédiction	Prédiction juste (covid)	Prédiction faux (normal)	Prédiction juste (normal)	Prédiction faux (covid)
le nombre des prédictions	1112	100	1365	15

**Tableau 3.4 :** Les tests avec la base de donnée COVID-19 Radiography Database

#### 5.4.2. Les tests avec la base de donnée COVID-19 chest xray [127]

Cette base contient un classe Covid contenant 357 photos de personnes atteintes de la maladie Covid-19 ces photos ne sont pas confirmées par PCR, nous avons pris ces photos et les avons testées avec notre modèle, et les résultats sont les suivants :

<b>les noms des classes</b>	<b>Covid</b>	
<b>la total des image a la classe</b>	357	
<b>la prédiction</b>	Prédiction juste (covid)	Prédiction faux (normal)
<b>le nombre des prédictions</b>	302	54

*Tableau 3.5 : Les tests avec la base de donnée COVID-19 chest xray*

#### 5.4.3. Les tests avec la base de donnée Covid-19 Image Dataset [128]

Cette base de données contient un total de 317 photos de personnes atteintes de maladies pulmonaires et de Covid et des photos thoraciques de personnes normales.

La classe de Maladies pulmonaires contient 90 images, et la classe Covid contient 137 images et la classe des Personnes normale en contient 90. Ces images n'ont pas non plus été confirmées par PCR.

Et nous avons pris les images des classes normal et covid car ils font l'objet de nos recherches et expérimentations pour les confirmer à l'aide de notre modèle, et les résultats étaient les suivants :

<b>les noms des classes</b>	<b>Covid</b>		<b>Normal</b>	
<b>la total des image a la classe</b>	137		90	
<b>la prédiction</b>	Prédiction juste (covid)	Prédiction faux (normal)	Prédiction juste (normal)	Prédiction faux (covid)
<b>le nombre des prédictions</b>	125	12	74	16

*Tableau 3.6 : Les tests avec la base de donnée Covid-19 Image Dataset*

#### 5.4.4. Les tests avec la base de donnée Chest Xray for covid-19 detection [134]

Cette base contient 371 images et est divisée en deux classes "Covid" et "Normal", et une partie de prédiction, chaque classe contient à peu près 174 images. Ces images ne sont pas non plus confirmées par PCR.

Nous avons pris des photos des deux classes et les avons testés à l'aide de notre modèle, et les résultats sont les suivants :

les noms des classes	Covid		Normal	
la total des image a la classe	143		169	
la prédiction	Prédiction juste (covid)	Prédiction faux (normal)	Prédiction juste (normal)	Prédiction faux (covid)
le nombre des prédictions	135	8	167	2

*Tableau 3.7 : Les tests avec la base de donnée Chest Xray for covid-19 detection*

#### 5.4.5. Les tests avec la base de donnée Covid Patients Chest X-Ray [138]

Dans le but de s'assurer davantage de la qualité du modèle, nous l'avons également testé sur la classe covid de cette base, il contient 162 images sur un total de 342. Ces images n'ont pas non plus été vérifiées par PCR.

Nous avons pris ces photos et les avons testées, et les résultats sont les suivants :

les noms des classes	Covid	
la total des image a la classe	162	
la prédiction	Prédiction juste (covid)	Prédiction faux (normal)
le nombre des prédictions	137	25

*Tableau 3.8 : Les tests avec la base de donnée Covid Patients Chest X-Ray*

## 6. Conclusion

À la fin de ce chapitre, nous avons vu la structure de notre réseau principal et toutes ses caractéristiques, paramètres, nombre de couches et les résultats obtenus lors de l'entraînement avec la nouvelle base de données entièrement traitée, en plus de toutes les étapes par lesquelles l'image est passée, depuis l'étape de recherche, de confirmation, de traitement et d'identification des zones requises à partir de détection pulmonaire avec un modèle de détection pré-entraîné déjà.

Et nous verrons dans le chapitre suivant la version finale de l'application qui utilise ces résultats obtenus.



# Chapitre 04

# L'implémentation

## 1. Introduction

Dans ce chapitre, et après avoir présenté dans le chapitre de la conception détaillée de notre application, nous expliquerons tout ce qui concerne la réalisation de cette application depuis l'environnement de développement et les bibliothèques jusqu'à l'explication de l'application et de toutes les fonctionnalités et résultats.

## 2. Environnement de développement et matériel

### 2.1. L'ordinateur et le système

- ✓ **Système d'exploitation** : Linux Mint 20 Cinnamon
- ✓ **Version de cinnamone** : 4.6.7
- ✓ **Kernel de linux** : 5.4.0-70-generic
- ✓ **Processeur** : Intel© Core™ i5-2520M CPU @ 2.50GHz × 2
- ✓ **Mémoire** : 4 gb
- ✓ **Carte graphique** : Intel Corporation 2nd Generation Core Processor Family Integrated Graphics Controller

### 2.2. Google colab

Google Collaboratory ou Colab, un outil Google simple et gratuit pour vous initier au deep Learning Permet d'écrire et d'exécuter du code en Python, Importation et l'enregistrement des blocs-notes depuis et vers Google Drive, basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud [213] [214] [215].

Google Collab permet également :

- ✓ D'améliorer vos compétences de codage en langage de programmation Python.
- ✓ De développer des applications en Deep Learning en utilisant des bibliothèques Python populaires telles que Keras, TensorFlow, PyTorch et OpenCV.
- ✓ D'utiliser un environnement de développement (Jupyter Notebook) qui ne nécessite aucune configuration.

Mais la fonctionnalité qui distingue Colab des autres services est l'accès à un processeur graphique GPU, totalement gratuitement.

### 2.2.1. Les types des GPU disponibles dans Google colab

Google fournit de nombreux GPUs pour de meilleures performances avec un accès gratuit, en particulier dans l'entraînement des modèles et parmi ces GPUs K80, T4, P4 et P100 de Nvidia [216].

Quelques-unes des bonnes choses que ce service offre :

- ✓ Accélérez les tâches de calcul telles que le machine learning et le HPC
- ✓ Un large choix de GPU adapté à une variété de besoins en termes de performances et de budgets
- ✓ Une tarification flexible et des personnalisations de machines en fonction de votre charge de travail

### 2.3. Pycharm

PyCharm est un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django. Développé par l'entreprise JetBrains, c'est un logiciel multiplateforme qui fonctionne sous Windows, Mac OS X et GNU/Linux. Il est décliné en édition professionnelle, diffusé sous licence propriétaire, et en édition communautaire diffusée sous licence Apache [217].

### 2.4. Qt Designer

Qt Designer est l'outil Qt permettant de concevoir et de réaliser des interfaces graphiques utilisateur (GUI) avec des widgets Qt. Vous pouvez composer et personnaliser vos fenêtres ou boîtes de dialogue à la manière de ce que vous voyez si vous obtenez a « what-you-see-is-what-you-get » (WYSIWYG), et les tester en utilisant différents styles et résolutions.

Les widgets et les formulaires créés avec Qt Designer s'intègrent parfaitement au code programmé, en utilisant le mécanisme de signaux et d'emplacements de Qt, de sorte que vous pouvez facilement attribuer un comportement aux éléments graphiques. Toutes les propriétés définies dans Qt Designer peuvent être modifiées dynamiquement dans le code. De plus, des fonctionnalités telles que la promotion de widgets et les plugins personnalisés vous permettent d'utiliser vos propres composants avec Qt Designer [218].

### **3. Langage de programmation, Framework et bibliothèques.**

#### **3.1. Python**

Python est le langage de programmation le plus populaire actuellement et dernières années parmi les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [219].

#### **3.2. TensorFlow**

TensorFlow est un Framework open source développé par les chercheurs de Google destiné pour l'implémentation des algorithmes d'apprentissage automatique l'apprentissage en profondeur et d'autres charges de travail d'analyse statistique et prédictive elle offre aussi une grande flexibilité dans le cadre de l'utilisation pour le développement d'un réseau CNN, alors nous avons utilisé cette bibliothèque pour définir les composants de base de notre l'architecture [220].

#### **3.3. Keras**

Keras est une bibliothèque d'apprentissage en profondeur écrite en Python, s'exécutant sur la plate-forme d'apprentissage automatique TensorFlow. Il a été développé dans le but de permettre une expérimentation rapide. Pouvoir passer de l'idée au résultat le plus rapidement possible est essentiel pour faire de bonnes recherches [209] [221].

Par conséquent, nous avons utilisé cette bibliothèque pour construire le réseau avec la plupart de ses couches comme la convolution, max pooling, flatting, dense etc ... et aussi dans le téléchargement et le l'enregistrement des données et l'affichage des résultats.

#### **3.4. NumPy**

NumPy est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que des tableaux et des matrices masqués) et un assortiment de routines pour des opérations rapides sur des tableaux, notamment mathématiques, logiques, manipulation de

forme, tri, sélection, nous avons utilisé cette bibliothèque pour adapter le type d'entrée et aussi pour la manipulation des tableaux [222].

### **3.5. OpenCv et PIL**

Nous avons utilisé les bibliothèques OpenCv et PIL pour des choses liées au traitement d'images comme l'utilisation de filtres, balançage des couleurs, etc., ainsi que pour la lecture et le stockage d'images [223] [224].

### **3.6. Matplotlib**

Matplotlib est une bibliothèque complète permettant de créer des visualisations statiques, animées et interactives en Python tandis que nous utilisons cette bibliothèque pour visualiser tous les graphiques donnés et les statistiques sous forme de graphiques [225].

### **3.7. Scikit-learn**

Nous avons utilisé cette bibliothèque pour effectuer les calculs nécessaires à la visualisation de certains graphiques et statistiques tels que la matrice de confusion et la fonction d'efficacité du récepteur ainsi que les arguments d'apprentissage du modèle [226].

### **3.8. FPDF**

Nous avons utilisé cette bibliothèque pour enregistrer les informations nécessaires dans un PDF. L'outil FPDF est une classe PHP permet de générer des fichiers PDF avec du PHP pur, c'est-à-dire sans utiliser la bibliothèque PDFlib. Il peut également l'utiliser pour tout type d'utilisation et la modification selon les besoins [227].

## **4. Présentation d'application**

Ce système est une application pour la classification d'images médicales du thorax de la maladie de Covid-19.

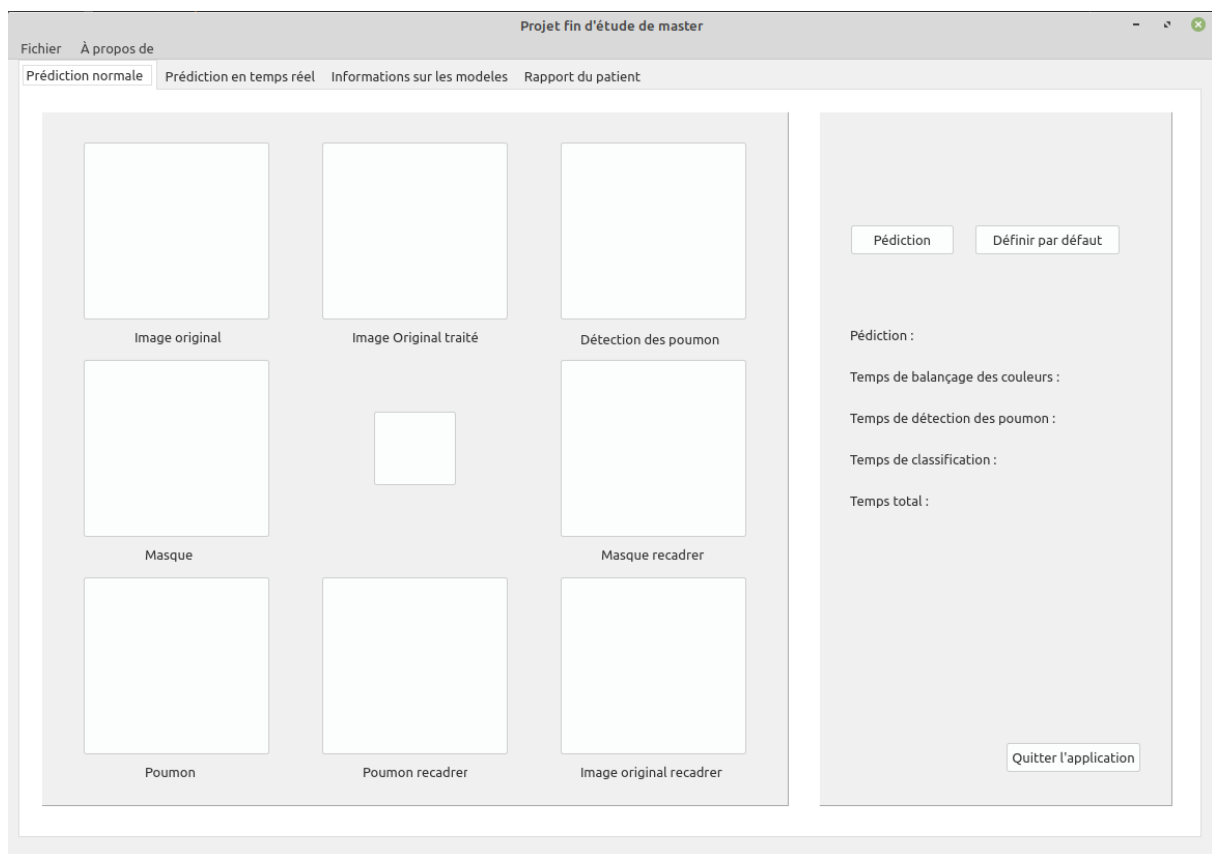
L'un des avantages de ce système est qu'il présente tous les choses à l'utilisateur, de la lecture de l'image au traitement de ses couleurs, en passant par la détection des poumons et la suppression des zones inutiles, jusqu'à la détermination de l'emplacement du vrai poumon et la prédiction d'image avec les temps de chaque opération.

Ce système fonctionne à deux niveaux, le premier niveau consiste à prendre une photo de l'ordinateur et à établir ce processus sur celle-ci, et le second niveau consiste à travailler en

temps réel, c'est-à-dire que dès que la caméra identifie une radiographie thoracique le système capture automatiquement et effectue le processus de traitement et de classification.

En plus de tout cela, et comme nous l'avons dit précédemment, ce système présente tous les paramètres du réseau neuronal convolutif, y compris les courbes, les statistiques et toutes les choses qui nous permettent de connaître la force et la précision du réseau, et surtout, il nous permet de voir l'image des différentes couches à l'intérieur du réseau convolutif et lorsque les filtres des couches convolutif lui sont appliqués.

Comme un autre avantage, et pour la raison que ce système peut être utilisé dans l'hôpital, il a l'avantage d'écrire la prédiction de la classification et toutes les informations sur le patient et l'image médicale thoracique et les notes sur un fichier avec PDF, et aussi il peut être imprimé.



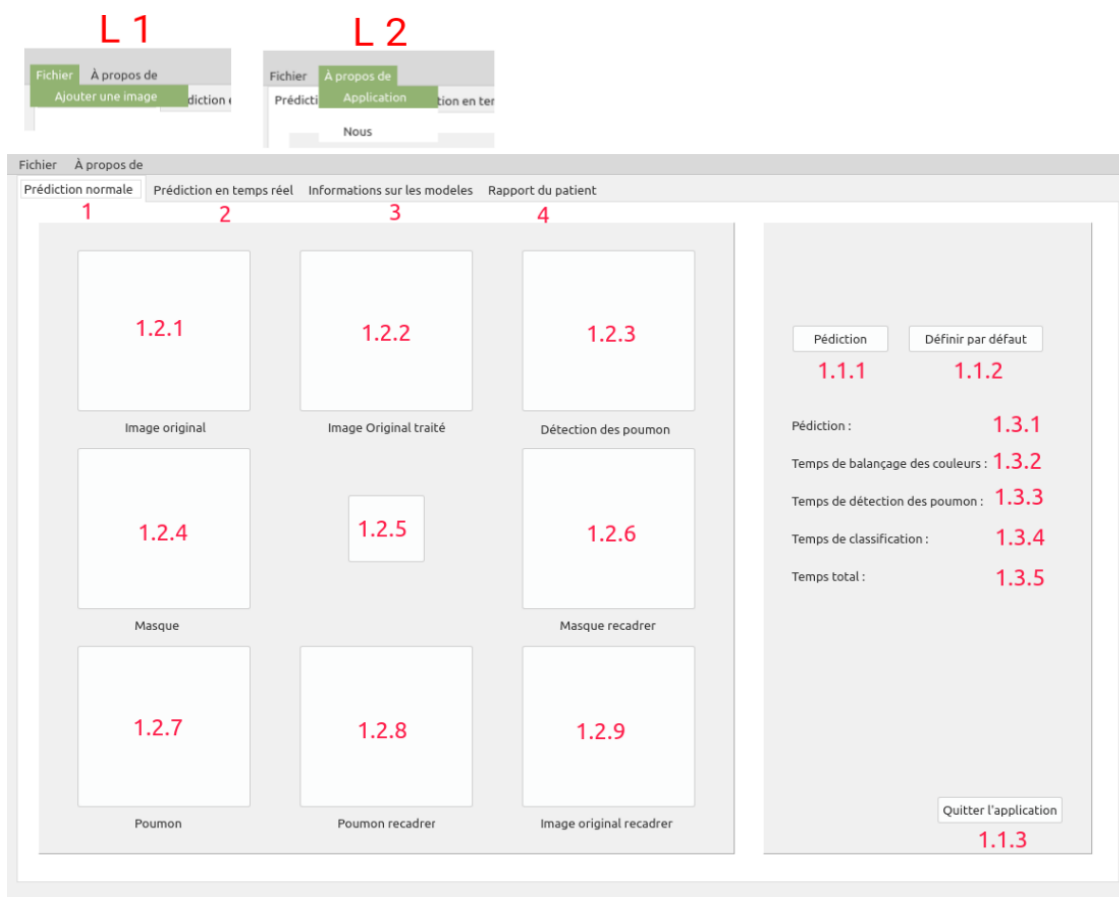
*Figure 4.1 : L'interface principale de l'application*

## 5. Présentation détaillée d'application

Cette application contient quatre pages, chacune d'elles a une fonction spéciale et nous allons maintenant présenter et expliquer les fonctions et caractéristiques de ces pages.

## 5.1. La première page

Cette page contient les fonctions qui effectuent la prédiction de classification pour Covid et présenter toutes les étapes par lesquelles l'image passe jusqu'à la fin, également le temps de chaque opération.



*Figure 4.2 : La première page d'application*

**L1** : liste contient l'action d'ajouter une image à partir de l'ordinateur.

**L2** : liste contient l'action d'à propos de l'application et du travail d'équipe.

**1** : la première page

**1.1.1** : début de prédiction.

**1.1.2** : définir toutes les interfaces textuelles et graphiques par défaut.

**1.1.3** : quitter l'application.

**1.2.1** : visualisation d'image originale, et si vous cliquez sur l'image, elle apparaîtra dans une grande image et vous pourrez zoomer et faire toutes les fonctions de l'image. Cette fonction est disponible pour toutes les images présentées sur l'application.

**1.2.2** : visualisation d'image traitée.

**1.2.3** : visualisation d'image traitée avec la détection des poumons.

**1.2.4** : visualisation du masque des poumons.

**1.2.5** : résultat prédit par couleur, si Covid est coloré en rouge, mais s'il est normal, il sera coloré en vert

**1.2.6** : visualisation de masque des poumons avec les espaces vides effacés.

**1.2.7** : visualisation des poumons.

**1.2.8** : visualisation des poumons avec les espaces vides effacés.

**1.2.9** : visualisation de l'image finale prête pour la prédiction.

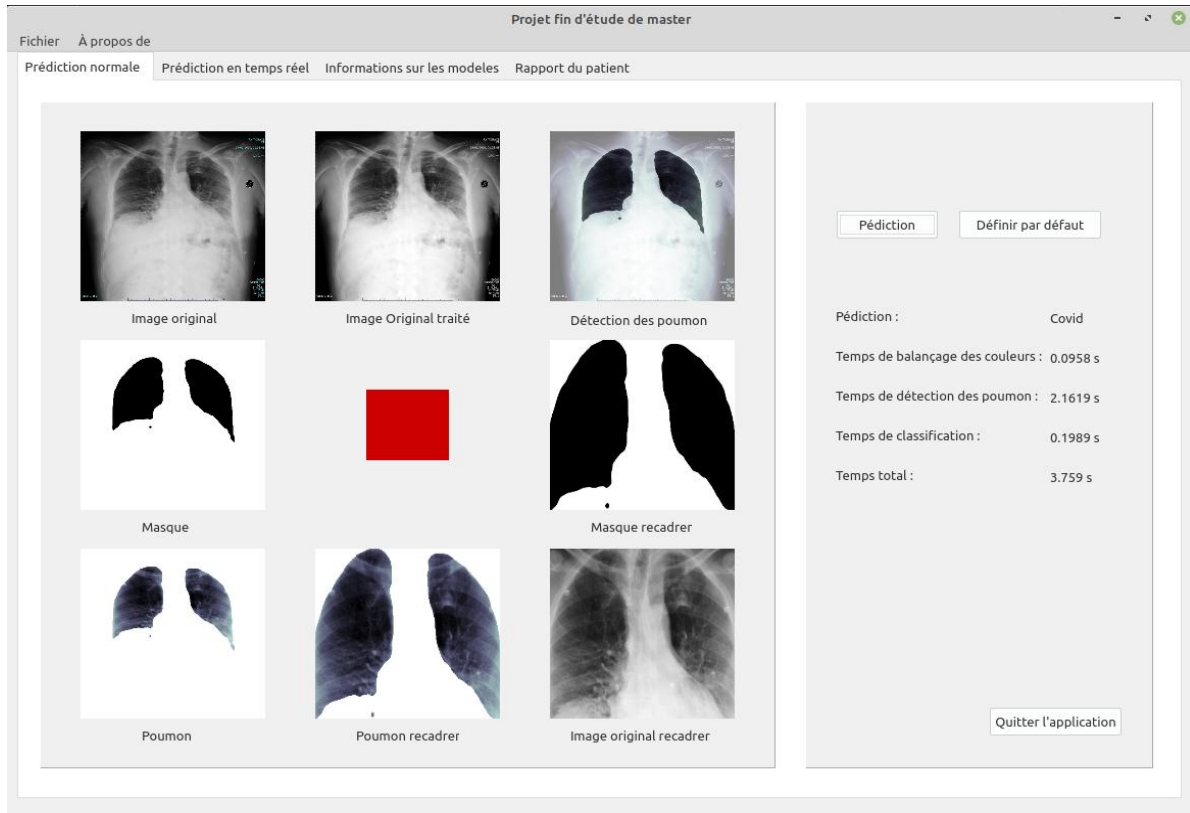
**1.3.1** : l'affichage de résultat de prédiction.

**1.3.2** : l'affichage de temps de traitement de couleur de l'image.

**1.3.3** : l'affichage de temps de détection des poumons.

**1.3.4** : l'affichage de temps de classification covid ou normal.

**1.3.5** : l'affichage de temps total d'exécution de processus.

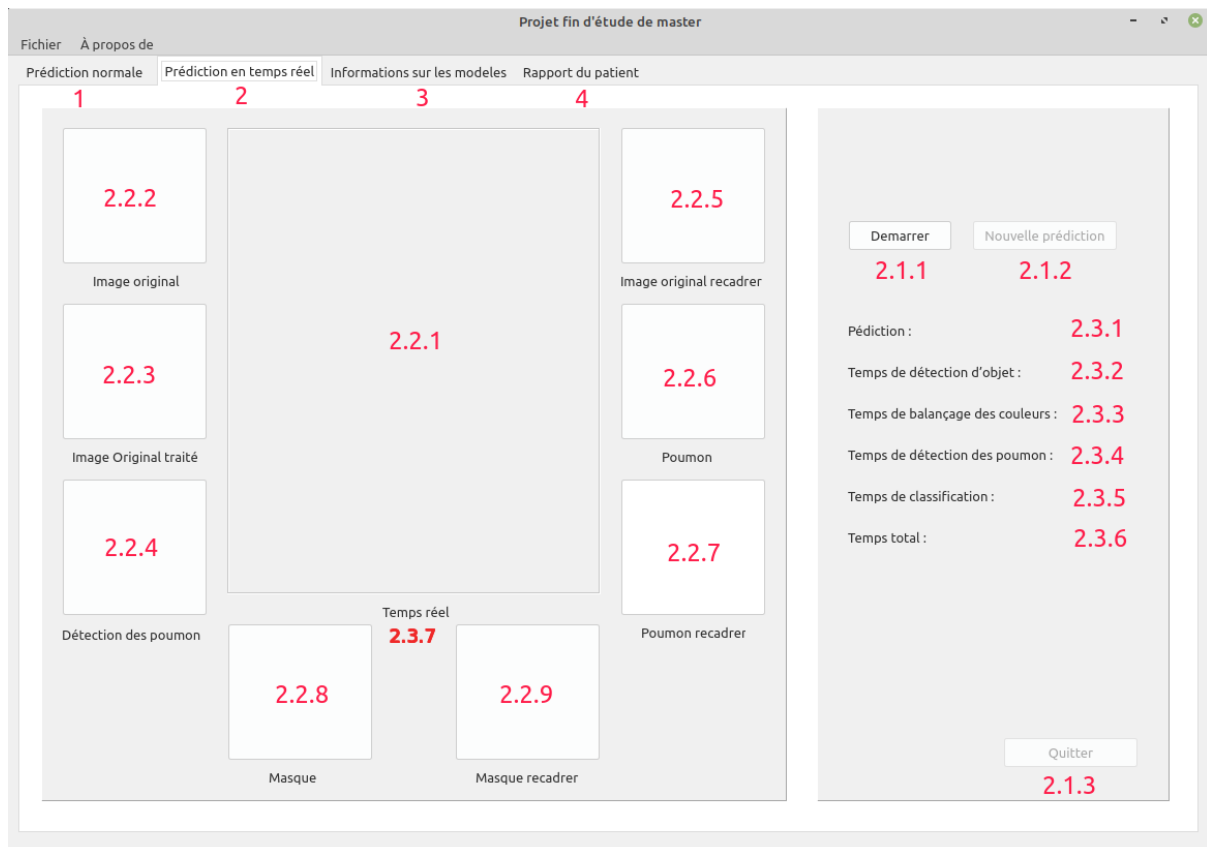


**Figure 4.3** : La première page d'application pendant l'exécution



## 5.2. La deuxième page

La plupart des fonctions de cette page sont identiques à celles de la première page, mais la différence est qu'ici nous pouvons travailler en temps réel, c'est-à-dire qu'au lieu de prendre la photo depuis l'ordinateur, nous prenons depuis caméra qui prend automatiquement l'image.



**Figure 4.4 :** La deuxième page d'application

**2 :** la deuxième page

**2.1.1 :** démarrer la caméra.

**2.1.2 :** faire une nouvelle prédiction après la fin de la première.

**2.1.3 :** quitter la prédiction en temps réel et fermer la caméra.

**2.2.1 :** streaming vidéo en temps réel.

**2.2.2 :** visualisation d'images originales capturées à partir de la caméra.

**2.2.3 :** visualisation d'image traitée.

**2.2.4 :** La détection des poumons.

**2.2.5** : visualisation de l'image finale prête pour la prédiction.

**2.2.6** : les poumons.

**2.2.7** : visualisation des poumons avec les espaces vides effacés.

**2.2.8** : le masque des poumons.

**2.2.9** : visualisation de masque des poumons avec les espaces vides effacés.

**2.3.1** : l'affichage de résultat de prédiction.

**2.3.2** : l'affichage de temps de détection d'objet (image thoracique) a partir la caméra.

**2.3.3** : l'affichage de temps de traitement de couleur de l'image.

**2.3.4** : l'affichage de temps de détection des poumons.

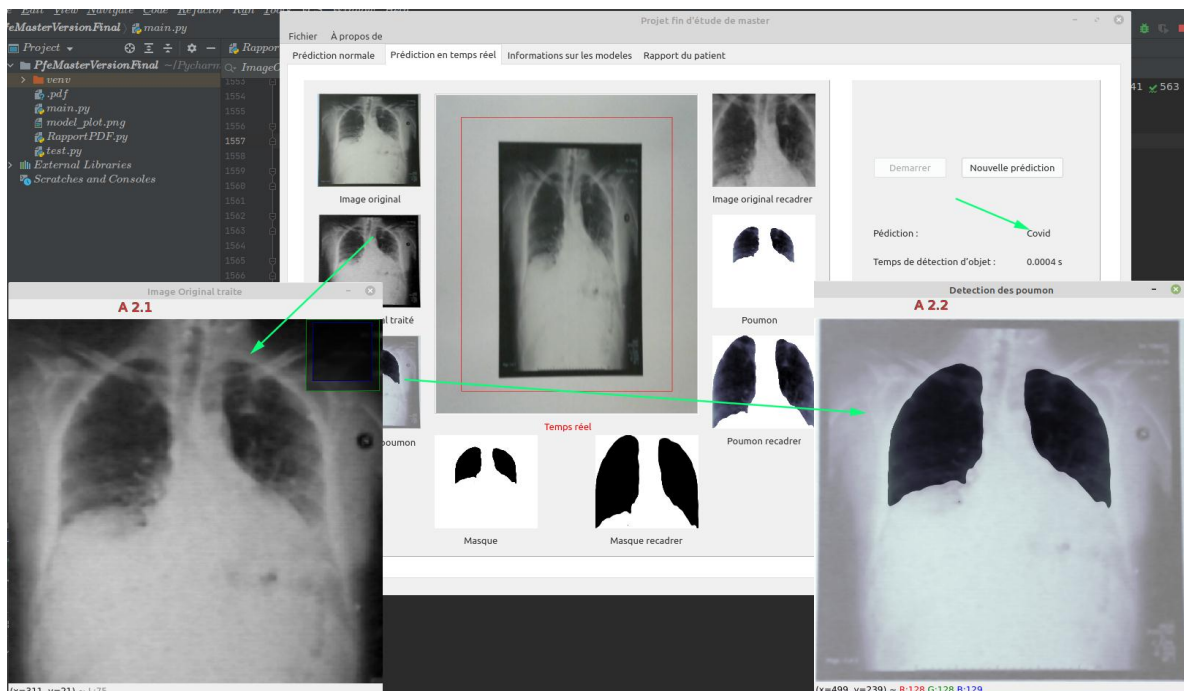
**2.3.5** : l'affichage de temps de classification covid ou normal.

**2.3.6** : l'affichage de temps total d'exécution de processus.

**2.3.7** : La couleur du mot Temps réel symbolise l'état de l'application. S'il est vert, il est prêt de détection. S'il est rouge, il est dans un état non fonctionnel.

**A 2.1** : L'image après avoir cliqué sur l'endroit où apparaît l'image originale " image traitée "

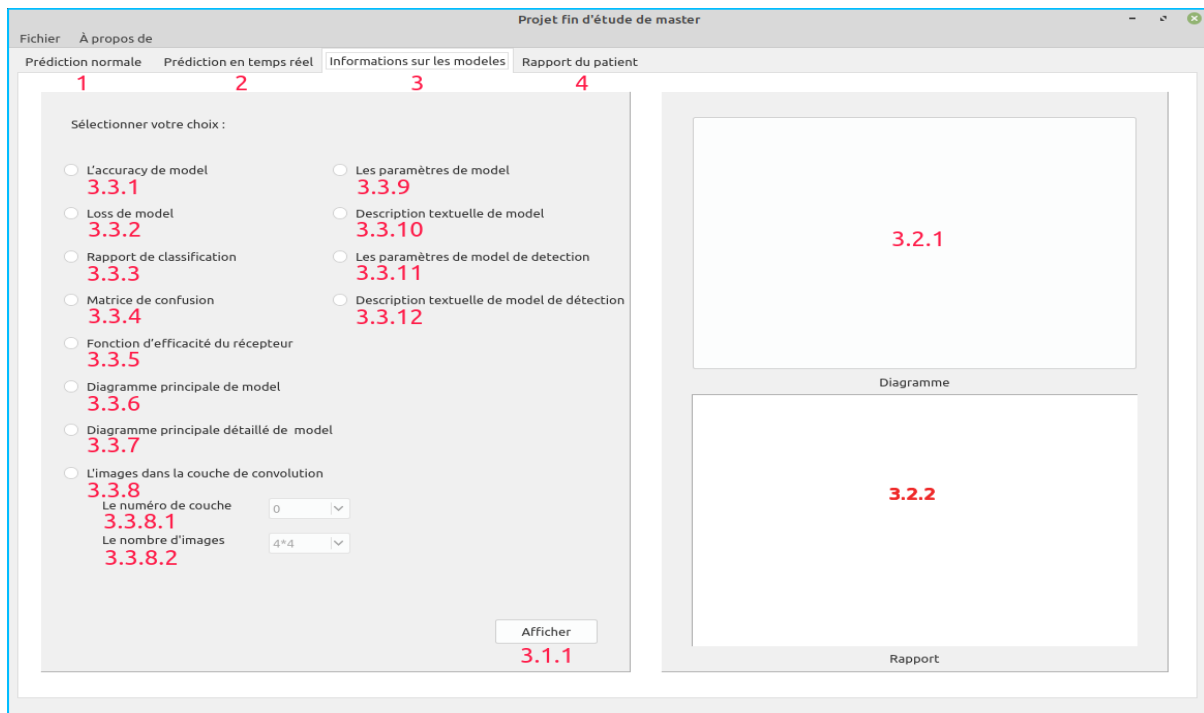
**A 2.2** : L'image après avoir cliqué sur l'endroit où apparaît l'image originale " image de détection des poumons "



*Figure 4.5 : La deuxième page d'application pendant l'exécution*

### 5.3. La troisième page

Les fonctions de cette page sont de présenter toutes les graphes, les statistiques, les descriptions textuelles et les illustrations des réseaux neuronaux convolutifs utilisés dans ce système. On peut aussi voir l'image à l'intérieur des couches convolutionnelles.



**Figure 4.6 :** La troisième page d'application

**3 :** la troisième page

**3.1.1 :** le bouton pour valider le choix sélectionné

**3.2.1 :** l'affichage des éléments graphiques.

**3.2.2 :** l'affichage des éléments textuels.

**3.3.1 :** le choix pour afficher l'accuracy de model.

**3.3.2 :** le choix pour afficher loss de model.

**3.3.3 :** le choix pour afficher le rapport de classification.

**3.3.4 :** le choix pour afficher la matrice de convolution.

**3.3.5 :** le choix pour afficher la fonction d'efficacité du récepteur.

**3.3.6 :** le choix pour afficher le diagramme principal de modèle.

**3.3.7 :** le choix pour afficher le diagramme principal détaillé de modèle.

**3.3.8 :** Le choix pour afficher l'image dans la couche de convolution.

**3.3.8.1 :** le numéro de couche.

**3.3.8.2 :** le nombre des images.

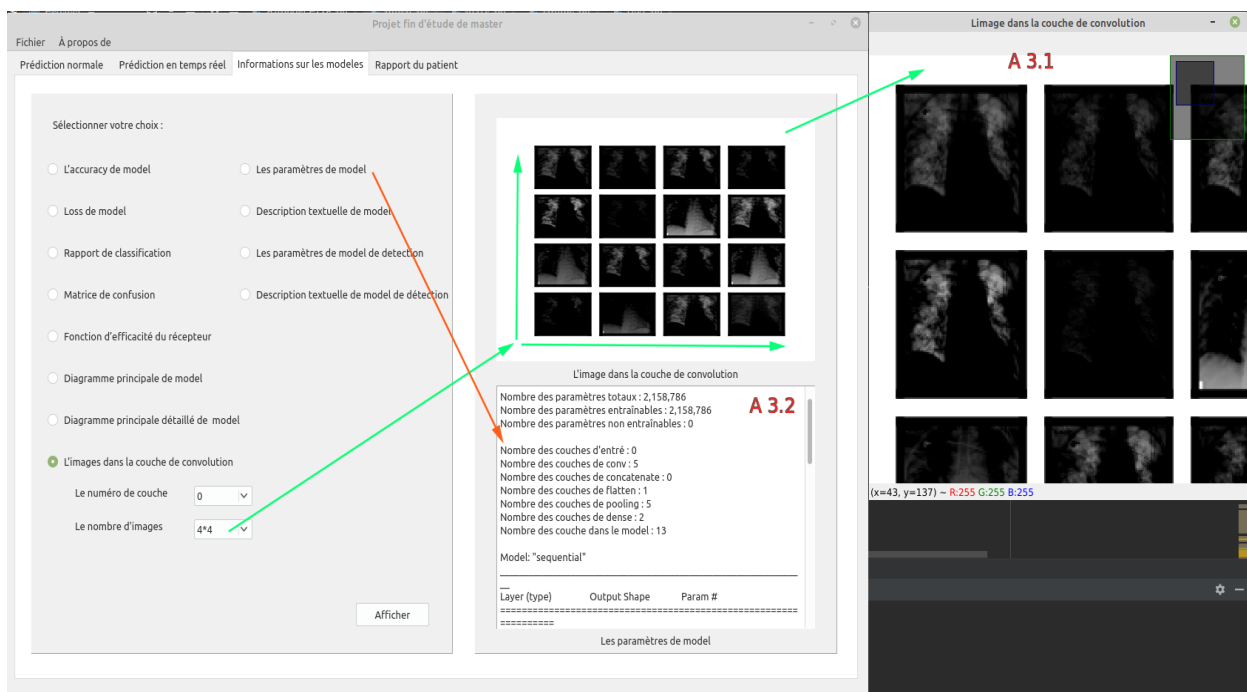
**3.3.9 :** Le choix pour afficher les paramètres de modèles de classification.

**3.3.10 :** Le choix pour afficher la description textuelle de modèles de classification.

**3.3.11 :** Le choix pour afficher les paramètres de modèles de détection.

**3.3.12 :** Le choix pour afficher la description textuelle de modèles de détection.

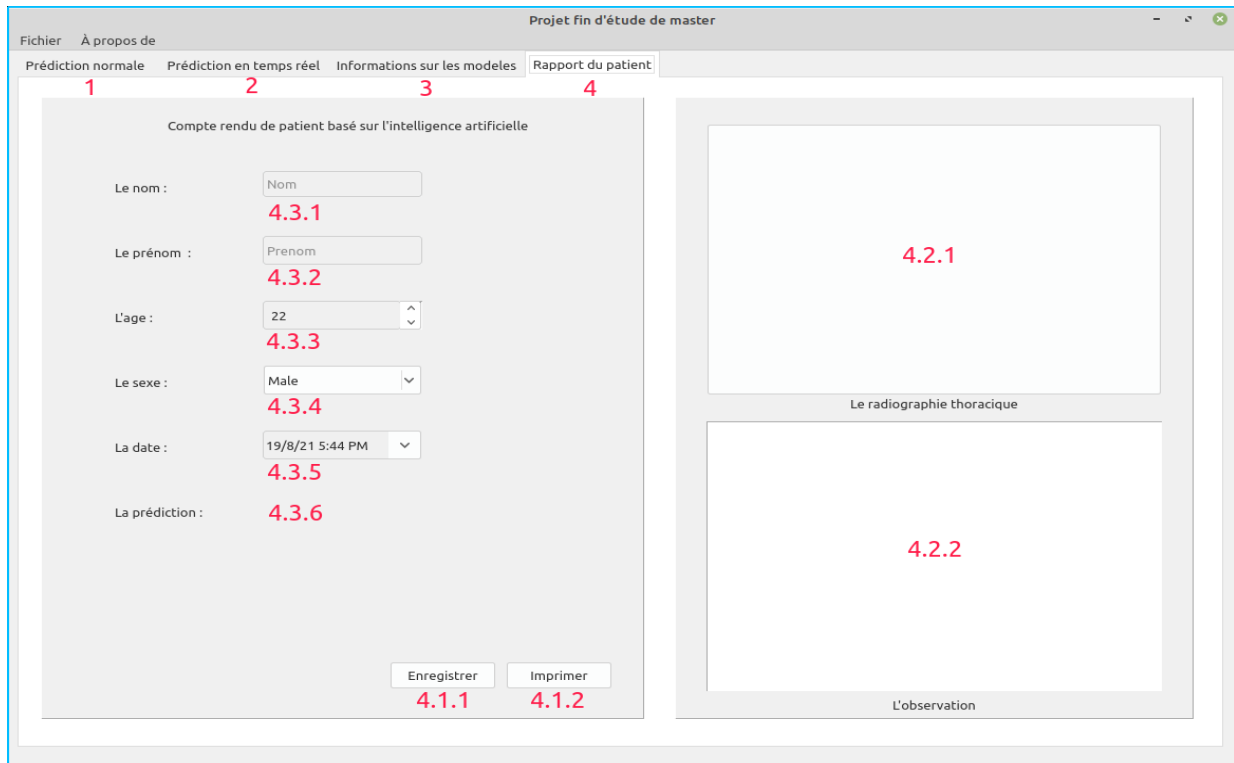
**A 3.1 :** L'image après avoir cliqué sur l'endroit où apparaît l'image originale “ le choix sélectionné ”.



*Figure 4.7 : La troisième page d'application pendant l'exécution*

## 5.4. La quatrième page

La fonction de cette page est d'écrire et de modifier les informations du patient, y compris le nom, le prénom, les informations privées, la note du médecin au résultat de la prédiction, et l'image médicale thoracique finale sur le fichier PDF, et elle peut également être imprimée.



*Figure 4.8 : La quatrième page d'application*

**4 :** la quatrième page

**4.1.1 :** le bouton pour enregistrer le fichier PDF.

**4.1.2 :** le bouton pour imprimer le fichier PDF.

**4.2.1 :** l'espace pour afficher image thoracique finale.

**4.2.2 :** l'espace d'observation de médecin.

**4.3.1 :** le nom du malade.

**4.3.2 :** le prénom du malade.

**4.3.3 :** l'âge de malade.

**4.3.4 :** le sexe de malade.

**4.3.5 :** la date de consultation.

**4.3.6 :** le résultat de prédiction.

Projet fin d'étude de master

Fichier À propos de

Prédiction normale Prédiction en temps réel Informations sur les modeles Rapport du patient

Compte rendu de patient basé sur l'intelligence artificielle

Le nom : Exemple d'un nom

Le prénom : Exemple d'un prenom

L'age : 23

Le sexe : Male

La date : 20/8/21 1:25 AM

La prédiction : Covid

Enregistrer Imprimer

Le radiographie thoracique

En l'absence de disponibilité de test biologique rapide, la réalisation d'un scanner thoracique pour le dépistage de lésions pulmonaires silencieuses chez des patients de statut COVID non connu, peut être recevable en cas d'urgences (ne permettant pas d'attendre les résultats de la PCR) pour une autre pathologie, telles que :  
Interventions chirurgicales en urgence : ORL, oncologie, etc..  
Situations thérapeutiques urgentes : AVC, situation hémorragique, etc..

L'observation

*Figure 4.9 : La quatrième page d'application pendant l'exécution*

**Compte rendu de patient basé sur  
l'intelligence artificielle**

Le : 20/8/21 11:06 AM

**Le nom :** Exemple d'un nom      **Le prenom :** Exemple d'un prenom

**L'age :** 23      **Le sexe :** Male

**La prédiction :** Covid

**Le radiographie thoracique médicale du patient :**



**L'observation :** En l'absence de disponibilité de test biologique rapide, la réalisation d'un image x ray thoracique pour le dépistage de lésions pulmonaires silencieuses chez des patients de statut COVID non connu, peut être recevable en cas d'urgences (ne permettant pas d'attendre les résultats de la PCR) pour une autre pathologie, telles que :  
Interventions chirurgicales en urgence : ORL, oncologie, etc..  
Situations thérapeutiques urgentes : AVC, situation hémorragique, etc..

**Signature**

Page 1/1

*Figure 4.10 : Exemple de compte rendu de malade*

## 6. Les points forts de notre système

- ✓ Prédiction de haute qualité parce qu'il est entraîné avec un base de données bien traité et confirmé avec 3 facteurs PCR, face frontale et xray.
- ✓ Prédiction en temps réel.
- ✓ Maintien de la qualité et de la valeur des informations dans l'image grâce aux grandes dimensions de l'image 512 \* 512 pixels.
- ✓ Utiliser uniquement la zone qui contient le poumon dans la formation et les tests, et ne pas faire attention et supprimer les parties non importantes de l'image.
- ✓ Améliorez la qualité de l'image pour afficher des détails fins grâce à l'algorithme de balance des couleurs.
- ✓ Quant à l'utilisation, il contient une interface graphique simple et facile qui présente toutes les étapes par lesquelles passe l'image et toutes les caractéristiques des modèles utilisés dans ce système.

## 7. Les points faibles de notre système

- ✓ Malgré la qualité de la base de données utilisée dans l'entraînement, elle est encore assez petite, en raison du manque de données confirmées.

## 8. Les difficultés que nous avons rencontrées

- ✓ Le manque de coopération de la Direction de la Santé et de la Population et de la plupart des hôpitaux pour fournir des clichés médicaux des patients Covid à cause du secret médical
- ✓ La plupart des appareils d'imagerie médicale avec lesquels nous avons pu travailler pendant notre période de stage n'ont pas la capacité de copier des images.
- ✓ Difficulté à trouver une base de données pour le Covid confirmé par PCR
- ✓ Les ressources de l'appareil ne sont pas assez puissantes pour fonctionner facilement : GPU, Ram ...
- ✓ Temps limité pour utiliser le processeur graphique sur Google.



## 9. Les perspectives pour le développement du système

- ✓ Recherche des bases de données plus volumineuses, et il est préférable qu'elles proviennent du même appareil sur lequel le système fonctionnera à l'avenir.
- ✓ La collection d'une grande base de données spécialement pour l'état de l'Algérie.
- ✓ Installation du système sur Raspberry Pi. Ou développement d'une version mobile.
- ✓ Dans le cas de bases de données volumineuses, il est recommandé d'utiliser l'apprentissage par transfert par d'autres réseaux VGG-16 VGG-19, inceptionV3.
- ✓ Prendre en considération les autres maladies pulmonaires.
- ✓ Utilisation de plus de ressources pour travailler plus facilement GPU, Ram, SSD drive...

## 10. Conclusion

A la fin de ce chapitre, nous avons pris connaissance de la plupart des outils utilisés pour mener à bien ce projet, des outils logiciels et des moyens physiques, en plus de cela nous avons pris connaissance de tous les avantages et caractéristiques de cette application en termes de fonctionnement interne. Structure du réseau de neurones utilisé et ce qui se passe à l'intérieur pour afficher le résultat et toutes les données, passant à la possibilité d'utiliser cette application dans la vie réelle.

# Conclusion générale

## Conclusion générale

La classification des maladies pulmonaires en général par l'intelligence artificielle et l'apprentissage profond est considérée comme une chose indispensable actuellement en raison du grand développement technologique dans tous les domaines, notamment le domaine médical, et de la disponibilité des moyens et équipements nécessaires pour le faire. La plupart des chercheurs et scientifiques cherchent à imposer leurs études et recherches sur le terrain en raison des nombreux défis scientifiques.

Dans notre recherche, nous avons synthétiser les études et les travaux liés à la classification des poumons qui dépendent des images pulmonaires afin d'avoir une meilleure vue et d'obtenir de meilleurs résultats dans ce domaine.

Cette classification dépend du réseau de neurones convolutifs, qui joue un rôle important dans la compréhension du contenu de l'image et dans les tâches liées à la segmentation, la classification, la détection et l'extraction d'objets.

Les difficultés que nous avons rencontrées dans le domaine d'image médicale sont la convergence et la similarité des informations dans l'image en raison de la qualité des moyens et appareils conçus pour la capture, et à cause des conditions d'environnement et des produits chimiques utilisés et d'autres.

Ce projet peut être approuvé au niveau des hôpitaux en raison des données d'entraînement confirmées par le PCR.

Enfin, il sera intéressant dans le futur de développer ce système et de pouvoir détecter des zones au niveau du poumon en développant un autre réseau CNN conçu pour cela.

# Bibliographie

- [1] Anatomie et physiologie <https://www.chuv.ch/fr/transplantation/cto-home/patients-et-familles/poumon/anatomie-et-physiologie> 04/04/2021
- [2] Jaouad Elkhalladi , Techniques radiologiques Viscérales , cours 3em semestre 2014/2015 Institut Supérieur des Professions Infirmières et Techniques de Santé de agadir
- [3] Wagner, Robert R. and Krug, Robert M. "Virus". Encyclopedia Britannica, 12 Nov. 2020, <https://www.britannica.com/science/virus>. Accessed 05 april 2021.
- [4] Structure des virus, [www.techno-science.net](http://www.techno-science.net) 06/04/2021
- [5] sante/epidemie , [www.doctissimo.fr](http://www.doctissimo.fr) 06/04/2021
- [6] Conférence du Pr P. Sansonetti au Collège de France [www.college-de-france.fr](http://www.college-de-france.fr)06/04/2021
- [7] Coronavirus pandemic [www.who.int](http://www.who.int) 13/04/2021
- [8] Jaimes JA et al., Phylogenetic Analysis and Structural Modeling of SARS-CoV-2 Spike Protein Reveals an Evolutionary Distinct and Proteolytically Sensitive Activation Loop, Journal of molecular biology, doi: 10.1016/j.jmb.2020.04.009 , [www.inserm.fr](http://www.inserm.fr)
- [9] Fehr and Perlman Meth Mol Biol 1282 1 (2015)
- [10] Huang C., Wang Y., Li X., Ren L., Zhao J., Hu Y. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. Lancet. 2020;395:497–506.
- [11] « Encyclopédie Larousse en ligne - intelligence artificielle » [archive], sur Larousse.[http://archive.wikiwix.com/cache/index2.php?url=https%3A%2F%2Fwww.larousse.fr%2Fencyclopedie%2Fdivers%2Fintelligence\\_artificielle%2F187257](http://archive.wikiwix.com/cache/index2.php?url=https%3A%2F%2Fwww.larousse.fr%2Fencyclopedie%2Fdivers%2Fintelligence_artificielle%2F187257) 08/04/2021
- [12] (en) Andreas Kaplan et Michael Haenlein, Siri, Siri in my Hand, who's the Fairest in the Land? On the Interpretations, Illustrations and Implications of Artificial Intelligence [archive], Business Horizons, vol. 62, no 1, 2018.
- [13] Haidekker M.A. (2013) X-Ray Projection Imaging. In: Medical Imaging Technology. SpringerBriefs in Physics. Springer, New York, NY. [https://doi.org/10.1007/978-1-4614-7073-1\\_2](https://doi.org/10.1007/978-1-4614-7073-1_2)
- [14] Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.
- [15] Koza J.R., Bennett F.H., Andre D., Keane M.A. (1996) Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In: Gero J.S., Sudweeks F. (eds) Artificial Intelligence in Design '96. Springer, Dordrecht. <https://doi.org/10.1007/978-94-009-0279-4>
- [16] Russell, Stuart J.; Norvig, Peter (2010). Artificial Intelligence: A Modern Approach (Third ed.). Prentice Hall. ISBN 9780136042594.
- [17] Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). Foundations of Machine Learning. The MIT Press. ISBN 9780262018258.
- [18] Haidekker M.A. (2013) Trends in Medical Imaging Technology. In: Medical Imaging Technology. SpringerBriefs in Physics. Springer, New York, NY. [https://doi.org/10.1007/978-1-4614-7073-1\\_7](https://doi.org/10.1007/978-1-4614-7073-1_7)
- [19] Nicolas Limare, Jose-Luis Lisani, Jean-Michel Morel, Ana Bel\_en Petro,Catalina Sbert, 'Simplest Color Balance', Published in Image Processing ISSN 2105-1232 ©IPOL,2011. <http://dx.doi.org/10.5201/ipol.2011.llmps-scb>
- [20] Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker (ed.). Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, Florida: Chapman & Hall/CRC Press LLC. ISBN 978-1-58488-360-9.
- [21] van Otterlo, M.; Wiering, M. (2012). Reinforcement learning and markov decision processes. Reinforcement Learning. Adaptation, Learning, and Optimization. 12. pp. 3–42. doi:10.1007/978-3-642-27645-3\_1. ISBN 978-3-642-27644-6.
- [22] Deng, L.; Yu, D. (2014). "Deep Learning: Methods and Applications" (PDF). Foundations and Trends in Signal Processing. 7 (3–4): 1–199. doi:10.1561/20000000039. Archived (PDF) from the original on 2016-03-14.

- [23] Bengio, Y.; Courville, A.; Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 35 (8): 1798–1828. arXiv:1206.5538. doi:10.1109/tpami.2013.50. PMID 23787338. S2CID 393948.
- [24] Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. 61: 85–117. arXiv:1404.7828. doi: 10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.
- [25] Bengio, Yoshua; LeCun, Yann; Hinton, Geoffrey (2015). "Deep Learning". *Nature*. 521 (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442. S2CID 3074096.
- [26] Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F. (2020). "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning". *IEEE Transactions on Vehicular Technology*. 69(12): 14413–14423. doi:10.1109/TVT.2020.3034800. S2CID 228989788.
- [27] Ciresan, D.; Meier, U.; Schmidhuber, J. (2012). "Multi-column deep neural networks for image classification". 2012 *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3642–3649. arXiv:1202.2745. doi:10.1109/cvpr.2012.6248110. ISBN 978-1-4673-1228-8. S2CID 2161592.
- [28] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey (2012). "ImageNet Classification with Deep Convolutional Neural Networks" (PDF). *NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada*.
- [29] "Google's AlphaGo AI wins three-match series against the world's best Go player". *TechCrunch*. 25 May 2017.
- [30] neurone-biologique, [www.medecine-et-robotique.fr](http://www.medecine-et-robotique.fr) 28-04-2021
- [31] Neurone Biologique : Structure et rôle <https://datascientest.com/deep-learning-reseau-de-neurones-biologiques-ou-artificiels> 28-04-2021
- [32] Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K., (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4), pp.611-629.
- [33] Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate cortex. *J Physiol*. doi: 10.1113/jphysiol.1959.sp006308
- [34] Schmidhuber J (2007) New millennium AI and the convergence of history. In: *Challenges for computational intelligence*. Springer, pp 15–35
- [35] Joachims T (1998) Text categorization with support vector machines: Learning with many relevant features. In: *European conference on machine learning*. pp 137–142
- [36] THE NEURAL NETWORK ZOO <https://www.asimovinstitute.org/neural-network-zoo/> 01/10/2021
- [37] Asifullah K et al A Survey of the Recent Architectures of Deep Convolutional Neural Networks arXiv:1901.06032
- [38] Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18:1527–1554
- [39] Khan et al. 2018b : Khan A, Zameer A, Jamal T, Raza A (2018b) Deep Belief Networks Based Feature Generation and Regression for Predicting Wind Power. arXiv Prepr arXiv180711682
- [40] Gu J, Wang Z, Kuen J, et al (2018) Recent advances in convolutional neural networks. *Pattern Recognit* 77:354–377. doi: 10.1016/j.patcog.2017.10.013
- [41] Sinha T, Verma B, Haidar A (2018) Optimization of convolutional neural network parameters for image classification. 2017 *IEEE Symp Ser Comput Intell SSCI 2017 - Proc 2018-Janua*:1–7. doi: 10.1109/SSCI.2017.8285338
- [42] Zhang Q, Zhang M, Chen T, et al (2019) Recent advances in convolutional neural network acceleration. *Neurocomputing* 323:37–51. doi: 10.1016/j.neucom.2018.09.038
- [43] Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. *Adv Neural Inf Process Syst* 1–9. doi: 10.1061/(ASCE)GT.1943-5606.0001284
- [44] Szegedy C, Wei Liu, Yangqing Jia, et al (2015) Going deeper with convolutions. In: 2015 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp 1–9
- [45] Srivastava RK, Greff K, Schmidhuber J (2015a) Highway Networks. doi: 10.1002/esp.3417

- [46] He K, Zhang X, Ren S, Sun J (2015a) Deep Residual Learning for Image Recognition. *Multimed Tools Appl* 77:10437–10453. doi: 10.1007/s11042-017-4440-4
- [47] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. *Proc - 30th IEEE Conf Comput Vis Pattern Recognition, CVPR 2017 2017-Janua:2261–2269*. doi: 10.1109/CVPR.2017.243
- [48] Kafi M, Maleki M, Davoodian N (2015) Functional histology of the ovarian follicles as determined by follicular fluid concentrations of steroids and IGF-1 in *Camelus dromedarius*. *Res Vet Sci* 99:37–40. doi: 10.1016/j.rvsc.2015.01.001
- [49] Shin H-CC, Roth HR, Gao M, et al (2016) Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Trans Med Imaging* 35:1285–1298. doi: 10.1109/TMI.2016.2528162
- [50] Bengio Y (2013) Deep learning of representations: Looking forward. In: *International Conference on Statistical Language and Speech Processing*. Springer, pp 1–37
- [51] Montufar GF, Pascanu R, Cho K, Bengio Y (2014) On the number of linear regions of deep neural networks. In: *Advances in neural information processing systems*. pp 2924–2932
- [52] Csáji B (2001) Approximation with artificial neural networks. MSc thesis 45. doi: 10.1.1.101.2647
- [53] Delalleau O, Bengio Y (2011) Shallow vs. deep sum-product networks. In: *Advances in Neural Information Processing Systems*. pp 666–674
- [54] Wang H, Raj B (2017) On the Origin of Deep Learning. 1–72. doi: 10.1016/0014-5793(91)81229-2
- [55] Bengio Y (2013) Deep learning of representations: Looking forward. In: *International Conference on Statistical Language and Speech Processing*. Springer, pp 1–37
- [56] Nguyen Q, Mukkamala M, Hein M (2018) Neural Networks Should Be Wide Enough to Learn Disconnected Decision Regions. *arXiv Prepr arXiv180300094*
- [57] Simonyan K, Zisserman A (2015) VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. *ICLR* 75:398–406. doi: 10.2146/ajhp170251
- [58] Szegedy C, Vanhoucke V, Ioffe S, et al (2016b) Rethinking the Inception Architecture for Computer Vision. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, pp 2818–2826
- [59] Szegedy C, Wei Liu, Yangqing Jia, et al (2015) Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp 1–9
- [60] Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int J Uncertainty, Fuzziness Knowledge-Based Syst* 6:107–116
- [61] Dong C, Loy CC, He K, Tang X (2016) Image super-resolution using deep convolutional networks. *IEEE Trans Pattern Anal Mach Intell* 38:295–307
- [62] Pascanu R, Mikolov T, Bengio Y (2012) Understanding the exploding gradient problem. *CoRR*, abs/12115063
- [63] Dauphin YN, Fan A, Auli M, Grangier D (2017) Language modeling with gated convolutional networks. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. pp 933–941
- [64] Kuen J, Kong X, Wang G, Tan YP (2018) DelugeNets: Deep Networks with Efficient and Flexible Cross-Layer Information Inflows. *Proc - 2017 IEEE Int Conf Comput Vis Work ICCVW 2017 2018-Janua :958–966*. doi: 10.1109/ICCVW.2017.117
- [65] Larsson G, Maire M, Shakhnarovich G (2016) Fractalnet: Ultra-deep neural networks without residuals. *arXiv Prepr arXiv160507648* 1–11
- [66] Mao X, Shen C, Yang Y-B (2016) Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: *Advances in neural information processing systems*. pp 2802–2810
- [67] Tong T, Li G, Liu X, Gao Q (2017) Image super-resolution using dense skip connections. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Pp 4809–4817
- [68] Kawaguchi K, Huang J, Kaelbling LP (2019) Effect of depth and width on local minima in deep learning. *Neural Comput* 31:1462–1498. doi: 10.1162/neco\_a\_01195

- [69] Lu H, Li B, Zhu J, et al (2017a) Wound intensity correction and segmentation with convolutional neural networks. *Concurr Comput Pract Exp* 29 : e3927
- [70] Hanin B, Sellke M (2017) Approximating Continuous Functions by ReLU Nets of Minimal Width. *arXiv Prepr arXiv171011278*
- [71] Lu Z, Pu H, Wang F, et al (2017b) The expressive power of neural networks : A view from the width. In : *Advances in Neural Information Processing Systems*. pp 6231–6239
- [72] LeCun Y, Kavukcuoglu K, Farabet CC, others (2010) Convolutional networks and applications in vision. In : *ISCAS. IEEE*, pp 253–256
- [73] Hu J, Shen L, Sun G (2018a) Squeeze-and-Excitation Networks. In : *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE*, pp 7132–7141
- [74] Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60 :91–110
- [75] Dollár P, Tu Z, Perona P, Belongie S (2009) Integral channel features
- [76] Do MN, Vetterli M (2005) The contourlet transform: an efficient directional multiresolution image representation. *IEEE Trans image Process* 14 :2091–2106
- [77] Oquab M, Bottou L, Laptev I, Sivic J (2014) Learning and transferring mid-level image representations using convolutional neural networks. In : *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE*, pp 1717–1724
- [78] Yang J, Xiong W, Li S, Xu C (2019) Learning structured and non-redundant representations with deep neural networks. *Pattern Recognit* 86 :224–235
- [79] Mikolov T, Karafiát M, Burget L, et al (2010) Recurrent neural network based language model. In: *Eleventh Annual Conference of the International Speech Communication Association*
- [80] Sundermeyer M, Schlüter R, Ney H (2012) LSTM neural networks for language modeling. In: *Thirteenth annual conference of the international speech communication association*
- [81] LeCun Y, Jackel LD, Bottou L, et al (1995) Learning algorithms for classification : A comparison on handwritten digit recognition. *Neural networks Stat Mech Perspect* 261 :276
- [82] Zeiler MD, Fergus R (2013) Visualizing and Understanding Convolutional Networks. *arXiv Prepr arXiv13112901v3* 30:225–231. doi: 10.1111/j.1475-4932.1954.tb03086.x
- [83] Simonyan K, Zisserman A (2015) VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. *ICLR* 75 :398–406. doi: 10.2146/ajhp170251
- [84] Srivastava RK, Greff K, Schmidhuber J (2015a) Highway Networks. doi: 10.1002/esp.3417
- [85] Zagoruyko S, Komodakis N (2016) Wide Residual Networks. *Proceedings Br Mach Vis Conf 2016* 87.1-87.12. doi: 10.5244/C.30.87
- [86] Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. *arXiv Prepr* 1610–2357
- [87] Wang F, Jiang M, Qian C, et al (2017a) Residual Attention Network for Image Classification. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, pp 6450–6458
- [88] Xie S, Girshick R, Dollár P, et al (2017) Aggregated Residual Transformations for Deep Neural Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, pp 5987–5995
- [89] Zhang X, Li Z, Loy CC, Lin D (2017) PolyNet: A Pursuit of Structural Diversity in Very Deep Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, pp 3900–3908
- [90] Han D, Kim J, Kim J (2017) Deep Pyramidal Residual Networks. In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, pp 6307–6315
- [91] Woo S, Park J, Lee JY, Kweon IS (2018) CBAM : Convolutional block attention module. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 11211 LNCS :3–19. doi: 10.1007/978-3-030-01234-2\_1

- [92] Roy AG, Navab N, Wachinger C (2018) Concurrent spatial and channel ‘squeeze & excitation’ in fully convolutional networks. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 11070 LNCS :421–429. doi: 10.1007/978-3-030-00928-1\_48
- [93] Khan A, Sohail A, Ali A (2018a) A New Channel Boosted Convolutional Neural Network using Transfer Learning. arXiv Prepr arXiv180408528
- [94] Hu Y, Wen G, Luo M, et al (2018b) Competitive Inner-Imaging Squeeze and Excitation for Residual Network. doi: arXiv:1807.08920v3
- [95] Dupond, Samuel (2019). "A thorough review on the current advance of neural network structures". Annual Reviews in Control. 14 : 200–230.
- [96] Abiodun, Oludare & Jantan, Aman & Omolara, Oludare & Dada, Kemi & Mohamed, Nachaat & Arshad, Humaira. (2018). State-of-the-art in artificial neural network applications : A survey. Heliyon. 4. e00938. 10.1016/j.heliyon.2018.e00938.
- [97] Tealab, Ahmed (2018-12-01). "Time series forecasting using artificial neural networks methodologies: A systematic review". Future Computing and Informatics Journal. 3 (2): 334–340. doi: 10.1016/j.fcij.2018.10.003. ISSN 2314-7288.
- [98] Graves, Alex; Liwicki, Marcus; Fernandez, Santiago; Bertolami, Roman; Bunke, Horst; Schmidhuber, Jürgen (2009). "A Novel Connectionist System for Improved Unconstrained Handwriting Recognition" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 31 (5): 855–868. CiteSeerX 10.1.1.139.4502. doi:10.1109/tpami.2008.137. PMID 19299860. S2CID 14635907.
- [99] Sak, Haşim; Senior, Andrew; Beaufays, Françoise (2014). "Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling" (PDF).
- [100] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
- [101] SIIM-FISABIO-RSNA COVID-19 Detection <https://www.kaggle.com/c/siim-covid19-detection/overview/prizes> 15/06/2021
- [102] COVID-19 Hospital Stay Prediction <https://www.kaggle.com/c/1056lab-covid19-hospital-stay-prediction> 15/06/2021
- [103] Covid-19 Risk <https://www.kaggle.com/c/covid19-risk-mexico/overview/evaluation> 18/06/2021
- [104] ECE657AW21-Assignment4-COVID <https://www.kaggle.com/c/ece657aw21-assignment4-covid> 18/06/2021
- [105] Small Covid Data <https://www.kaggle.com/c/small-covid-data> 18/06/2021
- [106] COMS 4771 COVID Challenge <https://www.kaggle.com/c/4771-sp20-covid> 19/06/2021
- [107] COVID19 Local US-CA Forecasting (Week 1) <https://www.kaggle.com/c/covid19-local-us-ca-forecasting-week-1> 16/06/2021
- [108] COVID-19 detection from coughs <https://www.kaggle.com/c/covid> 19/06/2021
- [109] ML2021Spring-hw1 <https://www.kaggle.com/c/ml2021spring-hw1> 20/06/2021
- [110] COVID-19 CT Images Segmentation <https://www.kaggle.com/c/covid-segmentation> 20/06/2021
- [111] COVID-19 Hospital Stay Prediction <https://www.kaggle.com/c/1056lab-covid19-hospital-stay-prediction> 17/06/2021
- [112] COVID-19 Segmentation <https://www.kaggle.com/c/medical-imaging-2021> 20/06/2021
- [113] COVID-19 Cases Prediction <https://www.kaggle.com/c/covid-19-ml-it-kmitl-2020> 18/06/2021
- [114] COVID-19 X-ray image classification <https://www.kaggle.com/c/stat946winter2021> 18/06/2021
- [115] COVID-19 Chest x-ray challenge <https://www.kaggle.com/c/csc532> 18/06/2021
- [116] ProtonX - Covid 19 Classification <https://www.kaggle.com/c/protonx-tf01-covid19> 18/06/2021
- [117] COVID19 Retweet Prediction Challenge 2020 <https://www.kaggle.com/c/covid19-retweet-prediction-challenge-2020> 18/06/2021



- [118] CT Scan COVID Prediction <https://www.kaggle.com/c/covidct> 18/06/2021
- [119] Multi-class COVID-19 Chest x-ray challenge <https://www.kaggle.com/c/csc532-2> 19/06/2021
- [120] Covid-19 detection using transfer learning <https://www.kaggle.com/c/covid-19-detection-using-transfer-learning> 19/06/2021
- [121] Hack-D-Covid'20 Track-1 <https://www.kaggle.com/c/kjsce-hack-d-covid20-track1> 19/06/2021
- [122] Prediction of COVID-19 diagnosis based on symptoms <https://www.kaggle.com/c/prediction-of-covid-19> 19/06/2021
- [123] Diagnóstico de COVID19 según síntomas del paciente <https://www.kaggle.com/c/mt616covid19> 19/06/2021
- [124] AIAT Hackathon: COVID Image <https://www.kaggle.com/c/aiat-hackathon-covid-image> 21/06/2021
- [125] 2020\_UCLA\_CS145\_Covid19\_Prediction <https://www.kaggle.com/c/2020-ucla-cs145-covid19-prediction> -- 21/06/2021
- [126] COVID-19 Radiography Database <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database> 21/06/2021
- [127] COVID-19 chest xray <https://www.kaggle.com/bachrr/covid-chest-xray?select=images> 21/06/2021
- [128] Covid-19 Image Dataset <https://www.kaggle.com/pranavraikokte/covid19-image-dataset> 21/06/2021
- [129] COVIDx CXR-2 [https://www.kaggle.com/andyczhao/covidx-cxr2?select=competition\\_test](https://www.kaggle.com/andyczhao/covidx-cxr2?select=competition_test) 22/06/2021
- [130] COVID-19 Xray Dataset (Train & Test Sets) <https://www.kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets> 22/06/2021
- [131] COVID-19 Patients Lungs X Ray Images 10000 <https://www.kaggle.com/nabeelsajid917/covid-19-x-ray-10000-images> 22/06/2021
- [132] Chest X-ray (Covid-19 & Pneumonia) <https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia> 22/06/2021
- [133] COVID-19 Chest X-ray Image Dataset <https://www.kaggle.com/alifrahman/covid19-chest-xray-image-dataset> 22/06/2021
- [134] Chest Xray for covid-19 detection <https://www.kaggle.com/fusicfenta/chest-xray-for-covid19-detection> 23/06/2021
- [135] CoronaHack -Chest X-Ray-Dataset <https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset> 23/06/2021
- [136] COVID19\_Pneumonia\_Normal\_Chest\_Xray\_PA\_Dataset <https://www.kaggle.com/amanullahasraf/covid19-pneumonia-normal-chest-xray-pa-dataset> 23/06/2021
- [137] COVID-19 X rays <https://www.kaggle.com/andrewmvd/convid19-x-rays?select=X+rays> 23/06/2021
- [138] Covid Patients Chest X-Ray <https://www.kaggle.com/ankitachoudhury01/covid-patients-chest-xray> 23/06/2021
- [139] Covid-GAN and Covid-Net mini Chest X-ray <https://www.kaggle.com/yash612/covidnet-mini-and-gan-enerated-chest-xray> 23/06/2021
- [140] Chest X-Ray Images (Pneumonia) <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia> 25/06/2021
- [141] covid-chestxray-dataset <https://github.com/ieee8023/covid-chestxray-dataset> 25/06/2021
- [142] Ba, J.L., Kiros, J.R. and Hinton, G.E., 2016. Layer normalization. arXiv preprint arXiv:1607.06450.
- [143] Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
- [144] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

- [145] Ulyanov, D., Vedaldi, A. and Lempitsky, V., 2016. Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022.
- [146] Miyato, T., Kataoka, T., Koyama, M. and Yoshida, Y., 2018. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.
- [147] Ogasawara, E., Martinez, L.C., De Oliveira, D., Zimbrão, G., Pappa, G.L. and Mattoso, M., 2010, July. Adaptive normalization: a novel data normalization approach for non-stationary time series. In The 2010 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
- [148] Salimans, T. and Kingma, D.P., 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In Advances in neural information processing systems (pp. 901-909).
- [149] De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O. and Courville, A.C., 2017. Modulating early visual processing by language. In Advances in Neural Information Processing Systems (pp. 6594-6604).
- [150] Wu, Y. and He, K., 2018. Group normalization. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).
- [151] Kingma, D.P. and Dhariwal, P., 2018. Glow: Generative flow with invertible 1x1 convolutions. In Advances in neural information processing systems (pp. 10215-10224).
- [152] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. and Aila, T., 2020. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8110-8119).
- [153] Luo, P., Ren, J., Peng, Z., Zhang, R. and Li, J., 2018. Differentiable learning-to-normalize via switchable normalization. arXiv preprint arXiv:1806.10779.
- [154] Foracchia, M., Grisan, E. and Ruggeri, A., 2005. Luminosity and contrast normalization in retinal images. *Medical Image Analysis*, 9(3), pp.179-190.
- [155] Qiao, S., Wang, H., Liu, C., Shen, W. and Yuille, A., 2019. Weight standardization. arXiv preprint arXiv:1903.10520.
- [156] Randa Aouassa CNN Based Deep Face Recognition <http://dspace.univ-guelma.dz/jspui/handle/123456789/10136>
- [157] Leuba, G., Kraftsik, R. Changes in volume, surface estimate, three-dimensional shape and total number of neurons of the human primary visual cortex from midgestation until old age. *Anat Embryol* 190, 351–366 (1994). <https://doi.org/10.1007/BF00187293>
- [158] Maunsell JH, Van Essen DC. Functional properties of neurons in middle temporal visual area of the macaque monkey. I. Selectivity for stimulus direction, speed, and orientation. *J Neurophysiol.* 1983 May ;49(5) :1127-47. doi: 10.1152/jn.1983.49.5.1127. PMID : 6864242.
- [159] 1. Braddick OJ, O'Brien JMD, Wattam-Bell J, Atkinson J, Hartley T, Turner R. Brain Areas Sensitive to Coherent Visual Motion. *Perception.* 2001 ;30(1):61-72. doi :10.1068/p3048
- [161] Zhang, & Wang, & Xu, Dongdong & Chen, (2019). Research on Scene Classification Method of High-Resolution Remote Sensing Images Based on RFPNet. *Applied Sciences.* 9. 2028. 10.3390/app9102028.
- [162] Deep Convolutional Neural Networks an Introduction <http://mbenhaddou.com/2019/12/23/deep-convolutional-neural-networks-an-introduction/> 29/05/2021
- [163] Applied Deep Learning - Part 4: Convolutional Neural Networks <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2> 30/05/2021
- [164] Deep Learning Interview questions and answers <https://www.i2tutorials.com/what-do-you-mean-by-filtering-stride-and-padding-in-convolutional-neural-network/> 02/06/2021
- [165] Student Notes : Convolutional Neural Networks (CNN) Introduction <https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/> 02/06/2021
- [166] Student Notes : Convolutional Neural Networks (CNN) Introduction <https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/> 02/06/2021
- [167] Comparison of Sigmoid, Tanh and ReLU Activation Functions <https://www.aitude.com/comparison-of-sigmoid-tanh-and-relu-activation-functions/> 03/06/2021

- [168] CNN | Introduction to Pooling Layer <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>  
03/06/2021
- [169] Foracchia, M., Grisan, E. and Ruggeri, A., 2005. Luminosity and contrast normalization in retinal images. *Medical Image Analysis*, 9(3), pp.179-190.
- [170] Qiao, S., Wang, H., Liu, C., Shen, W. and Yuille, A., 2019. Weight standardization. arXiv preprint arXiv:1903.10520.
- [171] Huang, X. and Belongie, S., 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1501-1510).
- [172] Huang, X. and Belongie, S., 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1501-1510).
- [173] Li, X., Sun, W. and Wu, T., 2019. Attentive normalization. arXiv preprint arXiv:1908.01259.
- [174] An Introduction to Convolutional Neural Networks <https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05> 06/06/2021
- [175] Goodfellow, Ian ; Bengio, Yoshua; Courville, Aaron (2016). "6.2.2.3 Softmax Units for Multinoulli Output Distributions". *Deep Learning*. MIT Press. pp. 180–184. ISBN 978-0-26203561-3.
- [176] Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer. ISBN 0-387-31073-8.
- [177] Salimans, T. and Kingma, D.P., 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems* (pp. 901-909).
- [178] Huang, X. and Belongie, S., 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1501-1510).
- [179] He, K., Girshick, R. and Dollár, P., 2019. Rethinking imagenet pre-training. In *Proceedings of the IEEE international conference on computer vision* (pp. 4918-4927).
- [180] Li, X., Sun, W. and Wu, T., 2019. Attentive normalization. arXiv preprint arXiv:1908.01259.
- [181] Huang, L., Yang, D., Lang, B. and Deng, J., 2018. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 791-800).
- [183] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, Ruslan R. Salakhutdinov *Computer Vision and Pattern Recognition* arXiv:1207.0580v1
- [184] John A. Hertz, hn Hertz, Anders S. Krogh, Richard G. Palmer *Introduction To The Theory Of Neural Computation* isbn:0201515601
- [185] Dropout : A Simple Way to Prevent Neural Networks from Overfitting Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov; 15(56):1929–1958, 2014.
- [186] <https://arxiv.org/abs/1312.6197>
- [187] David Warde-Farley, Ian J. Goodfellow, Aaron Courville, Yoshua Bengio An empirical analysis of dropout in piecewise linear networks arXiv:1312.6197v2
- [188] Kingma, D.P. and Ba, J., 2014. Adam : A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- [189] Robbins, H. and Monro, S., 1951. A stochastic approximation method. *The annals of mathematical statistics*, pp.400-407.
- [190] Boyd, S., Parikh, N. and Chu, E., 2011. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.
- [191] Geoffrey Hinton Neural Networks for machine learning online course, RMSprop-Optimization algorithms, <https://www.cs.toronto.edu/~hinton/nntut.html> last access 06/08/2020
- [192] Hoang Tran, Ashok Cutkosky Better SGD using Second-order Momentum arXiv:2103.03265v2
- [193] Duchi, J., Hazan, E. and Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

- [194] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. and Hochreiter, S., 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems* (pp. 6626-6637).
- [195] Pascanu, R., Mikolov, T. and Bengio, Y., 2013, February. On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310-1318).
- [196] You, Y., et al 2019. Large batch optimization for deep learning: Training bert in 76 minutes. arXiv preprint arXiv:1904.00962.
- [197] Reddi, S.J., Kale, S. and Kumar, S., 2019. On the convergence of adam and beyond. arXiv preprint arXiv:1904.09237.
- [198] Nesterov, Y. (1983). A method for solving a convex programming problem with convergence rate  $O(1/k^2)$ . *Soviet Math. Dokl.*, 27, 372--376.
- [199] Shazeer, N. and Stern, M., 2018. Adafactor: Adaptive learning rates with sublinear memory cost. arXiv preprint arXiv:1804.04235.
- [200] Rattray, M., Saad, D. and Amari, S.I., 1998. Natural gradient descent for on-line learning. *Physical review letters*, 81(24), p.5461.
- [201] You, Y., Gitman, I. and Ginsburg, B., 2017. Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888.
- [202] Jaderberg, M., et al 2017. Population based training of neural networks. arXiv preprint arXiv:1711.09846.
- [203] Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data* 6, 60 (2019). <https://doi.org/10.1186/s40537-019-0197-0>
- [204] Overfitting and Underfitting <https://www.youtube.com/watch?v=Ts0vGcc7fuE> 15 / 06/2021
- [205]. Abadi, M., et al 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
- [206] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A., 2017. Automatic differentiation in pytorch.
- [207] Patterson, J. and Gibson, A., 2017. *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc. ".
- [208] Seide, F. and Agarwal, A., 2016, August. CNTK: Microsoft's open-source deep-learning toolkit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 2135-2135).
- [209] Chollet, F., & others. (2015). Keras. GitHub. Retrieved from <https://github.com/fchollet/keras>
- [210] Lin, W.F., Tsai, D.Y., Tang, L., Hsieh, C.T., Chou, C.Y., Chang, P.H. and Hsu, L., 2019, March. ONNC: A compilation framework connecting ONNX to proprietary deep learning accelerators. In *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (pp. 214-218). IEEE.
- [211] Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C. and Zhang, Z., 2015. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274.
- [212] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T., 2014, November. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 675-678).
- [213] Initiation au Deep Learning avec Google Colab <https://moov.ai/fr/blog/deep-learning-avec-google-colab/> 08/07/2021
- [214] What is Google Colab <https://blog.education-ecosystem.com/what-is-google-colab/> 08/07/2021
- [215] Colaboratory <https://colab.research.google.com/> 08/07/2021
- [216] Cloud GPUs <https://cloud.google.com/gpu> 08/07/2021
- [217] <https://www.jetbrains.com/fr-fr/pycharm/> 09/07/2021
- [218] pycharm <https://doc.qt.io/qt-5/qt designer-manual.html> 09/07/2021
- [219] Python : définition et utilisation de ce langage informatique <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/> 09/07/2021

- [220] TensorFlow <https://searchdatamanagement.techtarget.com/definition/TensorFlow> 09/07/2021
- [221] About Keras <https://keras.io/about/> 09/07/2021
- [222] NumPy <https://numpy.org/doc/stable/user/whatisnumpy.html> 09/07/2021
- [223] OpenCV <https://opencv.org/courses/> 11/07/2021
- [224] Pillow <https://pillow.readthedocs.io/en/stable/> 11/07/2021
- [225] Matplotlib <https://matplotlib.org/> 11/07/2021
- [226] Scikit-learn Machine Learning in Python <https://scikit-learn.org/stable/> 11/07/2021
- [227] PDFlib library <http://www.fpdf.org/> 11/07/2021
- [228] Lung fields segmentation on chest X-ray images <https://github.com/rani700/xray> 04/02/2021
- [229] Talha Burak Alakus, Ibrahim Turkoglu, Comparison of deep learning approaches to predict covid-19 infection, *Chaos Solitons Fractals* 140 (2020), 110120.
- [230] Ali Narin, Ceren Kaya, and Ziyne Pamuk, Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849* (2020).
- [231] Do, Cuong, and Lan Vu. An approach for recognizing COVID-19 cases using Convolutional Neural Networks applied to CT scan images. *Applications of Digital Image Processing XLIII*. Vol. 11510. International Society for Optics and Photonics, 2020.
- [232] Halgurd S Maghdid, Aras T Asaad, Kayhan Zrar Ghafoor, Ali Safaa Sadiq, and Muhammad Khurram Khan. Diagnosing covid-19 pneumonia from x-ray and ct images using deep learning and transfer learning algorithms. *arXiv preprint arXiv: 2004.00038*, (2020).
- [233] Ioannis D Apostolopoulos, Tzani A Mpesiana, Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks, *Phys. Eng. Sci. Med.* 43 (2) (2020) 635–640.
- [234] Taban Majeed, Rasber Rashid, Dashti Ali, and Aras Asaad. Covid-19 detection using cnn transfer learning from x-ray images. *medRxiv*, (2020).
- [235] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), pp.1929-1958.
- [236] Loshchilov, I. and Hutter, F., 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- [237] Choe, J. and Shim, H., 2019. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2219-2228).
- [238] Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł. and Hinton, G., 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.
- [239] Grandvalet, Y. and Bengio, Y., 2006. Entropy Regularization.
- [240] Yao, Y., Rosasco, L. and Caponnetto, A., 2007. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2), pp.289-315.
- [241] Kingma, D.P., Salimans, T. and Welling, M., 2015. Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems* (pp. 2575-2583).
- [242] Wan, L., Zeiler, M., Zhang, S., Le Cun, Y. and Fergus, R., 2013, February. Regularization of neural networks using dropconnect. In *International conference on machine learning* (pp. 1058-1066).
- [243] Elden, L., 1977. Algorithms for the regularization of ill-conditioned least squares problems. *BIT Numerical Mathematics*, 17(2), pp.134-145.
- [244] Park, M.Y. and Hastie, T., 2007. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4), pp.659-677.
- [245] Gal, Y. and Ghahramani, Z., 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems* (pp. 1019-1027).

[246] Kim, S.J., Koh, K., Lustig, M., Boyd, S. and Gorinevsky, D., 2007. An interior-point method for large-scale  $\ell_1$ -regularized least squares. *IEEE journal of selected topics in signal processing*, 1(4), pp.606-617.

[247] Oster, H.S. and Rudy, Y., 1992. The use of temporal information in the regularization of the inverse problem of electrocardiography. *IEEE transactions on biomedical engineering*, 39(1), pp.65-75.