

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN MAÎTRISE EN MATHÉMATIQUES ET  
INFORMATIQUE APPLIQUÉES

PAR  
Mohamed Elamine Khoudour

Détection de la maladie du coronavirus (COVID-19) basée sur des caractéristiques  
profondes et les machines à vecteurs de support

DÉCEMBRE 2023

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

# Résumé

Dans le contexte d'urgence lié au retour du COVID-19 et de ses variants, un diagnostic précis et rapide revêt une importance capitale. Les radiographies thoraciques sont cruciales pour le dépistage précoce du virus, permettant de différencier précisément d'autres formes de pneumonie. De plus, ces examens fournissent des informations importantes sur la gravité de la maladie et son évolution au fil du temps, ce qui est essentiel pour garantir un traitement adéquat. L'objectif de cette recherche est de développer un modèle hybride CNN-SVM qui combine les capacités des réseaux de neurones convolutionnels profonds (CNN) pour extraire des caractéristiques à grande échelle, avec celles des machines à vecteurs de support (SVM) pour la classification. Le choix des CNNs dans cette étude découle de leur adoption généralisée et de leurs performances prouvées dans la littérature, notamment en ce qui concerne la classification des radiographies du COVID-19. Les SVMs ont été optimisés par rapport aux vecteurs de caractéristiques extraits par les CNNs pour améliorer les performances globales de l'approche. L'étude a été menée sur une base de données réelles, récente contenant des images radiographiques de la COVID-19. Les expérimentations sur 3 ensembles différents (COVID-19 vs Normale), (COVID-19 vs Normale vs Opacité pulmonaire) et (COVID-19 vs Normale vs Opacité pulmonaire vs Pneumonie virale) ont montré des précisions encourageantes, avec des taux de reconnaissance supérieurs par rapport aux études utilisant les CNNs seuls. De plus, cette approche a été validée sur un ensemble de données distinct lié à la population pakistanaise, afin d'évaluer sa robustesse face aux variations des données, obtenant des résultats satisfaisants.

**Mots-clés** : CNN, SVM, Classification, Radiographie.

# Abstract

In the emergency context of COVID-19 and its variants, accurate and rapid diagnosis is of paramount importance. Chest X-rays are crucial for early detection of the virus, enabling accurate differentiation from other forms of pneumonia. In addition, these examinations provide important information on the severity of the disease and its evolution over time, which is essential to ensure adequate treatment. The aim of this research is to develop a hybrid CNN-SVM model that combines the capabilities of deep convolutional neural networks (CNNs) for large-scale feature extraction, with those of support vector machines (SVMs) for classification. The choice of CNNs in this study stems from their widespread adoption and proven performance in the literature, particularly in the classification of COVID-19 X-rays. The SVMs were optimized using the feature vectors extracted by the CNNs to improve the overall performance of the approach. The study was conducted on a recent public database of radiographic images of COVID-19. Experiments on 3 different sets (COVID-19 vs Normal), (COVID-19 vs Normal vs Lung Opacity) and (COVID-19 vs Normal vs Lung Opacity vs Viral Pneumonia) showed encouraging accuracies, with higher recognition rates compared to studies using CNNs alone. In addition, this approach was validated on a separate dataset linked to population in Pakistan, in order to assess its robustness to data variations, obtaining satisfactory results.

**Keywords :** CNN, SVM, Classification, Radiography.

# Remerciements

Je tiens à exprimer ma profonde gratitude envers mon directeur de recherche Professeur Ismail Biskri, pour son soutien inestimable tout au long de mon travail de recherche. Ses conseils avisés, sa patience infinie et son engagement envers mon développement académique ont été essentiels pour ma réussite. Sa passion pour la recherche a été une source d'inspiration constante, et je suis reconnaissant de son rôle déterminant dans l'atteinte de mes objectifs académiques.

Je tiens à exprimer ma sincère reconnaissance envers mes Professeurs évaluateurs " Nadia Ghazzali " et " Boucif Amar Bensaber " , qui ont consacré le temps et l'engagement nécessaires pour évaluer en détail mon travail de recherche.

Je tiens à exprimer ma gratitude envers tous mes amis pour leur soutien constant et leurs encouragements tout au long de mon parcours académique. De plus, je souhaite exprimer ma profonde reconnaissance envers l'équipe du Laboratoire d'Intelligence Artificielle Appliquée (LI2A) pour leur précieux soutien matériel qui a contribué de manière significative à la réussite de mon projet.

Enfin, je tiens à exprimer ma profonde gratitude envers mes chers parents et mon cher frère pour leur soutien inébranlable tout au long de mon parcours académique. Je suis véritablement reconnaissant pour leur présence constante à mes côtés, les moments où ils m'ont encouragé à poursuivre mes objectifs académiques, et la façon

dont ils ont toujours cru en moi, même dans les moments les plus difficiles. Leur soutien fiable a été la pierre angulaire de mon succès académique, et je leur en suis éternellement reconnaissant.

# Table des matières

<b>Résumé</b>	ii
<b>Abstract</b>	iv
<b>Remerciements</b>	v
<b>Table des matières</b>	vii
<b>Liste des tableaux</b>	x
<b>Table des figures</b>	xii
<b>1 Introduction</b>	1
1.1 Mise en contexte . . . . .	1
1.2 Objectifs du mémoire . . . . .	7
1.3 Structure et organisation du mémoire . . . . .	7
<b>2 Réseaux de neurones à convolution (CNN)</b>	9
2.1 Architecture des CNNs . . . . .	10
2.1.1 La couche de convolution . . . . .	11
2.1.2 Couche de sous-échantillonnage (Pooling) . . . . .	13
2.1.3 Les fonctions d'activation . . . . .	14
2.1.4 Couche de Mise à Plat . . . . .	16
2.1.5 La couche entièrement connectée . . . . .	17
2.2 Optimisation et amélioration des CNNs . . . . .	18
2.2.1 Normalisation par lot (Batch Normalization) . . . . .	19



2.2.2	Dropout	19
2.3	Les CNNs profonds	20
2.4	Apprentissage par transfert	22
<b>3</b>	<b>Support Vector Machine (SVM)</b>	<b>24</b>
3.1	Maximisation de la marge	27
3.2	Soft margin SVM	29
3.3	Les noyaux	30
3.4	SVM vers une classification Multi-Classe	32
<b>4</b>	<b>Hybride CNN-SVM</b>	<b>33</b>
<b>5</b>	<b>Méthodologie</b>	<b>38</b>
5.1	Datasets	39
5.1.1	COVID-19 Radiography Database	40
5.1.2	COVID-19 Pakistani Patients X-ray Image Dataset	41
5.2	Prétraitement des données	42
5.2.1	Égalisation d'histogramme	43
5.3	Extraction des fonctionnalités	46
5.3.1	Sélection des CNNs	46
5.3.2	Extraction de caractéristiques via les CNNs	57
5.4	Optimisation des paramètres pour SVM	58
5.4.1	Recherche par grille (Grid Search)	60
5.5	Évaluation de la performance CNN-SVM	61
<b>6</b>	<b>Expérimentations et discussion</b>	<b>64</b>
6.1	Environnement matériel	64
6.2	Environnement logiciel	65
6.3	Expérimentations et discussion	66
6.3.1	Résultats expérimentaux deux classes	66
6.3.2	Résultats expérimentaux trois classes	70
6.3.3	Résultats expérimentaux quatre classes	74

6.4	Évaluation de VGG16-SVM sur de nouvelles données de test . . . . .	77
6.5	Etude comparative . . . . .	78
7	Conclusion et perspectives	80
	Bibliographie	83
A	Detection of coronavirus Disease (COVID-19) based on Deep Fea- tures and Support Vector Machine	87

# Liste des tableaux

5.1 Ensemble de données utilisées dans l'étude.	45
5.2 Échantillons proposés.	59
5.3 Paramètres optimaux.	60
6.1 Résultats de la classification problème à deux classes.	67
6.2 Analyse de la performance par rapport à chaque catégorie.	68
6.3 Résultats de la classification problème à trois classes.	71
6.4 Évaluation de la performance par rapport à trois classes.	73
6.5 Résultats de la classification problème à quatre classes.	75
6.6 Évaluation de la performance par rapport à quatre classes.	76
6.7 Comparaison de notre approche avec d'autres études.	79

# Table des figures

<a href="#">2.1 Architecture du CNN.</a>	11
<a href="#">2.2 Opération de convolution.</a>	12
<a href="#">2.3 Opération de pooling.</a>	14
<a href="#">2.4 Fonctions d'activation.</a>	16
<a href="#">2.5 Mise à plat.</a>	17
<a href="#">2.6 Réseau neuronal à couche entièrement connectée.</a>	18
<a href="#">2.7 Exploration du dropout dans les réseaux neuronaux.</a>	20
<a href="#">2.8 Les diverses architectures profondes des CNNs <a href="#">[1]</a>.</a>	22
<a href="#">3.1 L'hyperplan séparatif.</a>	25
<a href="#">3.2 Classes linéairement séparables.</a>	26
<a href="#">3.3 Solutions possibles pour l'hyperplan séparatif.</a>	28
<a href="#">3.4 Illustration de la notion de marge.</a>	29
<a href="#">3.5 Soft margin SVM.</a>	30
<a href="#">3.6 Projection non-linéaire des classes.</a>	31
<a href="#">3.7 Adaptation des SVMs à la classification multi-classe.</a>	32
<a href="#">4.1 Hybride CNN-SVM.</a>	35
<a href="#">5.1 Schéma bloc de la méthodologie.</a>	39
<a href="#">5.2 Images radiographiques : COVID-19 (A), Opacité pulmonaire (B), Nor-</a>	41
<a href="#">male (C), et Pneumonie virale (D).</a>	
<a href="#">5.3 Images radiographiques COVID-19 (A), Normale (B).</a>	42
<a href="#">5.4 Images radiographiques avant l'égalisation d'histogramme.</a>	44
<a href="#">5.5 Images radiographiques après l'égalisation d'histogramme.</a>	44

5.6	<i>Subdivision de nouvelles données améliorées.</i>	45
5.7	<i>Architecture du VGG16.</i>	48
5.8	<i>Nouvelle architecture du VGG16.</i>	48
5.9	<i>Architecture du VGG19.</i>	49
5.10	<i>Nouvelle architecture du VGG19.</i>	50
5.11	<i>Convolution profonde separable.</i>	51
5.12	<i>Architecture du MobileNet.</i>	52
5.13	<i>Nouvelle architecture du MobileNet.</i>	52
5.14	<i>Architecture du Inception-V3.</i>	54
5.15	<i>Nouvelle architecture du Inception-V3.</i>	54
5.16	<i>Architecture du DenseNet201.</i>	56
5.17	<i>Nouvelle architecture du DenseNet201.</i>	56
5.18	<i>Visualisation des premières couches de convolution.</i>	58
5.19	<i>Interprétation de la courbe ROC.</i>	63
6.1	<i>Analyse des performances via courbes ROC sur ensemble 1 (2 classes).</i>	68
6.2	<i>Matrice de confusion pour deux classes.</i>	69
6.3	<i>Les échantillons mal classifiés.</i>	70
6.4	<i>Analyse des performances via courbes ROC sur ensemble 2 (3 classes).</i>	72
6.5	<i>Matrice de confusion pour trois classes.</i>	74
6.6	<i>Analyse des performances via courbes ROC sur ensemble 3 (4 classes).</i>	75
6.7	<i>Matrice de confusion pour quatre classes.</i>	77
6.8	<i>Résultats de la prédiction : A : cas de COVID, B : cas Normaux.</i>	78

# Chapitre 1

## Introduction

### 1.1 Mise en contexte

De nos jours, les avancées technologiques de l'intelligence artificielle ont ouvert la voie à de nouvelles possibilités pour les machines, les rapprochant de plus en plus de la capacité de résoudre des problèmes complexes de manière autonome. Ces avancées se reflètent dans diverses capacités, notamment l'apprentissage par l'exemple, la résolution de problèmes, la reconnaissance d'objets, ainsi que la compréhension et la réponse à la parole. Toutes ces fonctionnalités dépendent fortement de la capacité des machines à traiter d'énormes quantités de données, qui servent de base pour améliorer leurs performances et leur compréhension des tâches spécifiques qu'elles sont conçues pour accomplir. Cette convergence croissante entre l'intelligence artificielle et l'intelligence humaine ouvre des perspectives prometteuses pour l'avenir de la technologie.

L'apprentissage automatique ou Machine Learning tient une place essentielle dans le domaine de l'intelligence artificielle, grâce à sa vaste gamme d'algorithmes permettant de résoudre des problèmes de classification, qu'ils soient supervisés ou non

supervisés. En développant des systèmes capables d'apprendre et de progresser au fil des expériences et des scénarios fournis, cette capacité a permis à ce domaine d'avoir un impact sur divers secteurs tels que la santé, la sécurité et l'industrie, en permettant aux systèmes de s'améliorer et de fournir des solutions intelligentes à des problèmes complexes. Les réseaux de neurones convolutifs (CNN) et les machines à vecteurs de support (SVM) sont deux techniques d'apprentissage automatique adaptées à la reconnaissance d'images complexes, notamment les images médicales telles que les radiographies de Covid-19, la tomographie et les scanners des tumeurs cancéreuses.

Les CNNs ont connu un grand succès dans ce type d'applications. Marios Constantinou et al. [2] ont atteint une précision de 96% en utilisant cinq réseaux de neurones convolutionnels profonds pour la détection de la COVID-19 à partir des images radiographiques. De même, Soheila Saeedi et al. [3] ont obtenu des résultats remarquables en développant deux réseaux de neurones convolutionnels distincts, atteignant des précisions de 96,47% et 95,63% respectivement, pour la détection de tumeurs à partir d'images IRM du cerveau. L'efficacité des CNNs provient essentiellement de leur capacité à extraire automatiquement des caractéristiques pertinentes en utilisant des couches convolutives, ce qui améliore considérablement leur capacité de classification.

Les SVMs, ont été développées par Vladimir Vapnik et Rauf Izmailov [4], dans le but de résoudre des problèmes de classification, de régression et de détection d'anomalies, ce qui les rend également compatibles avec ce genre d'application. Tim Adams et al. [5] ont proposé un système automatisé basé sur les SVM pour le diagnostic des tumeurs pulmonaires sur les scanners, atteignant une précision remarquable par rapport à l'ensemble de données minimal utilisé.

Récemment, plusieurs travaux de recherche ont mis en œuvre une approche combinatoire visant à fusionner ces deux applications afin de bénéficier de leurs avantages respectifs. Cette approche consiste à utiliser les CNNs pour extraire les caractéristiques des données, puis à remplacer la dernière couche des CNNs, responsable de la classi-

fication, par un SVM qui prend en charge cette tâche. Cette combinaison a rencontré un succès notable dans le domaine de la reconnaissance de formes, notamment pour la classification de chiffres et de lettres manuscrits. Dans leur article, Benkadja Abdallah. et al. [6], ont démontré l'excellente performance de cette approche par rapport à l'utilisation de chaque application de manière indépendante. Par ailleurs, Xiao-Xiao Niu et Ching Y. Suen [7] ont obtenu un taux de reconnaissance impressionnant de 99,81 % sans rejet en utilisant un modèle hybride CNN-SVM pour la classification des chiffres manuscrits MNIST. Néanmoins, des questions persistent quant à l'efficacité réelle de cette approche face à des données complexes, comme les images médicales, où la similarité entre les données peut être extrêmement élevée. Par conséquent, il est essentiel de mener des recherches approfondies pour déterminer si cette méthode conserve son efficacité dans de telles situations.

La pandémie COVID-19, causée par le coronavirus SARS-CoV-2, a profondément bouleversé nos vies depuis 2019. Malgré les efforts pour contenir sa propagation, il est évident que la COVID-19 est là pour rester, en raison des différentes variantes du virus. Nous devons nous adapter à cette réalité et prendre les mesures nécessaires pour nous protéger contre une nouvelle vague potentiellement mortelle du virus. Il existe diverses méthodes de détection de ce virus dans le corps humain, utilisant principalement des approches biologiques et chimiques.

Les tests PCR sont largement reconnus comme la référence pour le diagnostic de l'infection, étant considérés comme le « gold standard », cependant, les tests PCR peuvent donner des résultats faussement négatifs. Cela signifie que certains patients COVID-19 peuvent être testés négatifs alors qu'ils sont réellement infectés. De plus, ces tests mettent souvent beaucoup de temps à fournir des résultats, ce qui peut augmenter le délai de diagnostic [8]. Le test d'antigène est une alternative plus rapide que le test PCR, mais peut être légèrement moins sensible. Les tests sérologiques aident à identifier les infections passées mais ne conviennent pas à une détection précoce.



Il est bien connu que la radiographie est une technique d'imagerie utilisée depuis de nombreuses années pour diagnostiquer diverses maladies. Cependant, en raison de la pandémie du COVID-19, les radiographies thoraciques ont pris une importance particulière dans le diagnostic précoce de la maladie [9]. De plus, la radiographie thoracique peut fournir des informations importantes sur la gravité de la maladie et son évolution au fil du temps. Mais, avec l'approche de la saison grippale, le diagnostic de l'épidémie deviendra difficile en raison des nombreuses similitudes entre différents types de pneumonie et la Covid-19, ce qui ajoutera une charge supplémentaire au personnel de la santé.

Au cours des derniers temps, de nombreuses recherches ont été menées pour explorer la détection du COVID-19 à l'aide d'images radiographiques en utilisant différentes approches basées sur l'intelligence artificielle. Diverses techniques, telles que l'apprentissage par transfert, l'apprentissage profond et la conception de nouvelles architectures de CNN, ont été proposées dans le but d'améliorer les performances des modèles et de permettre une classification précise du COVID-19, des maladies pulmonaires normales ainsi que d'autres affections pulmonaires.

- Cagin Polat et al. [10] ont obtenu une précision de 97,1 % en utilisant un réseau neuronal convolutif appelé nCoV-NET, qui a été optimisé en utilisant l'architecture DenseNet-161. Leur modèle a été entraîné et évalué sur un ensemble de données comprenant deux cent quatre-vingt-dix-neuf cas du COVID-19 et mille cinq cent vingt-deux cas de non-COVID-19.

- Muhammad Umair et al. [11] ont utilisé quatre réseaux neuronaux profonds pré-entraînés, à savoir VGG16, ResNet-50, DenseNet-121 et MobileNet, en utilisant l'apprentissage par transfert sur un ensemble de données composé de sept mille deux cent trente-deux images de radiographie thoracique, comprenant des cas de COVID-19 et des images normales, avec des taux de précision respectifs de 83,27 %, 92,48 %, 96,49 % et 96,48 %.

- Fátima Saiz et al. [12] ont développé une architecture de détection d'objets qui a été formée et testée sur un ensemble de données composé de mille cinq cents images de patients non infectés, infectés par le COVID-19 et atteints de pneumonie. Pour cette étude, VGG16 a été utilisé comme modèle et a obtenu une sensibilité de 94,92 % et une spécificité de 92,00 % dans la détection du COVID-19.

- Asma Abbas et al. [13] ont validé un réseau de neurones convolutifs (CNN) profond appelé Decompose, Transfer, Compose (DeTraC) pour la classification des radiographies pulmonaires dans le contexte de la COVID-19, en utilisant un petit ensemble de données. DeTraC est capable de gérer les irrégularités dans les ensembles de données d'images en analysant les frontières de classe à l'aide d'un mécanisme de décomposition de classe. DeTraC a obtenu une précision de 93,1 % dans la détection des radiographies COVID-19.

- Asif Iqbal Khan et al. [14] ont développé un modèle de réseau de neurones convolutionnels, CoroNet, pour détecter la COVID-19 sur des radiographies. CoroNet a démontré une précision exceptionnelle de 89,6 % dans la détection globale, avec des taux de précision de 93 % et un rappel de 98,2 % pour les cas de COVID-19 dans une classification en quatre catégories, ainsi qu'une précision de 95 % dans une classification en trois catégories.

- Ioannis Apostolopoulos et al. [15] ont réentraîné cinq réseaux neuronaux profonds pré-entraînés (VGG19, MobileNet v2, Inception, Xception et Inception ResNet v2) sur un ensemble de données composé de mille quatre cent vingt-sept images par rayons X, comprenant deux cent vingt-quatre images de Covid-19, sept cents images de pneumonie bactérienne et cinq cent quatre images normales. Les résultats suggèrent que VGG19 et MobileNet v2 obtiennent la meilleure précision de classification par rapport aux autres CNN (98,75 % et 96,78 % respectivement pour les deux classes, et 93,48 % et 94,72 % pour les trois classes).

- Rachna Jain et al. [16] ont utilisé les modèles de réseaux de neurones convolutifs (CNN) basés sur l'apprentissage profond (Inception V3, Xception et ResNeXt) sur un ensemble de données de six mille quatre cent trente-deux échantillons de scans de radiographie thoracique. Parmi ces échantillons, cinq mille quatre cent soixante-sept ont été utilisés pour l'entraînement du modèle (comprenant quatre cent quatre-vingt-dix cas de Covid-19, mille trois cent quarante-cinq cas normaux et trois mille six cent trente-deux cas de pneumonie), tandis que neuf cent soixante-cinq ont été utilisés pour la validation. Le modèle Xception s'est démarqué de manière significative avec un taux de classification de 97,97 % dans la détection des images de radiographie thoracique, surpassant de manière significative les autres modèles.

Les études présentées indiquent que les CNNs profonds sont bien adaptés à la détection du COVID-19 à partir d'images radiographiques. Cependant, la plupart de ces recherches se confrontent à la limitation du nombre de données disponibles, ce qui oblige les auteurs à utiliser des techniques d'augmentation de données pour nourrir leurs réseaux. Cette contrainte peut rendre difficile la validation des résultats sur des échantillons réels ou sur un nombre important de données. De plus, il est important de noter que certaines de ces études considèrent les cas de pneumonie comme des cas non-COVID, sans spécifier le type de pneumonie en question. De ce fait, l'identification précise des différentes affections pulmonaires pourrait être limitée. Un autre aspect à considérer est que ces études ont été validées sur des échantillons restreints d'images provenant d'une seule source, ce qui pourrait engendrer une incertitude dans les résultats. Les connaissances acquises grâce à ces études seront mises à profit dans notre recherche, qui vise à prédire la COVID-19 par rapport à d'autres pneumonies à partir des images radiographiques, en utilisant l'approche hybride CNN-SVM. Nous procéderons ensuite à une comparaison de ces résultats avec les études utilisant uniquement les CNNs.

Cette recherche a fait l'objet d'une publication à la conférence ICMLA 2023, "22nd International Conference on Machine Learning and Applications", l'intitulée de cette

publication est "Detection of coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine". Nous avons inclus cette publication en Annexe A de ce mémoire.

## 1.2 Objectifs du mémoire

L'objectif de cette recherche est d'explorer une approche innovante qui combine les techniques d'apprentissage automatique CNN et SVM afin d'améliorer la classification des images radiographiques liées à la COVID-19. Nous avons opté pour les CNNs et le SVM en raison de leur performance avancée en matière de classification d'images médicales. Parallèlement, nous allons comparer cette approche avec les recherches existantes qui utilisent séparément les CNNs. De plus, nous avons soumis cette approche à des tests sur des données totalement indépendantes de l'ensemble d'entraînement et de validation afin d'évaluer sa capacité à maintenir ses performances face aux variations des données.

## 1.3 Structure et organisation du mémoire

Après l'introduction présentée dans le chapitre 1, le mémoire est structuré en cinq chapitres distincts avant de conclure :

- Au chapitre 2, nous explorons en profondeur les réseaux de neurones convolutifs (CNN).
- Au chapitre 3, nous concentrons sur les Support Vector Machines (SVM).

- Au chapitre 4, nous présentons l'approche hybride CNN-SVM.
- Au chapitre 5, nous décrivons la méthodologie utilisée dans notre recherche.
- Au chapitre 6, nous présentons les résultats de nos expérimentations.

# Chapitre 2

## Réseaux de neurones à convolution (CNN)

Les réseaux de neurones convolutifs sont largement reconnus comme les modèles les plus performants pour la classification d'images complexes. Leur succès repose principalement sur leur capacité à extraire automatiquement des caractéristiques pertinentes en utilisant des couches de convolution, ce qui entraîne une nette amélioration de leur capacité de classification. En général, ces réseaux prennent en entrée des images, identifient automatiquement les caractéristiques propres à chaque image, puis les exploitent pour réaliser la classification. Les recherches menées par Yann LeCun et ses collaborateurs [17] ont apporté une contribution majeure à la compréhension et à l'avancement des réseaux de neurones convolutifs (CNN) dans le domaine du traitement d'images.

L'analyse approfondie de ces derniers a révélé plusieurs aspects essentiels des CNNs et a illustré leur utilité dans divers domaines tels que la reconnaissance vocale, le traitement du langage, la traduction, la génération de texte, ainsi que dans les modèles de conversation avancés. Leur pertinence s'étend également au domaine médical, où ils

sont capables de prédire la structure des protéines et d'analyser des données médicales complexes pour extraire des informations cruciales.

Les réseaux de neurones convolutifs (CNN) se différencient des réseaux de neurones classiques par leur architecture spécialement conçue pour le traitement d'images. Tandis que les réseaux de neurones classiques sont habituellement employés pour des tâches de classification et d'analyse de données en général, les CNNs ont été spécifiquement élaborés pour tirer parti des caractéristiques spatiales et locales des images. Cette adaptation les rend particulièrement appropriés pour les applications en vision par ordinateur. Grâce à l'utilisation de couches convolutives et de techniques de pooling, les CNNs sont capables d'analyser et de capturer efficacement des structures visuelles et des motifs dans les images, ce qui les rend particulièrement efficaces pour des tâches telles que la détection d'objets, la reconnaissance d'objets et la segmentation sémantique.

## 2.1 Architecture des CNNs

La structure fondamentale d'un réseau de neurones convolutifs (CNN) est composée de quatre couches clés, telles qu'illustrées dans la Figure 2.1 : en premier lieu, la couche de convolution, qui extrait des caractéristiques spécifiques des données en utilisant des filtres ; ensuite, la couche de pooling, qui réduit la dimensionnalité des caractéristiques extraites ; suivie par la couche d'activation, qui est responsable de l'introduction de non-linéarités dans le réseau ; enfin, la couche entièrement connectée, qui établit des connexions entre chaque neurone et les neurones de la couche précédente en vue de la prise de décision finale [18].

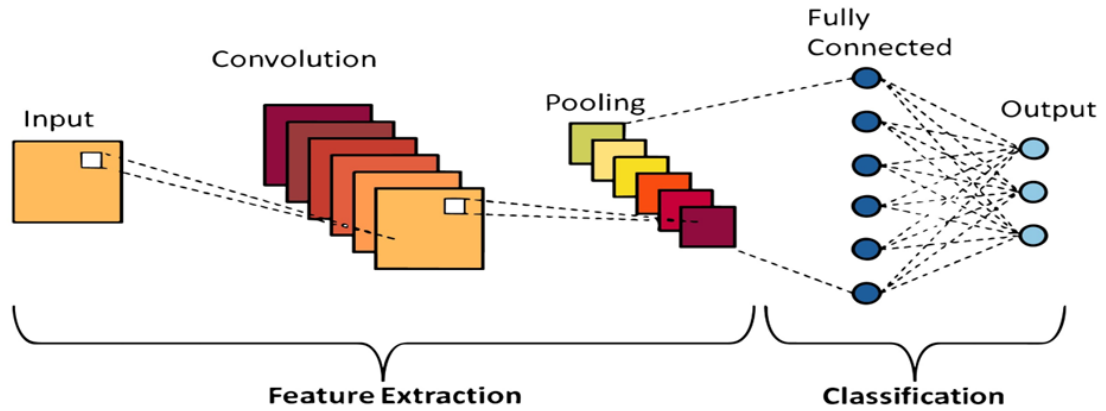


FIGURE 2.1 – Architecture du CNN.

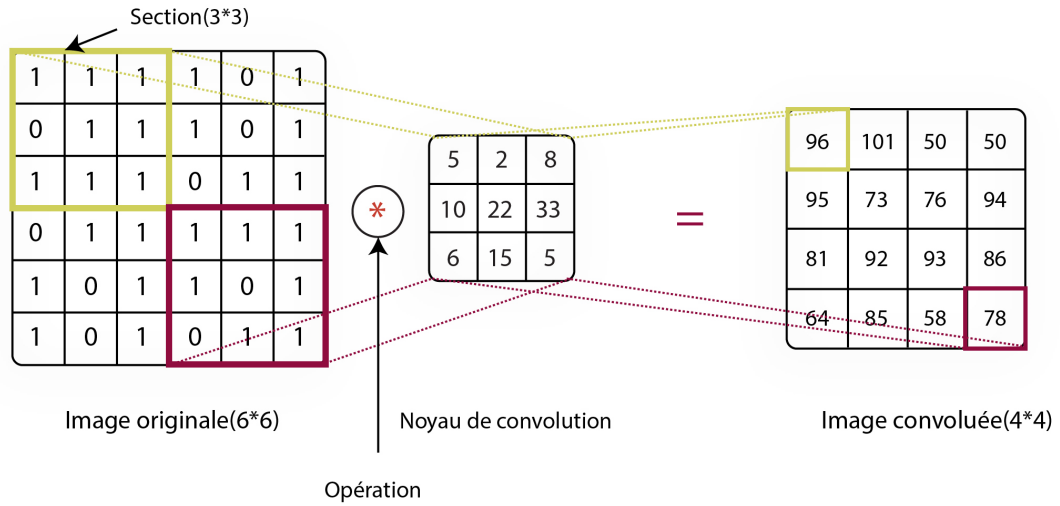
### 2.1.1 La couche de convolution

La convolution constitue l'élément central d'un réseau de neurones convolutif. Son objectif est de repérer la présence d'un ensemble de caractéristiques (features) au sein des images en entrée. Pour ce faire, une opération de filtrage par convolution est mise en œuvre : le principe consiste à faire glisser un filtre représentant la caractéristique sur l'image, puis à calculer le produit de convolution entre ladite caractéristique et chaque portion de l'image balayée, comme illustré dans la Figure 2.2. Cela permet de générer une nouvelle image (généralement de taille réduite) en sortie. L'opération de convolution au sein d'un réseau de neurones convolutif (CNN) est définie de la manière suivante :

$$F(i, j) = \sum_{m=0} \sum_{n=0} X[i + m, j + n] \cdot K[m, n] + b \quad (2.1)$$

Où :  $F(i, j)$  est la valeur de la carte des caractéristiques en sortie à la position  $(i, j)$ ,  $X$  est la carte des caractéristiques en entrée,  $K$  est le noyau de convolution,  $m$  et  $n$  sont les indices du noyau de convolution et  $b$  est le terme de biais.



FIGURE 2.2 – *Opération de convolution.*

Sélectionner les paramètres de convolution au sein des réseaux de neurones convolutifs (CNN) joue un rôle essentiel dans la recherche d'excellentes performances. Parmi les paramètres qui nécessitent une attention particulière :

- Taille du noyau (kernel size) : la sélection de la taille du noyau de convolution détermine la zone couverte par la convolution. Des tailles courantes incluent 3x3, 5x5 et 7x7. Les filtres de petite taille captent des caractéristiques plus fines, tandis que les filtres plus grands peuvent saisir des caractéristiques plus globales.

- Stride (pas) : Le stride définit l'écart entre les déplacements successifs du noyau de convolution. Un stride plus grand a le potentiel de réduire la dimension spatiale des caractéristiques obtenues, offrant ainsi des avantages tels que la réduction du calcul requis ou la possibilité de réaliser un sous-échantillonnage.

Plusieurs types de convolutions existent, [19] et bien que la version de base soit couramment employée, il peut s'avérer utile de connaître les outils à disposition :

- La convolution dilatée : Elle fonctionne de manière similaire à la convolution de base, avec la particularité que le noyau est étendu (par exemple, un pixel sur deux est pris en compte lors du calcul de la convolution). Un paramètre supplémentaire intervient : le taux de dilatation, qui indique le nombre de pixels à sauter.

- La convolution séparable : Elle décompose les convolutions complexes en une série de convolutions plus simples. Cela optimise les calculs en réduisant la complexité. C'est une approche efficace pour le traitement d'image ou de données.

- La convolution transposée : Également appelée déconvolution, construit la sortie de manière similaire à ce qui se passerait si l'opération de convolution était inversée sur l'image. Cela implique que la convolution est appliquée de manière expansive, ce qui permet d'augmenter la taille spatiale de la sortie.

### 2.1.2 Couche de sous-échantillonnage (Pooling)

Ce type de couche est généralement placé entre deux couches de convolution. Elle reçoit en entrée plusieurs cartes de caractéristiques et applique l'opération de regroupement (pooling) à chacune d'entre elles. L'objectif du regroupement est de réduire la taille des images tout en préservant les caractéristiques importantes. Cette opération implique la découpe de l'image en cellules régulières et de conserver la valeur maximale de chaque cellule, comme illustrer dans la Figure [2.3](#). Souvent, des cellules carrées de petite taille sont utilisées pour éviter la perte d'informations cruciales. Les choix les plus courants sont des cellules adjacentes de dimensions  $2 \times 2$  ou  $3 \times 3$  pixels. Il existe trois types de regroupement :

- Max pooling : Cette méthode consiste à extraire la valeur maximale de la sélection. C'est la méthode la plus fréquemment employée en raison de sa rapidité de calcul (instantanée) et de sa capacité à simplifier efficacement l'image.

- Mean pooling : Cette méthode implique, la prise de la moyenne des pixels dans la sélection. La somme de toutes les valeurs est calculée et ensuite divisée par le nombre de valeurs. Cela génère ainsi une valeur intermédiaire pour représenter ce groupe de pixels.

- Sum pooling : Il consiste à calculer la somme des valeurs sans les diviser par le nombre de valeurs (seules les valeurs sont additionnées).

Le Max Pooling est couramment utilisé dans les réseaux de neurones convolutionnels (CNN) et est largement recommandé pour la plupart des tâches de vision par ordinateur, notamment la classification d'images, la détection d'objets et la segmentation sémantique.

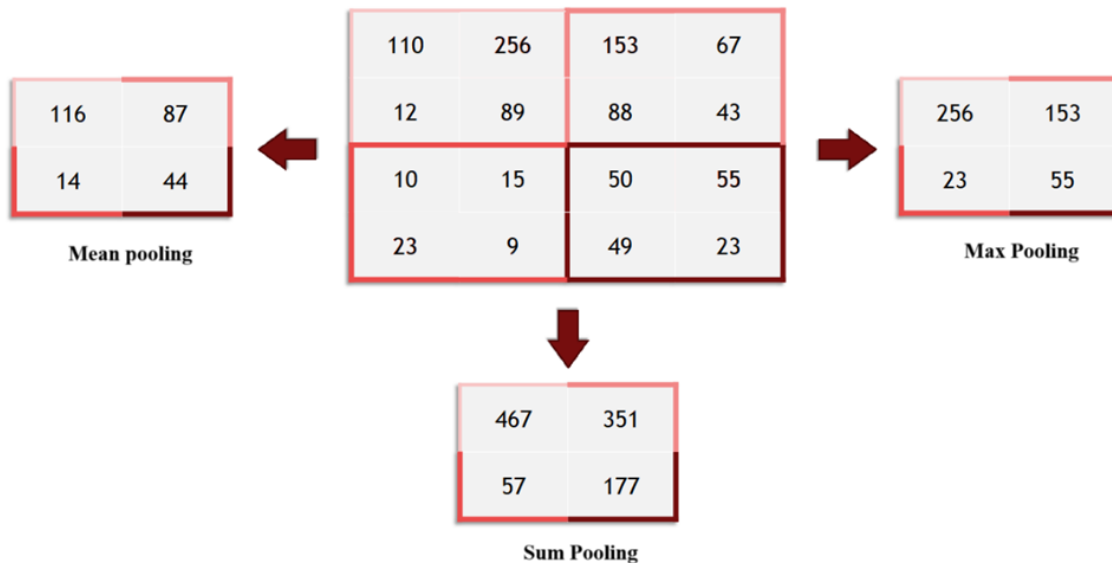


FIGURE 2.3 – Opération de pooling.

### 2.1.3 Les fonctions d'activation

La fonction d'activation joue un rôle crucial en déterminant si un neurone doit être activé ou non. Elle accomplit cette tâche en effectuant des opérations mathématiques

simples qui évaluent la pertinence de l'information d'entrée du neurone dans le contexte global de la prédiction. Son objectif central consiste à introduire de la complexité non linéaire au sein d'un réseau neuronal artificiel, engendrant ainsi une réponse spécifique à partir d'un ensemble de valeurs d'entrée qui est acheminé à travers une couche donnée.

Il existe différents types de fonctions d'activation utilisées dans les réseaux de neurones. La démonstration géométrique en 2 dimensions est illustrée dans la Figure [2.4](#). Parmi les fonctions fréquemment utilisées, on peut citer :

- Fonction d'activation linéaire : La fonction d'activation linéaire réalise une somme pondérée des entrées et renvoie le résultat tel quel, sans transformation non linéaire. Mathématiquement, cela peut s'exprimer comme suit :

$$F(X) = X \tag{2.2}$$

- Fonction Sigmoidale : La fonction opère, en prenant un nombre en entrée et génère un nombre qui se situe dans l'intervalle entre 0 et 1 en sortie. Elle trouve principalement son utilité dans les problèmes de classification binaire, où elle fournit la probabilité d'appartenance à une classe spécifique. Mathématiquement, et calculé de la manière suivante :

$$F(X) = \frac{1}{1 + e^{-X}} \tag{2.3}$$

- Fonction d'activation ReLU : La fonction d'activation ReLU annule les valeurs négatives pour accélérer les calculs et introduire de la non-linéarité dans le modèle.

Mathématiquement, elle est calculée comme suit :

$$\text{ReLU}(X) = \max(0, X) \quad (2.4)$$

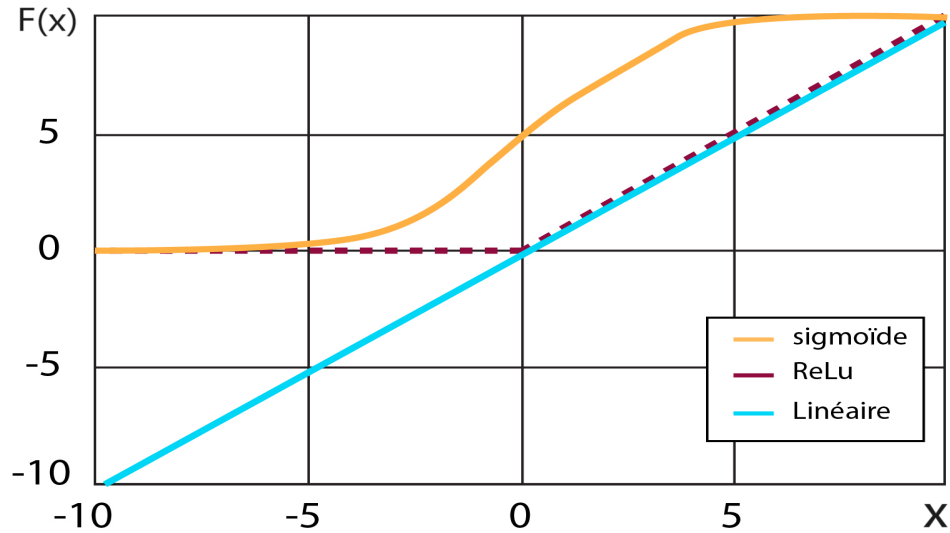


FIGURE 2.4 – Fonctions d'activation.

#### 2.1.4 Couche de Mise à Plat

La mise à plat, également appelée "flattening" en anglais, constitue la dernière étape de la section "extraction de caractéristiques". Elle implique simplement la concaténation de toutes les images (matrices) en un vecteur long. Les pixels, en réalité, ne représentent plus des images ni des pixels tels quels, mais plutôt des matrices de nombres. Ces matrices de nombres sont extraites ligne par ligne et incorporées dans le vecteur final comme illustré dans la Figure [2.5](#).

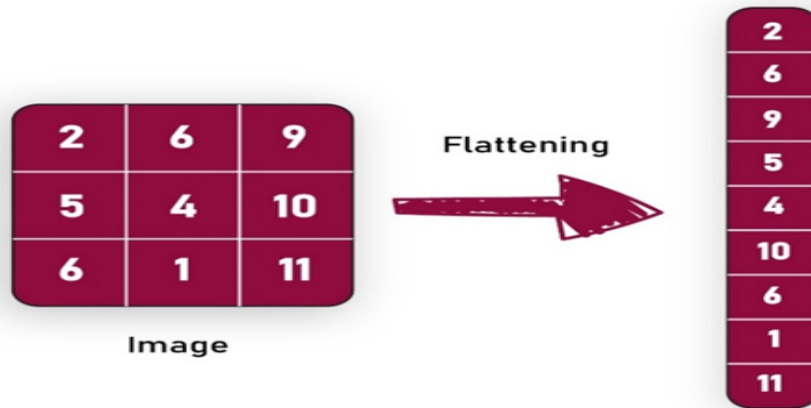


FIGURE 2.5 – Mise à plat.

### 2.1.5 La couche entièrement connectée

Dans les réseaux de neurones convolutionnels, la couche entièrement connectée est souvent utilisée à la fin du réseau pour effectuer la classification ou la régression sur les caractéristiques extraites par les couches de convolution précédentes. Elle prend un vecteur en entrée et produit un nouveau vecteur en sortie qui représente la probabilité d'appartenance d'une image à une classe. En général, la couche entièrement connectée consiste en un ensemble de neurones, où chaque neurone est relié à tous les neurones de la couche précédente. Chaque connexion entre les neurones est associée à un poids, qui indique l'importance de cette liaison. Chaque neurone est composé de deux fonctions : la pré-activation et l'activation, comme illustré dans la Figure [2.6](#).

- La fonction pré-activation est calculée comme suit :

$$Y = \sum_{i=1}^n (X_i \cdot W_i) + b \quad (2.5)$$

Où :  $X$  représente les caractéristiques d'entrée, généralement sous la forme d'un vecteur,  $W$  représente le poids de la connexion entre le neurone et  $b$  est le biais.

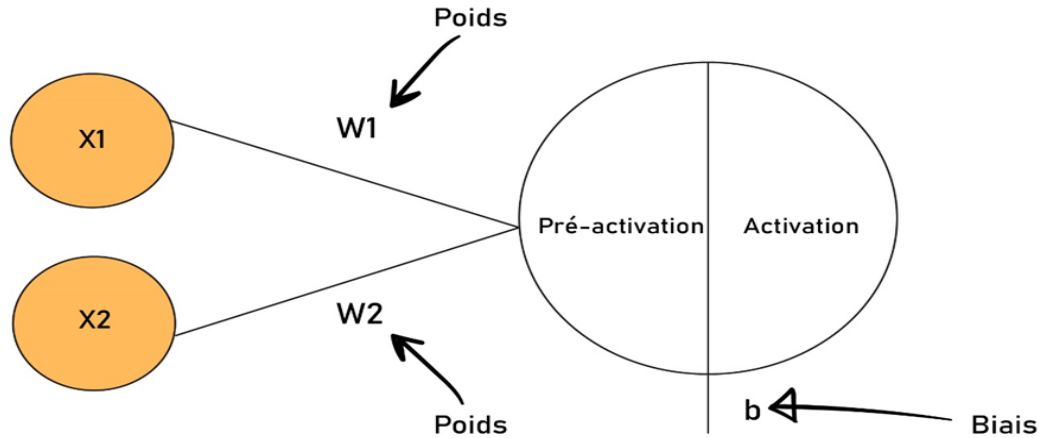


FIGURE 2.6 – Réseau neuronal à couche entièrement connectée.

- La fonction d'activation est calculée comme suit :

$$\text{Softmax}(X_i) = \frac{e^{X_i}}{\sum_{j=1}^N e^{X_j}} \quad (2.6)$$

Où :  $X$  représente le résultat de la pré-activation,  $N$  est le nombre total de neurones dans la couche entièrement connectée.

Diverses alternatives en termes de fonctions d'activation sont à disposition, leur choix dépendant du contexte et de l'application particulière du réseau neuronal. Néanmoins, la fonction Softmax est couramment préférée.

## 2.2 Optimisation et amélioration des CNNs

En complément des couches fondamentales précédemment mentionnées, d'autres éléments entrent en jeu et exercent un impact significatif sur le comportement ainsi que sur les performances des réseaux de neurones convolutifs (CNN). Parmi ces éléments

spécifiques, la normalisation par lot, également connue sous le nom de "Batch Normalization" en anglais, ainsi que le "Dropout", occupent une position particulière. Bien qu'elles ne participent pas directement à l'extraction des caractéristiques visuelles, Dan Hendrycks et Kevin Gimpel [20] ont démontré dans leur étude l'utilité de ces couches pour l'optimisation, la régularisation et l'amélioration globale des performances des CNN.

### 2.2.1 Normalisation par lot (Batch Normalization)

La technique de normalisation par lot a été élaborée dans le but d'accélérer la convergence de l'entraînement et de stabiliser les gradients. Cette méthode consiste à normaliser les activations en ajustant la moyenne et la variance des mini-lots pendant la phase d'entraînement. Cette approche prévient les déséquilibres internes et renforce la stabilité de l'apprentissage, ce qui se traduit par une convergence plus rapide et une capacité accrue à utiliser des taux d'apprentissage plus élevés. En uniformisant davantage les données d'entrée à travers chaque couche, la normalisation par lot simplifie également la formation de réseaux plus profonds. Cette technique assume un rôle crucial en accélérant le processus d'optimisation et en renforçant la résilience du modèle.

### 2.2.2 Dropout

Le "Dropout" est une technique visant à prévenir le surapprentissage en désactivant aléatoirement un certain pourcentage de neurones pendant l'entraînement. Cette approche crée des ensembles de sous-réseaux qui partagent les poids et les mises à jour. En évitant une spécialisation excessive des neurones, le "Dropout" favorise une généralisation plus solide. Cela conduit à la création de réseaux plus robustes, moins



sensibles aux variations mineures des données d'entraînement. L'introduction du "Dropout" renforce ainsi la capacité du modèle à se généraliser vers de nouvelles données, améliorant ses performances sur les ensembles de validation ou de test.

Cette technique reflète en quelque sorte l'apprentissage en ensemble. Pendant l'entraînement, des neurones sont aléatoirement désactivés selon un taux spécifié, ce qui les rend incapables d'apprendre à cette étape. Cette approche prévient le surapprentissage en éliminant la dépendance excessive envers certaines entrées ou neurones particuliers. Il est cependant crucial de ne pas fixer le taux à 1, sinon tous les neurones seraient désactivés, entravant ainsi la capacité du modèle à apprendre, comme illustré dans la Figure 2.7.

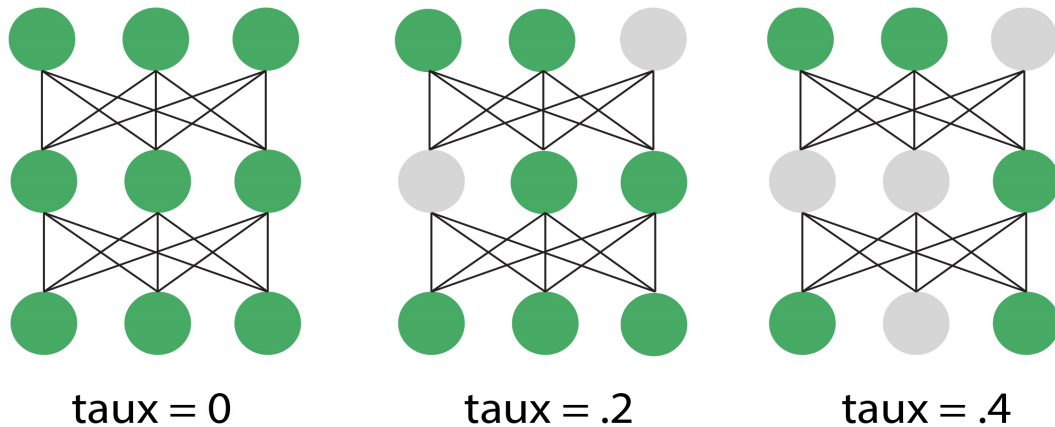


FIGURE 2.7 – Exploration du dropout dans les réseaux neuronaux.

## 2.3 Les CNNs profonds

Depuis leur introduction par LeCun et al [21] en 1989, la recherche sur les réseaux de neurones à convolution (CNN) demeure active et révèle un potentiel d'amélioration significatif. Au fil du temps, diverses améliorations ont été apportées à

l'architecture des CNN dans le but d'améliorer leurs performances. Ces améliorations sont directement liées à l'optimisation des paramètres, à l'application de méthodes de régularisation, à la restructuration de leur configuration, ainsi qu'à l'exploration d'autres avenues prometteuses. Néanmoins, il est clair que l'amélioration principale des performances des CNN découle de la restructuration des unités de traitement et de la conception de nouveaux blocs. En effet, la plupart des avancées dans les architectures des CNN se concentrent sur l'extension de leur profondeur ainsi que sur l'exploitation plus approfondie des caractéristiques spatiales.

L'étude réalisée par Asifullah Khan et son équipe [1] a proposé une classification globale en sept catégories distinctes pour les réseaux de neurones à convolution (CNN). Ces catégories englobent des aspects tels que l'exploitation spatiale, la profondeur, les trajets multiples, la largeur, l'exploitation des cartes de caractéristiques, le renforcement des canaux et les CNNs basés sur l'attention. Dans la Figure 2.8, les différentes architectures de CNN sont présentées en mettant en avant leurs catégories respectives. Pour une analyse approfondie de l'évolution des CNNs et de leurs architectures au fil du temps, il est recommandé de se plonger dans l'étude détaillée intitulée "A Survey of the Recent Architectures of Deep Convolutional Neural Networks", mentionnée dans la référence [1]. La recherche continue et l'engagement envers l'exploration d'innovations demeurent des éléments essentiels pour assurer le développement constant des Réseaux de Neurones Convolutifs (CNN) et pour leur application dans une variété de secteurs. De manière constante, de nouveaux défis se présentent et ouvrent la voie à l'exploration de nouvelles opportunités.

\* **N.B.** La Figure 2.8, extraite de l'article d'Asifullah Khan et al. [1], est considérée comme librement accessible au public, étant fréquemment utilisée dans les documentations liées aux CNNs profonds. Aucune modification n'a été apportée à cette figure, car elle contient suffisamment d'informations.

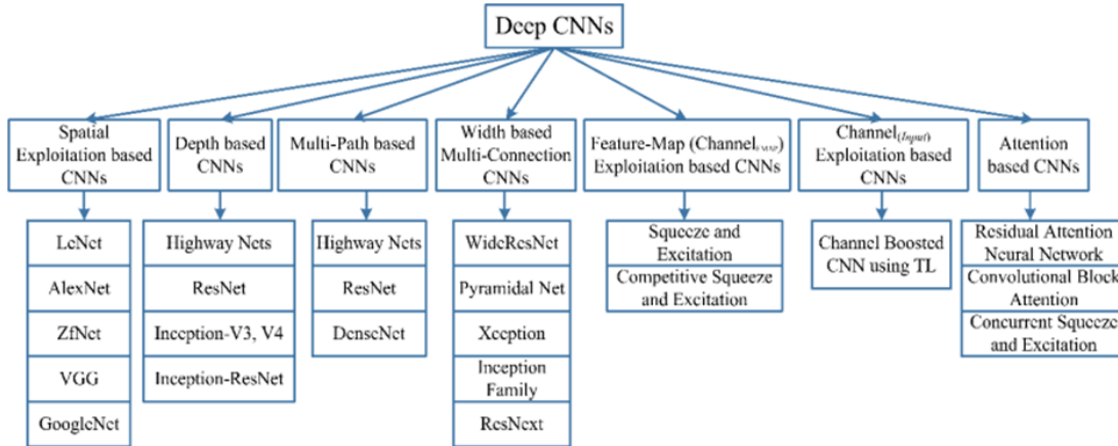


FIGURE 2.8 – Les diverses architectures profondes des CNNs [1].

## 2.4 Apprentissage par transfert

L'apprentissage par transfert, connu également sous le nom de Transfer Learning en anglais, présente une approche puissante en matière de résolution de problèmes. Il repose sur la capacité à exploiter les connaissances préalablement acquises lors de l'exécution de tâches antérieures similaires. Cette méthodologie est fondée sur l'idée que les compétences et les informations accumulées durant la résolution d'une tâche source peuvent être transférées et adaptées à une nouvelle tâche, même si cette dernière présente des caractéristiques différentes. Ce concept peut être comparé à la façon dont les êtres humains utilisent leurs expériences antérieures pour affronter de nouveaux défis. En appliquant les connaissances acquises précédemment à de nouveaux domaines qui présentent des similarités, l'apprentissage par transfert accélère l'acquisition de compétences et favorise une meilleure diffusion des connaissances.

Muhammad et al. [22] ont brillamment démontré l'efficacité de l'apprentissage par transfert en appliquant cette méthode à la formation et à l'évaluation de divers réseaux neuronaux convolutifs (CNN) préalablement entraînés sur la base de données Image-

Net. Ils ont ensuite utilisé ces connaissances acquises pour réaliser des prédictions précises sur des images radiographiques liées au COVID-19. Cette approche s'est révélée fructueuse, générant des résultats significatifs dans la prédiction des issues à partir des images radiographiques associées à cette maladie. Dans un contexte similaire, Wael Alhazmi et Turki Turki [23] ont également adopté l'apprentissage par transfert profond pour une investigation spécifique. Leur travail s'est concentré sur l'évaluation de l'impact des différentes modalités d'imagerie dans la détection du cancer du côlon. Les résultats obtenus ont été remarquablement performants, soulignant ainsi l'utilité et la puissance de l'approche de l'apprentissage par transfert dans des domaines médicaux cruciaux tels que celui-ci.

# Chapitre 3

## Support Vector Machine (SVM)

Les SVMs, également connues sous le nom de Machines à Vecteurs de Support, constituent une technique d'apprentissage supervisé qui a été développée en 1992 par Vapnik et ses collaborateurs [4]. Elles se sont avérées particulièrement efficaces dans diverses applications liées au traitement d'images. À titre d'exemple, Mohamad El-Abed et son équipe [24] ont élaboré un modèle SVM pour évaluer la qualité des données biométriques, obtenant des résultats prometteurs qui mettent en évidence les performances exceptionnelles des SVMs. De plus, dans le domaine de l'imagerie médicale, Tim Adams et ses collègues [5] ont déployé un système automatisé basé sur les SVM pour le diagnostic des tumeurs pulmonaires à partir de scans, atteignant une précision remarquable même avec un ensemble de données minimal.

Dans le contexte des SVMs (Support Vector Machines), l'objectif est de résoudre un problème de classification en utilisant un ensemble d'apprentissage constitué de paires instance-étiquette, où  $(x_i, y_i)$ ,  $i = 1, \dots, l$ , où  $x_i \in \mathbb{R}^n$  et  $y \in \{1, -1\}$ . Ici,  $x_i$  représente un vecteur de caractéristiques en  $n$  dimensions qui décrit chaque point de données, tandis que  $y_i$  est l'étiquette associée à  $x_i$ . Plus spécifiquement, dans le cadre des SVM, les étiquettes sont généralement définies comme suit : si  $x_i$  est positif,  $y_i$

est fixé à 1, sinon,  $y_i$  est fixé à -1. L'objectif principal des SVM est de trouver une fonction  $F(x)$  capable de prédire la valeur de  $y$  pour chaque instance  $x$  donnée. Dans les faits, les SVM cherchent à identifier une marge maximale entre deux classes. Pour simplifier, les SVM peuvent repérer un hyperplan qui sépare autant que possible les points de la même classe du même côté, comme cela est illustré dans la Figure 3.1, tout en maximisant la distance entre chaque classe et l'hyperplan.

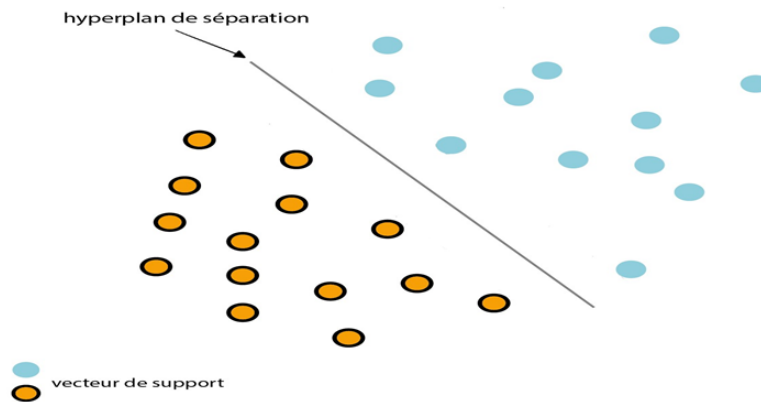


FIGURE 3.1 – L'hyperplan séparatif.

Dès leurs origines, les SVMs avaient pour objectif de réaliser une classification binaire, en séparant un ensemble de données en deux catégories distinctes. Dans ce contexte, leur utilisation prédominante consiste à résoudre des problèmes de classification comportant deux classes de sortie possibles, généralement notées comme +1 et -1, ou 1 et 0. De plus, on postulait qu'il devait exister une fonction linéaire capable d'accomplir cette tâche, définie comme suit :

$$F(x) = x^T \beta + \beta_0 = 0 \quad (3.1)$$

Dans l'équation (3.1),  $F(x)$  représente la fonction que nous cherchons à déterminer. L'hyperplan de séparation peut être caractérisé par son vecteur normal  $\beta$  et une constante  $\beta_0$ . Deux classes se trouvent de part et d'autre de l'hyperplan, ce qui nous permet d'utiliser cet hyperplan comme un critère de discrimination entre ces deux

classes. Dans chaque région, les points de données qui sont les plus proches de l'hyperplan sont appelés vecteurs de support. Ils revêtent une importance particulière dans l'ensemble d'apprentissage, car ce sont ces données qui sont utilisées exclusivement pour déterminer l'équation de l'hyperplan de séparation. Les hyperplans  $H_{p+}$  et  $H_{p-}$  sont parallèles à l'hyperplan de séparation et englobent les vecteurs de support. Ces hyperplans sont définis par :

$$H_{p+} : x_i^T \beta + \beta_0 = +1 \quad \text{pour tout } i, y_i = +1 \quad (3.2)$$

$$H_{p-} : x_i^T \beta + \beta_0 = -1 \quad \text{pour tout } i, y_i = -1 \quad (3.3)$$

La Figure 3.2 illustre ce que l'on appelle la frontière de décision, délimitant les régions classifiées en tant que positives et négatives par le classifieur. Les vecteurs de support sont marqués par des cercles, représentant les exemples les plus proches de cette frontière de décision. Ils déterminent la marge qui sépare les deux classes. Il est important de noter que chaque point de l'ensemble d'apprentissage se situe entre ces deux hyperplans. Ainsi, chaque donnée d'apprentissage satisfait aux conditions suivantes :

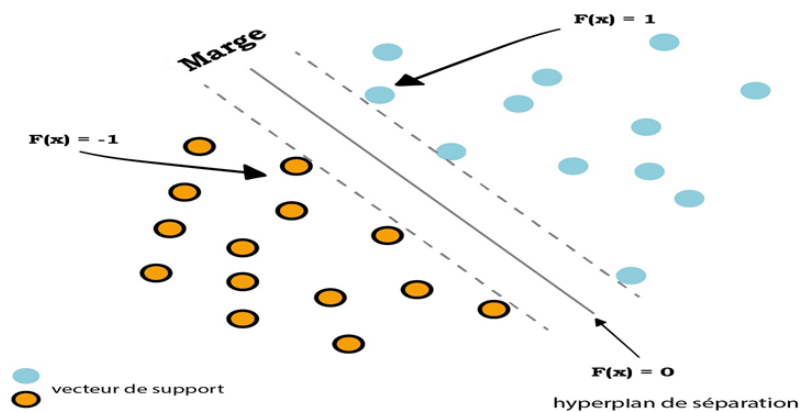


FIGURE 3.2 – Classes linéairement séparables.

$$F(x) = \begin{cases} y_i = +1 & \text{si } x_i^T \beta + \beta_0 \geq +1 \\ y_i = -1 & \text{si } x_i^T \beta + \beta_0 \leq -1 \end{cases} \quad (3.4)$$

Ces équations décrivent des hyperplans de séparation parallèles qui divisent les données et peuvent être formulées plus succinctement de la manière suivante :

$$y_i(w^T x_i + b) \geq 1 \quad \forall i \quad (3.5)$$

L'équation (3.5) énonce la contrainte fondamentale de SVM, exigeant que les données d'entraînement satisfaisant à cette condition. Cependant, cette contrainte, en soi, ne garantit pas l'existence d'une solution unique. En réalité, elle ouvre la porte à plusieurs solutions potentielles sous forme d'hyperplans, comme illustré dans la Figure 3.3. L'équation (3.5) pose ainsi un problème, ce qui signifie qu'il peut exister une multitude d'hyperplans qui satisfaisant cette contrainte. Pour résoudre ce problème, il est nécessaire de sélectionner le meilleur hyperplan parmi toutes les possibilités d'hyperplans séparateurs. Le critère de sélection est basé sur la maximisation de la marge. C'est là que la notion de marge entre en jeu. En favorisant une marge maximale, elle guide le processus vers la sélection d'une solution unique qui représente au mieux la séparation des données.

### 3.1 Maximisation de la marge

Pour sélectionner l'hyperplan optimal parmi ces hyperplans séparateurs, il est crucial de maximiser la marge. Pour calculer la distance entre l'hyperplan et le vecteur le plus proche qui représente la marge, il convient tout d'abord de calculer la distance entre  $H_{p+}$  et  $H_{p-}$  à partir de l'origine. La distance entre  $H_{p+}$  et  $H_{p-}$  est définie comme



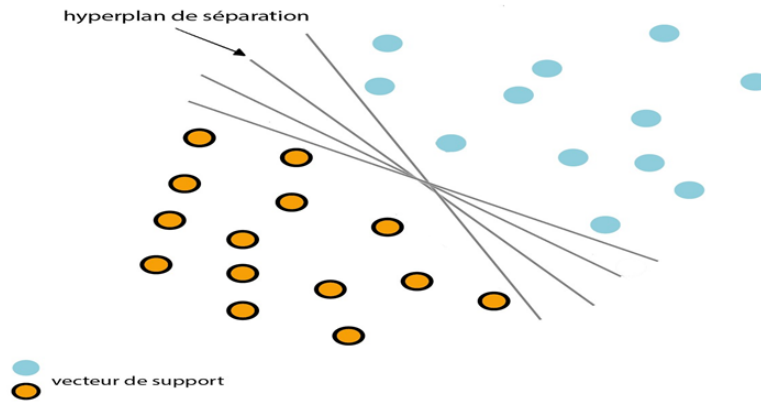


FIGURE 3.3 – Solutions possibles pour l'hyperplan séparatif.

suit :  $\frac{\beta_0+1}{\|\beta\|}$  et  $\frac{\beta_0-1}{\|\beta\|}$  respectivement. La marge est définie comme la distance entre  $H_{p+}$  et  $H_{p-}$  :

$$\text{Marge} = \frac{\beta_0 + 1}{\|\beta\|} - \frac{\beta_0 - 1}{\|\beta\|} = \frac{2}{\|\beta\|} \quad (3.6)$$

L'hyperplan optimal est celui qui permet d'obtenir la marge maximale tout en satisfaisant les contraintes de minimisation de la norme  $\|\beta\|^2$ . Ce processus conduit à la formulation d'un problème d'optimisation quadratique. Par conséquent, la recherche de la marge maximale se résume à la minimisation de la norme de  $\|\beta\|$ . En résumé, ce problème peut être exprimé comme une instance particulière de l'optimisation convexe, où l'objectif est de trouver l'hyperplan qui sépare les données de manière optimale en maximisant la marge tout en minimisant la norme de  $\|\beta\|$ .

$$\begin{aligned} &\text{Minimiser} && \frac{1}{2}\|\beta\|^2 \\ &\text{Sujet à} && y_i(\beta^T x_i + \beta_0) \geq 1 \quad \forall i \end{aligned} \quad (3.7)$$

La Figure [3.4](#) illustre le concept de marge, où les lignes en pointillés interrompues représentent les frontières de séparation définies par l'équation (3.4).

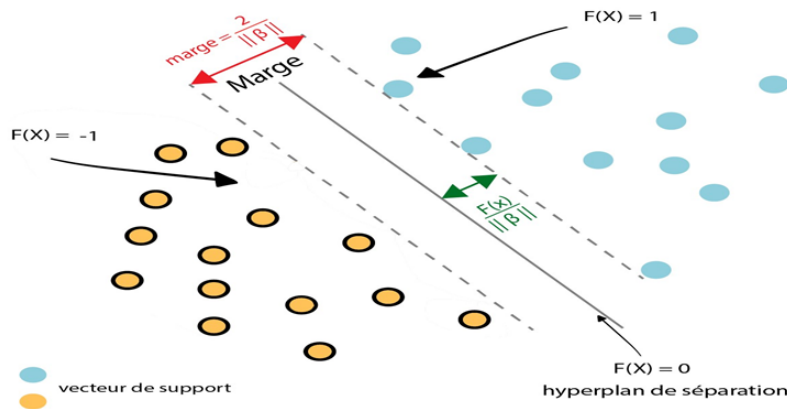


FIGURE 3.4 – Illustration de la notion de marge.

## 3.2 Soft margin SVM

Dans les cas où les données sont perturbées par du bruit et où il est impossible de séparer de manière parfaitement linéaire les différentes classes de données, comme illustré dans la Figure [3.5](#), la méthode préférée est la machine à vecteurs de support (SVM) à marge souple. Cette approche introduit ce que l'on appelle des "variables d'écart" (slack variables)  $\xi_i$ . Ces variables ont pour rôle de quantifier à quel point les vecteurs de caractéristiques sont mal classés. L'optimisation dans ce contexte implique un équilibre subtil entre deux objectifs : d'une part, maximiser la marge de séparation entre les classes, et d'autre part, minimiser le taux d'erreur de classification. Ce compromis est régi par un paramètre  $C$ , qui détermine comment l'optimisation contrainte est formulée, afin de s'adapter aux caractéristiques spécifiques des données. En ajustant la valeur de  $C$ , il devient possible de moduler l'importance accordée à la maximisation de la marge par rapport à la minimisation des erreurs de classification. Ainsi, l'optimisation contrainte peut être exprimée comme suit :

$$\begin{aligned}
 \text{Minimiser } & \frac{1}{2} \|\mathbf{W}\|^2 + C \sum_{i=1}^n \xi_i \\
 \text{Sujet à } & Y_i(w_0 + \mathbf{X}_i \mathbf{w}) \geq 1 - \xi_i \\
 & \xi_i \geq 0
 \end{aligned} \tag{3.8}$$

$n$  représente le nombre total de vecteurs d'entraînement.  $W$  est le vecteur de poids et  $w_0$  est le bias. Le paramètre  $C$  contrôle la balance entre la marge et les erreurs de classification lors de l'apprentissage. Un  $C$  plus grand permet de classifier plus d'observations correctement, mais avec une marge plus étroite. Un  $C$  plus petit produit une marge plus large, mais peut permettre davantage d'erreurs de classification.

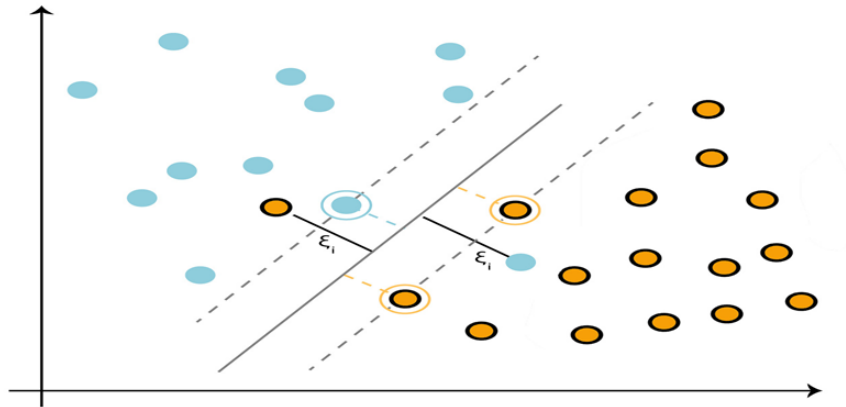


FIGURE 3.5 – *Soft margin SVM.*

### 3.3 Les noyaux

L'idée fondamentale derrière les SVMs non linéaires réside dans l'utilisation d'une transformation non linéaire appropriée pour projeter le problème dans un nouvel espace appelé l'espace des caractéristiques. Dans cet espace, un modèle linéaire peut être appliqué, ce qui équivaut à l'utilisation d'un modèle non linéaire dans l'espace d'entrée d'origine. Cette technique est communément désignée sous le nom de "Ker-

nel". Lorsque les données ne peuvent pas être séparées de manière linéaire dans l'espace des caractéristiques d'entrée, une fonction non linéaire  $\varphi(x)$  est employée pour projeter les points de données dans un espace de dimensions supérieures. Le Kernel consiste alors à remplacer le produit scalaire entre les points d'origine par le produit scalaire entre les points utilisant des noyaux (kernels). Cette approche transforme le problème non linéaire en un problème de classification linéaire, qui peut être résolu à l'aide de SVM. Comme illustré dans la Figure 3.6, les données d'origine ne sont pas linéairement séparables. Cependant, après l'application de la fonction  $\varphi$  grâce à cette astuce, elles deviennent linéairement séparables.

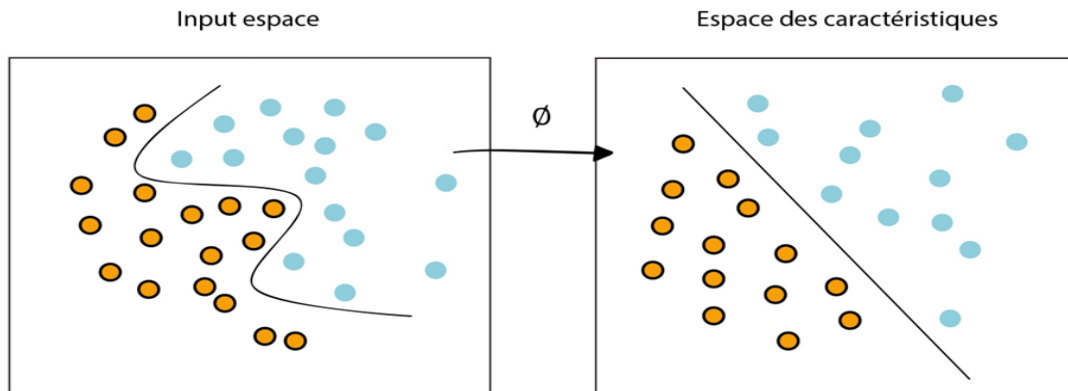


FIGURE 3.6 – Projection non-linéaire des classes.

Diverses fonctions de noyau sont accessibles dans le cadre de méthodes d'apprentissage automatique, parmi les plus fréquemment utilisées se trouvent :

$$\text{Polynomial} : K(x_i, x_j) = (\gamma x_i^T x_j + c)^d \quad (3.9)$$

$$\text{Linéaire} : K(x_i, x_j) = \tanh(x_i^T x_j + c) \quad (3.10)$$

$$\text{Noyau RBF} : K(x_i, x_j) = \exp\{-\gamma|x_i - x_j|^2\} \quad (3.11)$$

$$\text{Sigmoid} : K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \quad (3.12)$$

### 3.4 SVM vers une classification Multi-Classe

Tout comme la régression logistique, nous supposons que les valeurs de  $y$  appartiennent à un ensemble de classes allant de 0 à  $K - 1$ , avec  $K$  étant supérieur à 2. Que ce soit en linéaire ou non-linéaire, les SVMs permettent de passer du cas binaire au multi-classe par l'une des stratégies usuelles suivantes :

- **Un contre tous** : Cela implique la formation d'un SVM distinct pour chaque classe en la comparant aux autres classes, puis de calculer leur score et de prédire la classe ayant le score le plus élevé, comme illustré dans la Figure 3.7(b). Cette méthode nécessite l'entraînement de  $K$  classifieurs, où  $K$  correspond au nombre de classes.

- **Un contre un** : Cela implique l'entraînement d'un SVM distinct pour chaque paire de classes possible, puis de prédire avec chaque classifieur et de déterminer la classe la plus fréquemment prédite, comme illustré dans la Figure 3.7(a). Cette méthode nécessite  $K \cdot (K - 1)/2$  classifieurs, où  $K$  représente le nombre de classes. 1

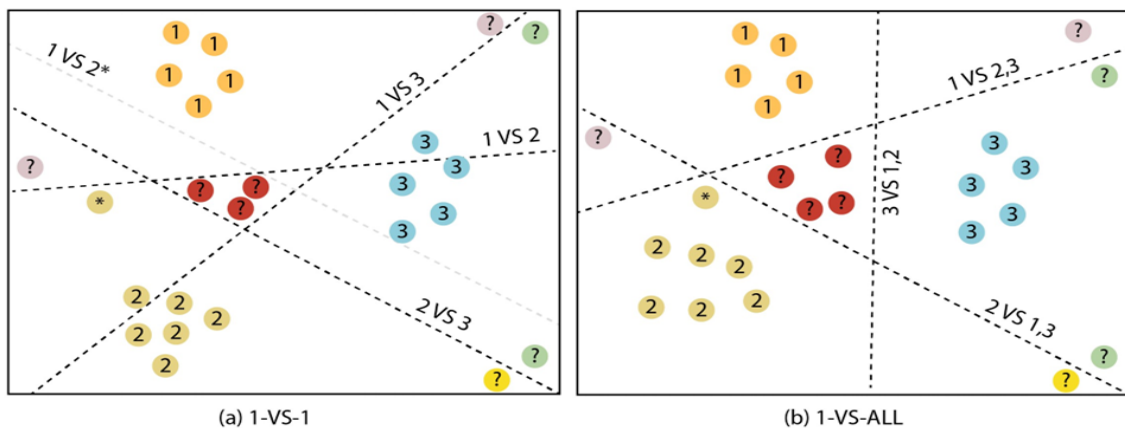


FIGURE 3.7 – Adaptation des SVMs à la classification multi-classe.

1. <https://openclassrooms.com/fr/SVM>

# Chapitre 4

## Hybride CNN-SVM

Les réseaux de neurones convolutifs (CNN) et les machines à vecteurs de support (SVM) représentent deux méthodes essentielles en apprentissage automatique, chacune offrant des avantages spécifiques pour aborder de manière efficace des tâches complexes de reconnaissance d'images. Les CNN sont aptes à traiter des images de diverses résolutions, tout en ayant la capacité d'apprendre des motifs de bas niveau (comme les contours) et de les combiner pour reconnaître des motifs de plus en plus complexes. En revanche, les SVM permettent d'identifier quelles caractéristiques (appelées vecteurs de support) contribuent le plus à la séparation entre les classes, ce qui s'avère précieux pour la prise de décision.

Récemment, de nombreuses recherches ont adopté une approche combinée pour tirer profit des avantages des deux applications mentionnées. Cette méthode intègre l'utilisation de réseaux de neurones convolutionnels (CNN) pour extraire les informations essentielles des données. Ensuite, la dernière couche des CNN, habituellement dédiée à la classification, est remplacée par un support vector machine (SVM) pour prendre en charge cette étape spécifique. Les résultats de ces recherches ont révélé des performances supérieures par rapport à l'utilisation indépendante de chaque ap-

plication. Cette méthode a fait ses preuves dans de multiples domaines, apportant une contribution significative à l'amélioration de la précision de la reconnaissance des caractères manuscrits, facilitant des avancées majeures en matière de sécurité et d'identification personnelle dans le domaine de la biométrie, et procurant d'importants avantages en biologie.

La conception générale de l'architecture CNN-SVM repose sur l'utilisation des réseaux de neurones convolutifs (CNN), des outils puissants pour extraire des caractéristiques significatives des données. Les couches convolutionnelles des CNNs jouent un rôle essentiel dans cette capacité, étant conçues pour capturer des caractéristiques hiérarchiques à différentes échelles en appliquant des filtres aux données d'entrée. Ces filtres identifient des motifs spécifiques dans différentes régions de l'image. Les sorties de la dernière couche, avant la couche de classification du CNN, sont collectées comme caractéristiques d'entrée pour le SVM, comme illustré dans la Figure [4.1](#). Ces caractéristiques d'entrée sont ensuite utilisées pour former un modèle SVM, qui cherche à trouver un hyperplan optimal pour séparer les différentes classes dans l'espace des caractéristiques. Plusieurs paramètres SVM, tels que le noyau, le terme de régularisation C et le paramètre Gamma, peuvent être ajustés pour optimiser les performances du modèle. Une fois que le modèle SVM est entraîné, il peut être évalué sur des données de test afin de mesurer ses performances en matière de classification.

- **La reconnaissance des caractères manuscrits**

Dans leur étude, Benkadja et son équipe [\[6\]](#) ont introduit un modèle hybride qui fusionne les capacités des réseaux de neurones convolutionnels (CNN) avec celles du classifieur SVM. Leurs travaux ont été réalisés en effectuant des expérimentations sur des données comprenant des chiffres et des lettres manuscrits, extraits de l'ensemble de données EMNIST. Les résultats obtenus ont démontré des performances de reconnaissance nettement supérieures par rapport à l'utilisation individuelle du CNN ou du SVM. De manière significative, l'équipe a également identifié que des ajustements

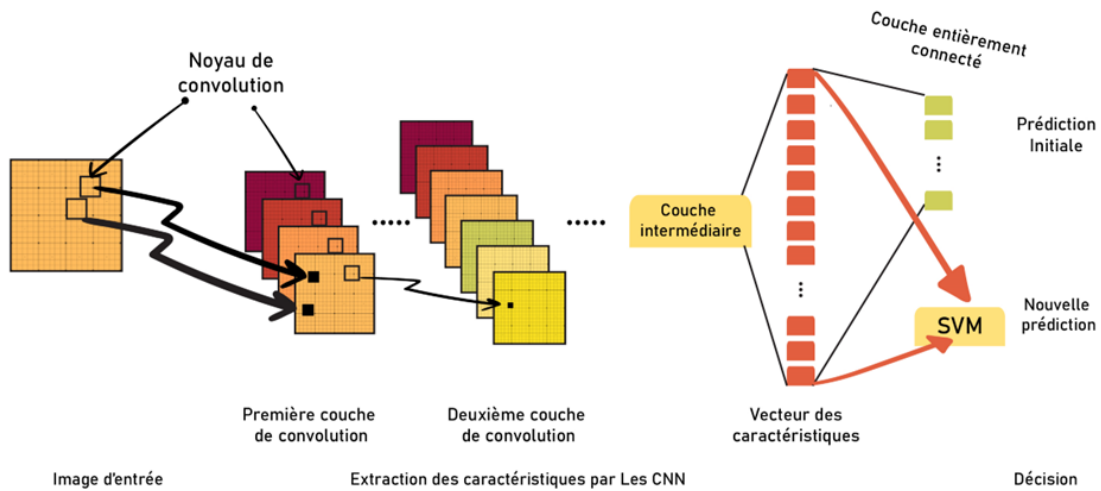


FIGURE 4.1 – Hybride CNN-SVM.

dans les paramètres du CNN et du SVM pouvaient conduire à une amélioration encore plus importante des taux de reconnaissance dans le cadre de cette approche hybride CNN-SVM.

Xiao-Xiao Niu et Ching Y. Sue [7] ont proposé un modèle hybride qui capitalise sur la synergie de deux classificateurs performants : les réseaux de neurones convolutionnels (CNN) et les machines à vecteurs de Support (SVM). Ces deux approches ont démontré des résultats convaincants dans la reconnaissance de divers motifs. Dans ce modèle, le CNN agit en tant qu'extracteur de caractéristiques adaptables, tandis que l'SVM joue le rôle de dispositif de reconnaissance. Ce modèle hybride extrait automatiquement des caractéristiques à partir des images brutes et génère des prédictions. Des expériences ont été menées sur la célèbre base de données MNIST, largement reconnue. Les comparaisons ont également été effectuées avec d'autres études portant sur la même base de données et utilisant le CNN et l'SVM de manière indépendante. Ces comparaisons ont montré que cette fusion a produit des résultats supérieurs : un taux de reconnaissance de 99,81 % sans rejet, et un taux de reconnaissance de 94,40 % avec un taux de rejet de 5,60 %. Ces performances ont été analysées en comparaison avec celles obtenues par d'autres chercheurs.



- **La biométrie**

La recherche menée par Serin J et ses collaborateurs [25] a présenté un modèle hybride CNN-SVM pour la classification du genre à partir d'empreintes digitales. Ce modèle se décompose en trois éléments essentiels : le prétraitement, l'extraction des caractéristiques et la phase de classification. L'objectif principal de cette étude était d'utiliser un réseau de neurones convolutifs (CNN) pour extraire des informations à partir des empreintes digitales. Par la suite, ces données ont été transmises à un classifieur SVM pour déterminer le genre des individus. Les performances de ce modèle hybride ont été évaluées et comparées à celles du modèle CNN classique. En utilisant le modèle hybride CNN-SVM, la précision de la classification du genre basée sur les empreintes digitales a atteint un taux de 99,25 %, tandis que le CNN en tant que tel a obtenu une précision de 98 %.

- **La biologie**

Yanjuan Wu et son équipe [26] ont mené une étude mettant en évidence l'importance des réseaux de neurones convolutifs (CNN) en tant que modèles d'apprentissage profond largement utilisés pour améliorer la productivité agricole grâce à l'inférence autonome des conditions sur le terrain. Dans leur recherche, ils ont également utilisé un CNN-SVM en combinant un CNN avec un SVM. Ils ont utilisé les architectures CNN de ResNet-50 et VGG16 pour créer les modèles correspondants, ResNet-50-SVM et VGG16-SVM. Leur méthodologie comportait deux étapes : l'extraction des caractéristiques avec ResNet-50 et VGG16, suivie de la classification avec SVM. Ils ont utilisé l'ensemble de données public DeepWeeds, comprenant plusieurs classes de mauvaises herbes, pour l'entraînement et les tests. Les approches ResNet-50-SVM et VGG16-SVM ont obtenu des taux de reconnaissance respectifs de 97,6 % et 95,9 % sur DeepWeeds, tandis que d'autres architectures de pointe ont atteint des précisions légèrement inférieures. En fin de compte, les méthodes proposées ont amélioré la précision de la classification des mauvaises herbes de 1,5 % et 2,7 %, montrant ainsi

leur efficacité dans ce contexte.

- **La santé**

Bien que cette approche ait connu un succès dans d'autres domaines, elle reste relativement peu courante dans le secteur de la santé, plus précisément en médecine. L'utilisation de cette approche soulève des interrogations quant à son efficacité, notamment lorsqu'elle est appliquée à des données complexes, telles que les images médicales, où les similitudes entre différentes images peuvent être très marquées. Par conséquent, il est essentiel de mener des recherches approfondies pour évaluer la pertinence et l'efficacité de cette méthode dans de telles situations. L'objectif de notre travail de recherche est de fournir des informations précieuses sur la meilleure manière d'adapter cette approche pour répondre aux besoins spécifiques de la communauté médicale, en améliorant le diagnostic et le traitement des patients, en particulier face à diverses situations liées à la COVID-19, et valider son utilisation dans ce secteur.

# Chapitre 5

## Méthodologie

Notre approche pour la classification d'images radiographiques consiste à utiliser des CNNs pré-entraînés pour extraire les fonctionnalités des images radiographiques et les classer à l'aide du classifieur SVM, voir Figure [5.1](#). Pour cela, nous commençons par améliorer la qualité de nos données. Ensuite, nous prenons les réseaux de neurones convolutionnels pré-entraînés, et nous les entraînons pour extraire les fonctionnalités de chaque classe. Cela nous permet de bénéficier des avantages des architectures de CNN préformés, qui peuvent extraire des caractéristiques complexes des images. Les sorties de la dernière couche de convolution sont alors aplaties en un vecteur unidimensionnel, ce qui permet d'obtenir un ensemble de fonctionnalités pour chaque image. Le vecteur extrait est ensuite utilisé comme entrée pour entraîner un classifieur SVM avec des paramètres spécifiquement choisis pour optimiser les performances du modèle dans le but d'effectuer la classification.

Dans ce chapitre, nous introduisons les divers ensembles de données que nous avons utilisés pour atteindre nos objectifs de recherche. Ensuite, nous examinerons en détail chaque étape de notre approche et expliquerons les avantages de celle-ci.

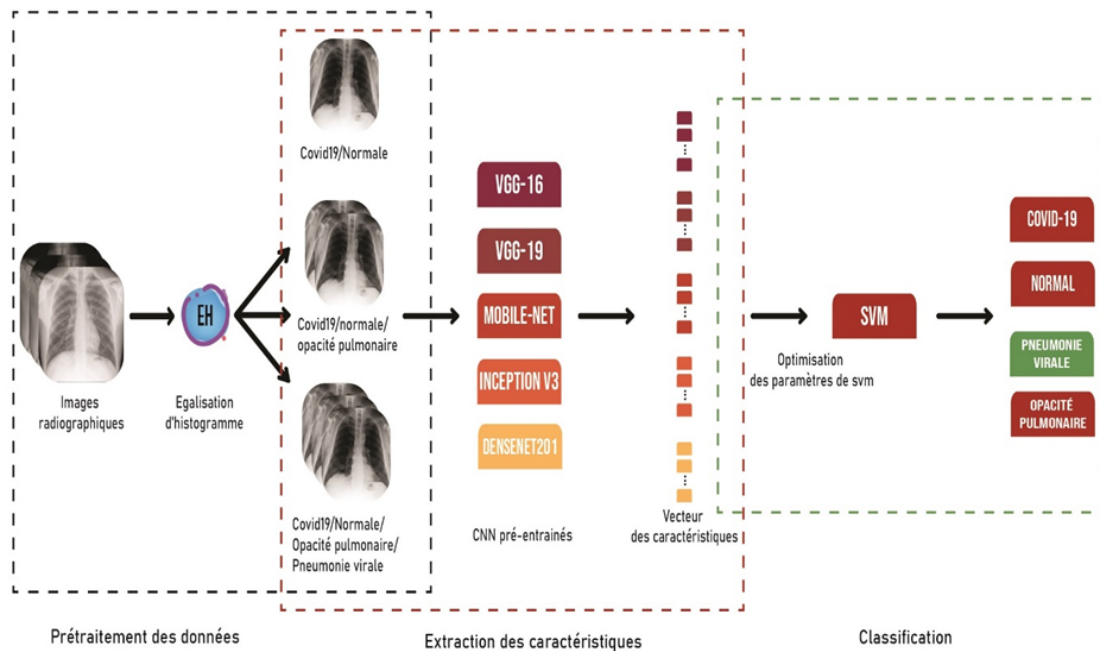


FIGURE 5.1 – Schéma bloc de la méthodologie.

## 5.1 Datasets

Le rôle majeur des données dans le domaine de l'apprentissage automatique met en lumière l'importance fondamentale du choix de l'ensemble de données. En optant pour une base de données préalablement vérifiée, appuyée par des travaux de recherches antérieures, les chercheurs s'assurent de l'utilisation de données à la fois représentatives et de haute qualité, notamment dans des secteurs comme la médecine, où la validation peut nécessiter des tests biologiques et chimiques pour garantir leur crédibilité. De plus, en partageant une base de données commune avec d'autres chercheurs, il devient possible de comparer équitablement les performances de différentes approches et techniques, ce qui revêt une importance essentielle pour évaluer l'efficacité de l'approche proposée en comparaison avec les méthodes existantes. Dans le cadre de notre travail de recherche, nous avons utilisé deux bases de données distinctes accessibles au public pour enrichir et valider notre approche.

### 5.1.1 COVID-19 Radiography Database

Une équipe de chercheurs de l'Université du Qatar, Doha, Qatar, et de l'Université de Dhaka, Bangladesh, ainsi que leurs partenaires du Pakistan et de Malaisie, en collaboration avec des médecins, [22],[27] ont créé une base de données d'images radiographiques pulmonaires pour les cas positifs au COVID-19 ainsi que des images de pneumonie normale et virale. Cette base de données a été construite en regroupant des informations provenant de diverses sources<sup>1</sup> :

- La base de données de la Société Italienne de Radiologie Médicale et Interventionnelle (SIRM) COVID-19 DATABASE [28] .

- Le jeu de données PadChest de la Banque d'images de la Région de Valence (BIMCV) est utilisé dans cette étude. La banque d'images PadChest contient une collection d'images radiographiques thoraciques provenant de patients avec diverses conditions pulmonaires, y compris des cas de COVID-19. Elle est principalement basée sur des images provenant de l'hôpital universitaire et polytechnique La Fe, situé dans la région de Valence en Espagne [29].

- Le jeu de données du nouveau coronavirus 2019 a été collecté par Joseph Paul Cohen, Paul Morrison et Lan Dao sur GitHub [30], et les images ont été extraites de 43 publications différentes.

- Finalement, afin de constituer une base de données d'images radiographiques complète comprenant des images normales, des opacités pulmonaires, des images de pneumonie virale et bactérienne, les chercheurs ont utilisé les bases de données "Images radiographiques thoraciques (pneumonie)" de Kaggle ainsi que celle de la Radiological Society of North America (RSNA) Kaggle [31],[32].

---

1. <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>.

Cet ensemble de données COVID-19, normal et autres infections pulmonaires est publié par étapes. Dans la première version, ils ont publié deux cent dix-neuf images COVID-19, mille trois cent quarante et un images normales et mille trois cent quarante-cinq images de radiographie pulmonaire de pneumonie virale (CXR). Dans la première mise à jour, ils ont augmenté la classe COVID-19 à mille deux cents images CXR. Dans la 2e mise à jour, ils ont augmenté la base de données à trois mille six cent seize cas positifs au COVID-19 ainsi qu'à dix mille cent quatre-vingt-douze images normales, six mille douze d'opacité pulmonaire (infection pulmonaire non-COVID) et 1345 images de pneumonie virale et masques pulmonaires correspondants. La Figure 5.2 présente des échantillons représentatifs de chaque classe de cette base de données.

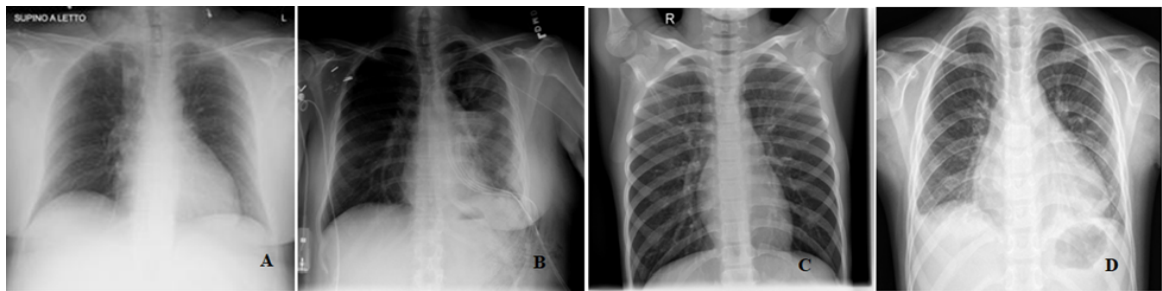


FIGURE 5.2 – Images radiographiques : COVID-19 (A), Opacité pulmonaire (B), Normale (C), et Pneumonie virale (D).

### 5.1.2 COVID-19 Pakistani Patients X-ray Image Dataset

L'équipe de chercheurs, provenant de l'université d'Édimbourg Napier au Royaume Uni, de l'université HITEC Taxila et de PNEC Karachi au Pakistan, en collaboration avec des experts médicaux, ont mis en place une base de données regroupant des radiographies thoraciques. cette base de données comprend des images de cas confirmés de covid-19 ainsi que des images représentant des situations thoraciques normales. au total, elle contient 390 images de covid-19 et 60 images de thorax normaux. la collecte de ces données a été supervisée par Muhammad Umair et ses collègues [11].

Nous comptons utiliser cette base de données comme référence pour évaluer la performance de notre approche, tout en mettant à l'épreuve sa robustesse face à des variations de données. Cette étape de validation nous permettra d'assurer que notre méthode peut maintenir un niveau de performance optimal dans divers scénarios, renforçant ainsi sa fiabilité et sa pertinence dans le domaine médical. Pour mieux illustrer la diversité de cette base de données, la Figure 5.3 présente des échantillons représentatifs de chaque catégorie. Vous pouvez accéder à cette base de données sur le site Kaggle.

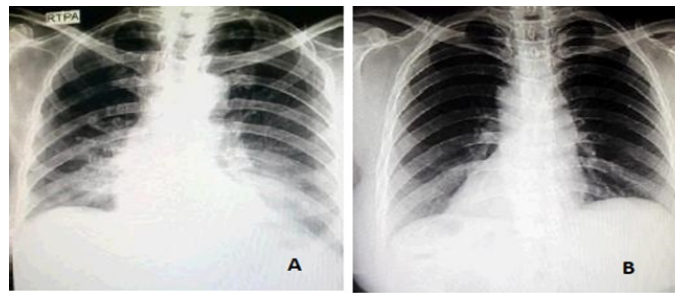


FIGURE 5.3 – Images radiographiques COVID-19 (A), Normale (B).

## 5.2 Prétraitement des données

Dans le domaine de l'apprentissage automatique, la qualité des données joue un rôle capital dans l'obtention de résultats optimaux. Par conséquent, il est impératif de procéder au prétraitement des données avant d'entamer la phase d'entraînement, afin de garantir que le modèle soit nourri avec des données pertinentes et de haute qualité. Le prétraitement des données constitue une technique essentielle qui vise à transformer les données brutes en un ensemble de données propre et significatif, créant ainsi les conditions nécessaires pour élaborer des modèles plus performants. L'importance du prétraitement des données réside dans son impact direct sur la qualité des résultats produits par le modèle d'apprentissage automatique.

En effet, des données de faible qualité ou mal préparées peuvent entraîner des performances insatisfaisantes, voire des erreurs lors des prédictions ou des classifications effectuées par le modèle. En revanche, lorsque les données sont soumises à un prétraitement approprié, le modèle est en mesure de généraliser plus efficacement à partir des exemples d'entraînement, d'atteindre des performances supérieures et de prendre des décisions plus précises, comme l'ont souligné Tawsifur Rahman et al. [27] dans leur publication où ils ont entraîné différents modèles des CNNs préentraînés sur l'ensemble de données radiographiques avant et après avoir appliqué des techniques de prétraitement visant à améliorer la qualité des données et à éliminer les informations indésirables. Cette amélioration des données a conduit à d'excellents résultats, contrairement à l'utilisation de données bruitées.

### 5.2.1 Égalisation d'histogramme

Parmi les diverses techniques d'amélioration des données étudiées par Tawsifur Rahman et son équipe [27], il est important de noter que l'application de l'égalisation d'histogramme a eu un impact significatif, entraînant une amélioration notable des performances et une meilleure précision dans les décisions prises. Cette technique fonctionne en ajustant la distribution des niveaux de gris de l'image, en étirant ou en comprimant l'histogramme afin d'optimiser l'utilisation de l'ensemble de la plage dynamique des niveaux de gris disponibles. Cela permet d'obtenir une distribution plus uniforme des niveaux de gris dans l'image, ce qui améliore la perception des détails et des contrastes dans l'image. Les Figures 5.4 et 5.5 illustrent la transformation des images de COVID-19 (A), d'opacité pulmonaire (B), normales (C) et de pneumonie virale (D) avant et après l'application de l'égalisation d'histogramme, respectivement.



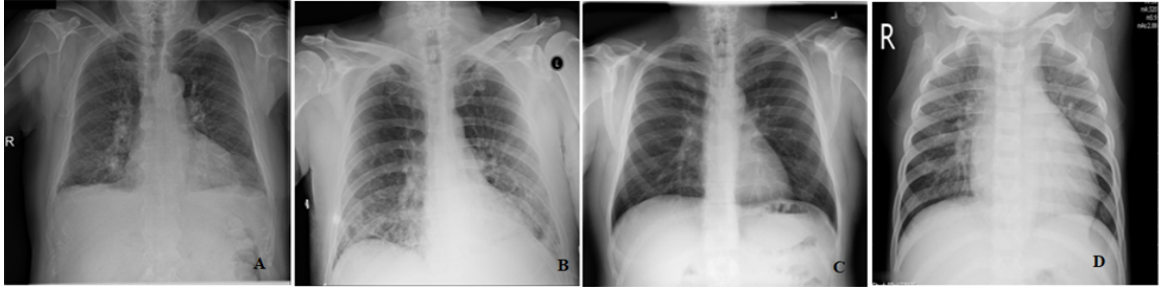


FIGURE 5.4 – Images radiographiques avant l'égalisation d'histogramme.

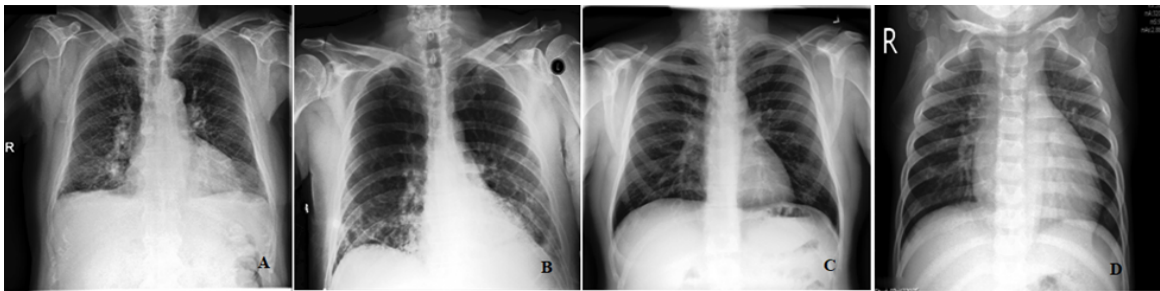


FIGURE 5.5 – Images radiographiques après l'égalisation d'histogramme.

Nous avons élaboré un tout nouveau jeu de données à partir de la base de données COVID-19 Radiography Database que nous avons présenté, en intégrant des images ayant subi une amélioration grâce à l'égalisation d'histogramme. Par la suite, ce jeu de données a été subdivisé en trois ensembles distincts : (COVID-19 vs. Normale), (COVID-19 vs. Normale vs. Opacité pulmonaire), et (COVID-19 vs. Normale vs. Opacité pulmonaire vs. Pneumonie virale), comme illustré dans la Figure 5.6. Nous avons utilisé ces ensembles de données pour entraîner et tester notre approche dans plusieurs scénarios. De plus, cette démarche a été entreprise dans le but d'évaluer les performances de notre méthode par rapport à celles d'autres chercheurs, dans des conditions similaires. Ces ensembles de données permettront une analyse comparative approfondie de l'efficacité de notre approche par rapport aux travaux antérieurs dans des contextes de recherche similaires.

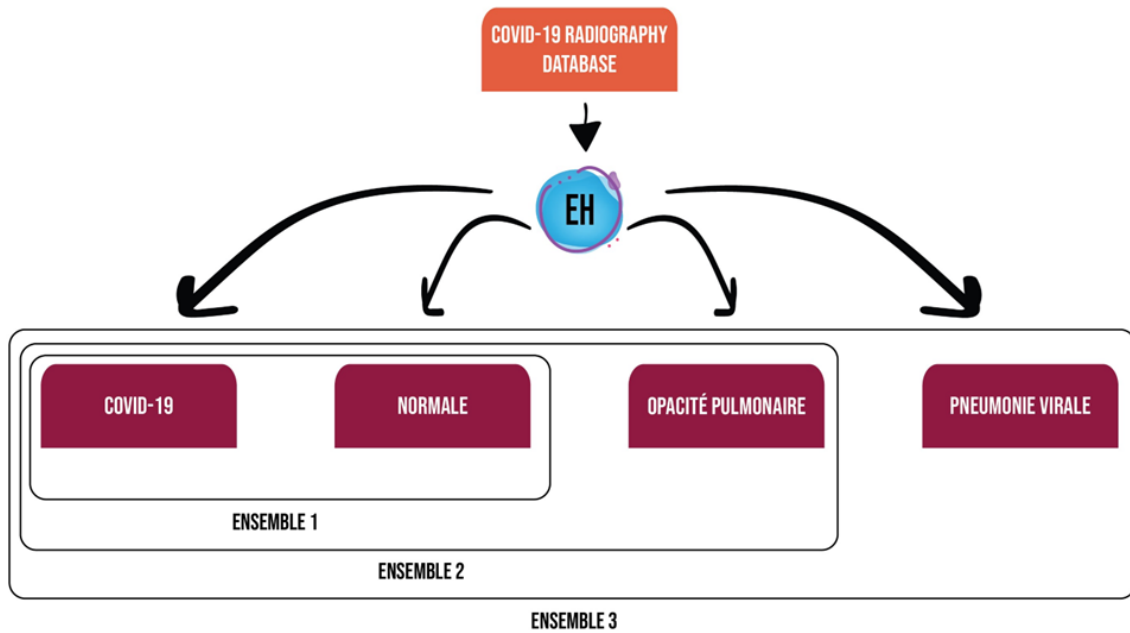


FIGURE 5.6 – Subdivision de nouvelles données améliorées.

Le tableau [5.1](#) présente une synthèse des données utilisées dans cette étude, en fournissant des détails sur la quantité d’images utilisées pour chaque classe dans chaque ensemble lors des phases d’entraînement et de validation.

TABLE 5.1 – Ensemble de données utilisées dans l’étude.

Cas	Entraînement	Validation	Total
COVID-19	3064	550	3614
Normale	9142	1050	10192
Opacité pulmonaire	5361	650	6011
Pneumonie virale	1123	200	1323

## 5.3 Extraction des fonctionnalités

En complément du prétraitement initial des images en entrée, l'extraction des caractéristiques des images joue un rôle essentiel dans la réussite de la classification. Le but de cette étape est de repérer les caractéristiques les plus spécifiques parmi les différentes classes de données tout en préservant simultanément les caractéristiques qui demeurent invariables au sein d'une même classe. Cette étape permet au modèle d'acquérir les informations essentielles pour effectuer la classification en identifiant les motifs significatifs dans les données d'entrée qui permettent de distinguer les différentes catégories. Une extraction performante des paramètres des images permet au modèle de développer des représentations de grande qualité, ce qui apporte une contribution importante à l'amélioration de ses performances dans le domaine de la classification.

### 5.3.1 Sélection des CNNs

Les CNNs profonds sont une technique d'apprentissage automatique adaptée à la reconnaissance d'images complexes. Les performances des réseaux de neurones convolutifs profonds dépendent en grande partie de la profondeur du réseau. En effet, une profondeur importante permet une représentation plus fine des caractéristiques de l'image et une meilleure capacité de généralisation. C'est pourquoi les réseaux de neurones convolutifs profonds sont souvent utilisés dans les applications médicales, notamment pour la détection des tumeurs cancéreuses ou du COVID-19 à partir de radiographies ou d'images médicales. Notre approche en matière de classification se base sur l'utilisation de ces réseaux neuronaux pré-entraînés comme extracteurs de caractéristiques, dans le but d'exploiter au maximum les informations extraites par ces réseaux profonds afin d'effectuer une classification plus précise et adaptée. Plus précisément, nous envisageons d'adapter leur architecture en effectuant une modifica-

tion ciblée, à savoir la substitution de la couche entièrement connectée de ces réseaux par un classifieur SVM.

Dans notre étude, nous avons utilisé cinq réseaux de neurones profonds pré-entraînés pour extraire les caractéristiques de nos données. Nous avons opté pour ces réseaux en raison de leur large adoption et de leur performance prouvée dans la littérature présentée dans le chapitre 1, notamment dans le contexte de l'analyse d'images médicales, en particulier pour les tâches de classification des images radiographiques liées au COVID-19. Les CNNs utilisés sont :

- **VGG16** modèle de réseau de neurones convolutifs (CNN) profond qui a été développé par l'équipe Visual Geometry Group (VGG) de l'Université d'Oxford en 2014. Ce modèle est reconnu pour sa simplicité architecturale, constituée de 16 couches de convolution et de pooling, d'où son nom "VGG16". Dans la Figure [5.7](#)<sup>\*</sup>, l'architecture détaillée de ce réseau est présentée en mettant en lumière les différentes couches utilisées ainsi que les filtres appliqués. Le vecteur en gras dans ce tableau représente la sortie finale aplatie de la phase d'extraction des caractéristiques, en particulier celui de la couche "**block5\_pool**" dans notre cas, avant de passer à la phase de classification via les couches entièrement connectées pour aboutir à la décision finale. Ce vecteur sera collecté pour être utilisé comme entrée pour notre classifieur SVM qui se chargera lui-même de cette tâche.

\* **N.B.** Les Figures [5.7](#), [5.9](#), [5.12](#), [5.14](#), [5.16](#) représentent des tableaux générés à partir d'un script Python. Nous les avons présentées sous forme de figures car nous avons des architectures qui dépassent 100 colonnes, ce qui rendrait difficile leur présentation sous forme de tableau. Nous avons préféré utiliser des figures avec des points de continuation afin de bien détailler notre idée.

Layer Name	Layer Type	Kernel Size	Stride	Padding	Output Shape
input_1	InputLayer		-	-	(224, 224, 3)
block1_conv1	Conv2D	(3, 3)	(1, 1)	same	(224, 224, 64)
block1_conv2	Conv2D	(3, 3)	(1, 1)	same	(224, 224, 64)
block1_pool	MaxPooling2D		(2, 2)	valid	(112, 112, 64)
block2_conv1	Conv2D	(3, 3)	(1, 1)	same	(112, 112, 128)
block2_conv2	Conv2D	(3, 3)	(1, 1)	same	(112, 112, 128)
block2_pool	MaxPooling2D		(2, 2)	valid	(56, 56, 128)
block3_conv1	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 256)
block3_conv2	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 256)
block3_conv3	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 256)
block3_pool	MaxPooling2D		(2, 2)	valid	(28, 28, 256)
block4_conv1	Conv2D	(3, 3)	(1, 1)	same	(28, 28, 512)
block4_conv2	Conv2D	(3, 3)	(1, 1)	same	(28, 28, 512)
block4_conv3	Conv2D	(3, 3)	(1, 1)	same	(28, 28, 512)
block4_pool	MaxPooling2D		(2, 2)	valid	(14, 14, 512)
block5_conv1	Conv2D	(3, 3)	(1, 1)	same	(14, 14, 512)
block5_conv2	Conv2D	(3, 3)	(1, 1)	same	(14, 14, 512)
block5_conv3	Conv2D	(3, 3)	(1, 1)	same	(14, 14, 512)
block5_pool	MaxPooling2D		(2, 2)	valid	(7, 7, 512)
<b>flatten</b>	<b>Flatten</b>		-	-	<b>(25088,)</b>
fc1	Dense		-	-	(4096,)
fc2	Dense		-	-	(4096,)
predictions	Dense		-	-	(1000,)

FIGURE 5.7 – Architecture du VGG16.

La nouvelle architecture du VGG16, suite à la substitution de la couche entièrement connectée par un classifieur SVM, est présentée dans la Figure 5.8.

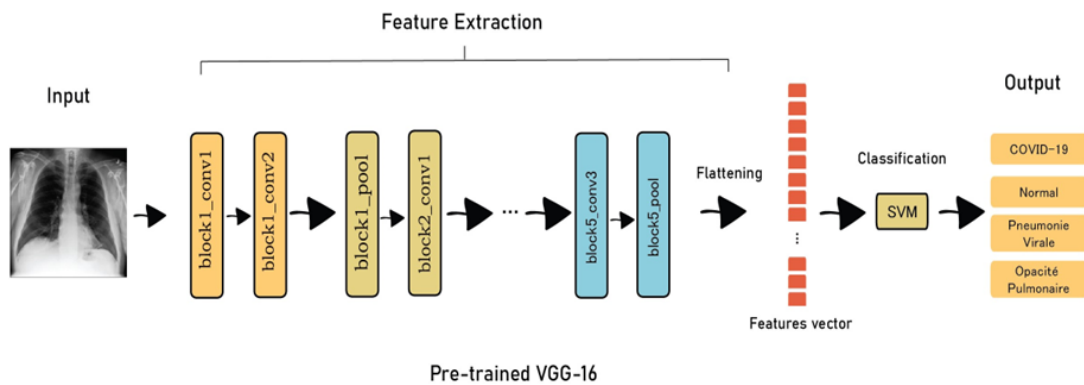


FIGURE 5.8 – Nouvelle architecture du VGG16.

• **VGG19**, créé par la même équipe de l’Université d’Oxford en 2014, est une extension de l’architecture VGG16. Il se distingue par sa profondeur accrue, composée de 19 couches de convolution et de pooling. Ce modèle se démarque par sa remarquable performance sur le jeu de données ImageNet et est réputé pour sa simplicité et sa facilité d’utilisation. Les modèles VGG16 et VGG19 partagent une architecture similaire jusqu’à la couche ”**block5\_pool**”, qui représente la dernière couche de la phase d’extraction des caractéristiques, comme illustré dans la Figure 5.9\*. La principale différence entre les vecteurs de caractéristiques de VGG16 et de VGG19 réside dans leur dimension, en raison du nombre de couches convolutives. En général, le vecteur de caractéristiques de VGG19 est légèrement plus grand que celui de VGG16, car il traverse un plus grand nombre de couches de convolution.

Layer Name	Layer Type	Kernel Size	Stride	Padding	Output Shape
input_5	InputLayer		-	-	(224, 224, 3)
block1_conv1	Conv2D	(3, 3)	(1, 1)	same	(224, 224, 64)
block1_conv2	Conv2D	(3, 3)	(1, 1)	same	(224, 224, 64)
block1_pool	MaxPooling2D		(2, 2)	valid	(112, 112, 64)
block2_conv1	Conv2D	(3, 3)	(1, 1)	same	(112, 112, 128)
block2_conv2	Conv2D	(3, 3)	(1, 1)	same	(112, 112, 128)
block2_pool	MaxPooling2D		(2, 2)	valid	(56, 56, 128)
block3_conv1	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 256)
block3_conv2	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 256)
block3_conv3	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 256)
block3_conv4	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 256)
block3_pool	MaxPooling2D		(2, 2)	valid	(28, 28, 256)
block4_conv1	Conv2D	(3, 3)	(1, 1)	same	(28, 28, 512)
block4_conv2	Conv2D	(3, 3)	(1, 1)	same	(28, 28, 512)
block4_conv3	Conv2D	(3, 3)	(1, 1)	same	(28, 28, 512)
block4_conv4	Conv2D	(3, 3)	(1, 1)	same	(28, 28, 512)
block4_pool	MaxPooling2D		(2, 2)	valid	(14, 14, 512)
block5_conv1	Conv2D	(3, 3)	(1, 1)	same	(14, 14, 512)
block5_conv2	Conv2D	(3, 3)	(1, 1)	same	(14, 14, 512)
block5_conv3	Conv2D	(3, 3)	(1, 1)	same	(14, 14, 512)
block5_conv4	Conv2D	(3, 3)	(1, 1)	same	(14, 14, 512)
block5_pool	MaxPooling2D		(2, 2)	valid	(7, 7, 512)
<b>flatten</b>	<b>Flatten</b>		-	-	<b>(25088,)</b>
fc1	Dense		-	-	(4096,)
fc2	Dense		-	-	(4096,)
predictions	Dense		-	-	(1000,)

FIGURE 5.9 – Architecture du VGG19.

La figure 5.10 illustre la nouvelle structure de l’architecture VGG19, qui a été modifiée en remplaçant la couche entièrement connectée par un classifieur SVM.

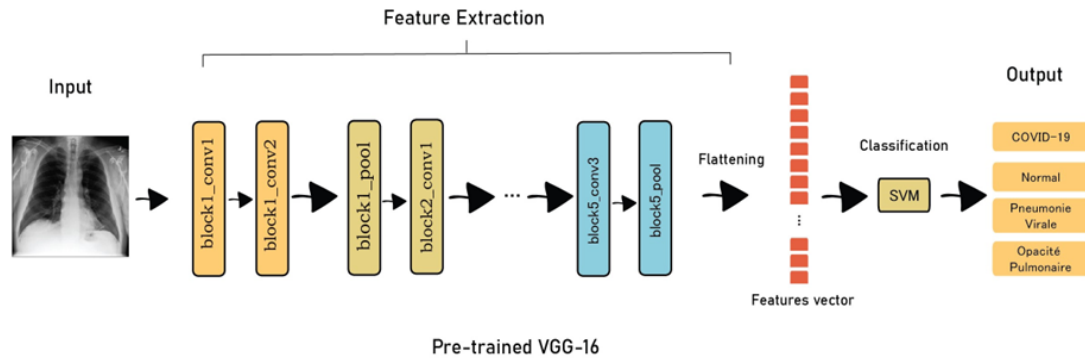
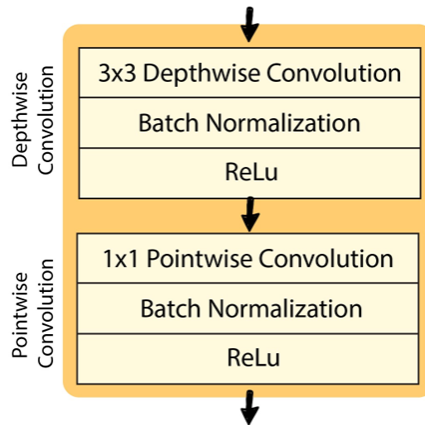


FIGURE 5.10 – Nouvelle architecture du VGG19.

• **MobileNet** introduit en 2017, famille de CNN spécialement conçue pour offrir légèreté et efficacité en termes de calcul. Cette caractéristique les rend parfaitement adaptés aux dispositifs mobiles et embarqués. L’architecture de MobileNet se distingue par l’utilisation de couches de convolution profondes employant une technique appelée **”convolution profonde séparable”**. Cette technique vise à diviser la convolution traditionnelle en deux étapes distinctes : la convolution profonde et la convolution spatiale, comme illustré dans la Figure 5.11, permettant de réduire significativement le nombre de paramètres et le coût de calcul par rapport aux architectures conventionnelles. Cette conception en fait une option idéale pour les applications de vision par ordinateur sur des dispositifs dotés de ressources limitées.

L’architecture MobileNet utilisée dans notre étude est composée de 88 couches. Les 83 premières couches sont dédiées à la phase d’extraction des caractéristiques. Les résultats de la couche 83 **”conv\_pw\_13\_relu”** représentent les dernières caractéristiques extraites de cette phase, présentées en format bidimensionnel, comme illustré dans la Figure 5.12\*. Les couches suivantes du réseau sont conçues pour minimiser la taille des caractéristiques extraites avant de les transmettre à la dernière couche entièrement connectée pour effectuer la prédiction et donner la décision finale. Dans le cadre de notre étude, nous avons extrait les résultats de la couche

FIGURE 5.11 – *Convolution profonde separable.*

`conv_pw_13_relu` sans aucune réduction de leur taille, afin de conserver toutes les informations extraites. Ensuite, nous les avons convertis en un vecteur unidimensionnel afin de les rendre compatibles avec notre classifieur SVM, qui remplace la couche entièrement connectée de ce réseau et occupe la phase de classification. La nouvelle structure du réseau est illustrée dans la Figure [5.13](#).

La principale différence entre les vecteurs de caractéristiques extraits par VGG16, VGG19 et MobileNet réside dans le fait que VGG16 et VGG19 génèrent généralement des vecteurs de caractéristiques plus volumineux et de dimensions plus élevées par rapport à MobileNet.



Layer Name	Layer Type	Kernel Size	Stride	Padding	Output Shape
input_4	InputLayer				(224, 224, 3)
conv1	Conv2D	(3, 3)	(2, 2)	same	(112, 112, 32)
conv1_bn	BatchNormalization				(112, 112, 32)
conv1_relu	ReLU				(112, 112, 32)
conv_dw_1	DepthwiseConv2D	(3, 3)	(1, 1)	same	(112, 112, 32)
conv_dw_1_bn	BatchNormalization				(112, 112, 32)
conv_dw_1_relu	ReLU				(112, 112, 32)
conv_pw_1	Conv2D	(1, 1)	(1, 1)	same	(112, 112, 64)
conv_pw_1_bn	BatchNormalization				(112, 112, 64)
conv_pw_1_relu	ReLU				(112, 112, 64)
conv_pad_2	ZeroPadding2D			((0, 1), (0, 1))	(113, 113, 64)
conv_dw_2	DepthwiseConv2D	(3, 3)	(2, 2)	valid	(56, 56, 64)
conv_dw_2_bn	BatchNormalization				(56, 56, 64)
conv_dw_2_relu	ReLU				(56, 56, 64)
conv_pw_2	Conv2D	(1, 1)	(1, 1)	same	(56, 56, 128)
conv_pw_2_bn	BatchNormalization				(56, 56, 128)
conv_pw_2_relu	ReLU				(56, 56, 128)
conv_dw_3	DepthwiseConv2D	(3, 3)	(1, 1)	same	(56, 56, 128)
conv_dw_3_bn	BatchNormalization				(56, 56, 128)
conv_dw_3_relu	ReLU				(56, 56, 128)
conv_pw_3	Conv2D	(1, 1)	(1, 1)	same	(56, 56, 128)
conv_pw_3_bn	BatchNormalization				(56, 56, 128)
conv_pw_3_relu	ReLU				(56, 56, 128)
conv_pad_4	ZeroPadding2D			((0, 1), (0, 1))	(57, 57, 128)
conv_dw_4	DepthwiseConv2D	(3, 3)	(2, 2)	valid	(28, 28, 128)
conv_dw_4_bn	BatchNormalization				(28, 28, 128)
conv_dw_4_relu	ReLU				(28, 28, 128)
conv_pw_4	Conv2D	(1, 1)	(1, 1)	same	(28, 28, 256)
conv_pw_4_bn	BatchNormalization				(28, 28, 256)
conv_pw_4_relu	ReLU				(28, 28, 256)
conv_dw_5	DepthwiseConv2D	(3, 3)	(1, 1)	same	(28, 28, 256)
conv_dw_5_bn	BatchNormalization				(28, 28, 256)
⋮	⋮	⋮	⋮	⋮	⋮
conv_pw_12	Conv2D	(1, 1)	(1, 1)	same	(7, 7, 1024)
conv_pw_12_bn	BatchNormalization				(7, 7, 1024)
conv_pw_12_relu	ReLU				(7, 7, 1024)
conv_dw_13	DepthwiseConv2D	(3, 3)	(1, 1)	same	(7, 7, 1024)
conv_dw_13_bn	BatchNormalization				(7, 7, 1024)
conv_dw_13_relu	ReLU				(7, 7, 1024)
conv_pw_13	Conv2D	(1, 1)	(1, 1)	same	(7, 7, 1024)
conv_pw_13_bn	BatchNormalization				(7, 7, 1024)
conv_pw_13_relu	ReLU				(7, 7, 1024)
global_average_poo...	GlobalAveragePooli...				(1, 1, 1024)
dropout	Dropout				(1, 1, 1024)
conv_preds	Conv2D	(1, 1)	(1, 1)	same	(1, 1, 1000)
reshape_2	Reshape				(1000,)
predictions	Activation				(1000,)

FIGURE 5.12 – Architecture du MobileNet.

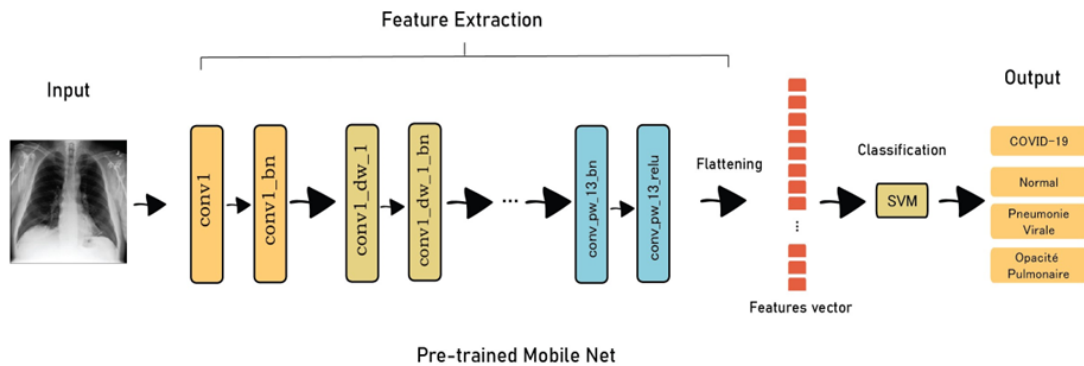


FIGURE 5.13 – Nouvelle architecture du MobileNet.

• **Inception-V3** appelé GoogLeNetV3, modèle d'architecture CNN développé par Google en 2015. Sa particularité réside dans l'utilisation de modules Inception, qui intègrent simultanément des filtres de convolution de différentes tailles, permettant au réseau de saisir des informations à diverses échelles. InceptionV3 est célèbre pour ses performances remarquables sur le jeu de données ImageNet et sa capacité à extraire des caractéristiques complexes de manière efficace. La dernière couche d'extraction de caractéristiques dans le modèle InceptionV3 correspond aux sorties de la couche "mixed10". Cette couche joue un rôle de regroupement en consolidant les différentes caractéristiques extraites par les modules d'Inception, qui sont responsables de la capture d'informations à différentes échelles spatiales. Concrètement, la couche **mixed10** fusionne ces caractéristiques diverses en une seule sortie, qui est ensuite transmise à la dernière couche du réseau chargée de la classification, comme illustré dans la Figure 5.14\* du modèle.

Les résultats obtenus à partir de cette couche sont généralement de grande dimension, ce qui signifie qu'ils contiennent de nombreuses valeurs. Pour gérer cette dimension élevée, nous avons utilisé une couche de regroupement (pooling) pour réduire la taille de ce vecteur, puis nous l'aplatissons en une forme unidimensionnelle. Ce vecteur résultant est ensuite transmis à l'SVM qui va prendre en charge la tâche de classification finale. L'architecture du réseau Inception-V3 après les modifications est présentée dans la Figure 5.15.

Layer Name	Layer Type	Kernel Size	Stride	Padding	Output Shape
input_12	InputLayer				(299, 299, 3)
conv2d_188	Conv2D	(3, 3)	(2, 2)	valid	(149, 149, 32)
batch_normalization_188	BatchNormalization				(149, 149, 32)
activation_188	Activation				(149, 149, 32)
conv2d_189	Conv2D	(3, 3)	(1, 1)	valid	(147, 147, 32)
batch_normalization_189	BatchNormalization				(147, 147, 32)
activation_189	Activation				(147, 147, 32)
conv2d_190	Conv2D	(3, 3)	(1, 1)	same	(147, 147, 64)
batch_normalization_190	BatchNormalization				(147, 147, 64)
activation_190	Activation				(147, 147, 64)
max_pooling2d_8	MaxPooling2D		(2, 2)	valid	(73, 73, 64)
conv2d_191	Conv2D	(1, 1)	(1, 1)	valid	(73, 73, 80)
batch_normalization_191	BatchNormalization				(73, 73, 80)
activation_191	Activation				(73, 73, 80)
conv2d_192	Conv2D	(3, 3)	(1, 1)	valid	(71, 71, 192)
batch_normalization_192	BatchNormalization				(71, 71, 192)
activation_192	Activation				(71, 71, 192)
max_pooling2d_9	MaxPooling2D		(2, 2)	valid	(35, 35, 192)
conv2d_196	Conv2D	(1, 1)	(1, 1)	same	(35, 35, 64)
batch_normalization_196	BatchNormalization				(35, 35, 64)
activation_196	Activation				(35, 35, 64)
conv2d_194	Conv2D	(1, 1)	(1, 1)	same	(35, 35, 48)
conv2d_197	Conv2D	(3, 3)	(1, 1)	same	(35, 35, 96)
batch_normalization_194	BatchNormalization				(35, 35, 48)
batch_normalization_197	BatchNormalization				(35, 35, 96)
activation_194	Activation				(35, 35, 48)
activation_197	Activation				(35, 35, 96)
average_pooling2d_18	AveragePooling2D		(1, 1)	same	(35, 35, 192)
conv2d_193	Conv2D	(1, 1)	(1, 1)	same	(35, 35, 64)
conv2d_195	Conv2D	(5, 5)	(1, 1)	same	(35, 35, 64)
conv2d_198	Conv2D	(3, 3)	(1, 1)	same	(35, 35, 96)
conv2d_199	Conv2D	(1, 1)	(1, 1)	same	(35, 35, 32)
batch_normalization_193	BatchNormalization				(35, 35, 64)
batch_normalization_195	BatchNormalization				(35, 35, 64)
⋮	⋮	⋮	⋮	⋮	⋮
conv2d_281	Conv2D	(1, 1)	(1, 1)	same	(8, 8, 192)
batch_normalization_273	BatchNormalization				(8, 8, 320)
activation_275	Activation				(8, 8, 384)
activation_276	Activation				(8, 8, 384)
activation_279	Activation				(8, 8, 384)
batch_normalization_280	BatchNormalization				(8, 8, 384)
batch_normalization_281	BatchNormalization				(8, 8, 192)
activation_273	Activation				(8, 8, 320)
mixed9_1	Concatenate				(8, 8, 768)
concatenate_5	Concatenate				(8, 8, 768)
activation_281	Activation				(8, 8, 192)
mixed10	Concatenate				(8, 8, 2048)
avg_pool	GlobalAveragePooling2D				(2048,)
predictions	Dense				(1000,)

FIGURE 5.14 – Architecture du Inception-V3.

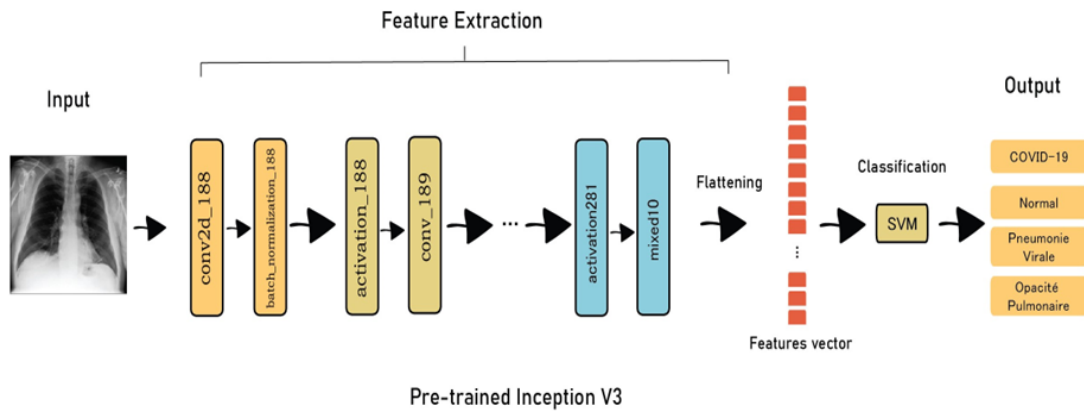


FIGURE 5.15 – Nouvelle architecture du Inception-V3.

- **DenseNet201** réseau de neurones convolutifs basé sur l'architecture DenseNet, développé en 2016, et il est composé de 201 couches. DenseNet se distingue par son concept d'architecture dense, où chaque couche est connectée à toutes les couches précédentes, favorisant ainsi le partage d'informations et la réutilisation des caractéristiques extraites. Cette conception unique contribue à réduire le risque de disparition du gradient et à minimiser le nombre de paramètres par rapport aux architectures traditionnelles. En conséquence, DenseNet201 démontre des performances élevées dans une gamme variée de tâches de vision par ordinateur.

La dernière couche de la phase d'extraction des caractéristiques d'un modèle DenseNet201 est généralement la couche de sortie **relu**, qui est le résultat de la dernière unité du bloc dense "conv5\_bloc\_32\_concat" comme illustré dans la Figure 5.16\*. Dans le modèle DenseNet, les blocs sont composés de plusieurs couches de convolution, de normalisation et d'activation. La sortie de chaque bloc est concaténée avec les sorties précédentes pour former les caractéristiques en profondeur. Les résultats de cette couche seront normalisés, puis activés, et enfin aplatis pour les transformer en un vecteur d'une seule dimension, ce qui le rend compatible avec le SVM qui s'occupera de la phase de classification finale. La nouvelle architecture de DenseNet201, après la substitution de la couche entièrement connectée, est présentée dans la Figure 5.17.

Pour obtenir plus de détails sur les différents CNN profonds ainsi que sur leur architecture et leur développement au fil du temps, nous vous invitons à consulter l'article de Nikolas Adaloglou référencé [33], qui présente tous les CNNs profonds disponibles jusqu'à aujourd'hui.

Layer Name	Layer Type	Kernel Size	Stride	Padding	Output Shape
input_9	InputLayer	-	-	-	(224, 224, 3)
zero_padding2d_6	ZeroPadding2D	-	-	((3, 3), (3, 3))	(230, 230, 3)
conv1/conv	Conv2D	(7, 7)	(2, 2)	valid	(112, 112, 64)
conv1/bn	BatchNormalization	-	-	-	(112, 112, 64)
conv1/relu	Activation	-	-	-	(112, 112, 64)
zero_padding2d_7	ZeroPadding2D	-	-	((1, 1), (1, 1))	(114, 114, 64)
pool1	MaxPooling2D	-	(2, 2)	valid	(56, 56, 64)
conv2_block1_0_bn	BatchNormalization	-	-	-	(56, 56, 64)
conv2_block1_0_relu	Activation	-	-	-	(56, 56, 64)
conv2_block1_1_conv	Conv2D	(1, 1)	(1, 1)	valid	(56, 56, 128)
conv2_block1_1_bn	BatchNormalization	-	-	-	(56, 56, 128)
conv2_block1_1_relu	Activation	-	-	-	(56, 56, 128)
conv2_block1_2_conv	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 32)
conv2_block1_concat	Concatenate	-	-	-	(56, 56, 96)
conv2_block2_0_bn	BatchNormalization	-	-	-	(56, 56, 96)
conv2_block2_0_relu	Activation	-	-	-	(56, 56, 96)
conv2_block2_1_conv	Conv2D	(1, 1)	(1, 1)	valid	(56, 56, 128)
conv2_block2_1_bn	BatchNormalization	-	-	-	(56, 56, 128)
conv2_block2_1_relu	Activation	-	-	-	(56, 56, 128)
conv2_block2_2_conv	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 32)
conv2_block2_concat	Concatenate	-	-	-	(56, 56, 128)
conv2_block3_0_bn	BatchNormalization	-	-	-	(56, 56, 128)
conv2_block3_0_relu	Activation	-	-	-	(56, 56, 128)
conv2_block3_1_conv	Conv2D	(1, 1)	(1, 1)	valid	(56, 56, 128)
conv2_block3_1_bn	BatchNormalization	-	-	-	(56, 56, 128)
conv2_block3_1_relu	Activation	-	-	-	(56, 56, 128)
conv2_block3_2_conv	Conv2D	(3, 3)	(1, 1)	same	(56, 56, 32)
conv2_block3_concat	Concatenate	-	-	-	(56, 56, 160)
conv2_block4_0_bn	BatchNormalization	-	-	-	(56, 56, 160)
conv2_block4_0_relu	Activation	-	-	-	(56, 56, 160)
conv2_block4_1_conv	Conv2D	(1, 1)	(1, 1)	valid	(56, 56, 128)
⋮	⋮	⋮	⋮	⋮	⋮
conv5_block31_2_conv	Conv2D	(3, 3)	(1, 1)	same	(7, 7, 32)
conv5_block31_concat	Concatenate	-	-	-	(7, 7, 1888)
conv5_block32_0_bn	BatchNormalization	-	-	-	(7, 7, 1888)
conv5_block32_0_relu	Activation	-	-	-	(7, 7, 1888)
conv5_block32_1_conv	Conv2D	(1, 1)	(1, 1)	valid	(7, 7, 128)
conv5_block32_1_bn	BatchNormalization	-	-	-	(7, 7, 128)
conv5_block32_1_relu	Activation	-	-	-	(7, 7, 128)
conv5_block32_2_conv	Conv2D	(3, 3)	(1, 1)	same	(7, 7, 32)
conv5_block32_concat	Concatenate	-	-	-	(7, 7, 1920)
bn	BatchNormalization	-	-	-	(7, 7, 1920)
relu	Activation	-	-	-	(7, 7, 1920)
avg_pool	GlobalAveragePooli...	-	-	-	(1920,)
predictions	Dense	-	-	-	(1000,)

FIGURE 5.16 – Architecture du DenseNet201.

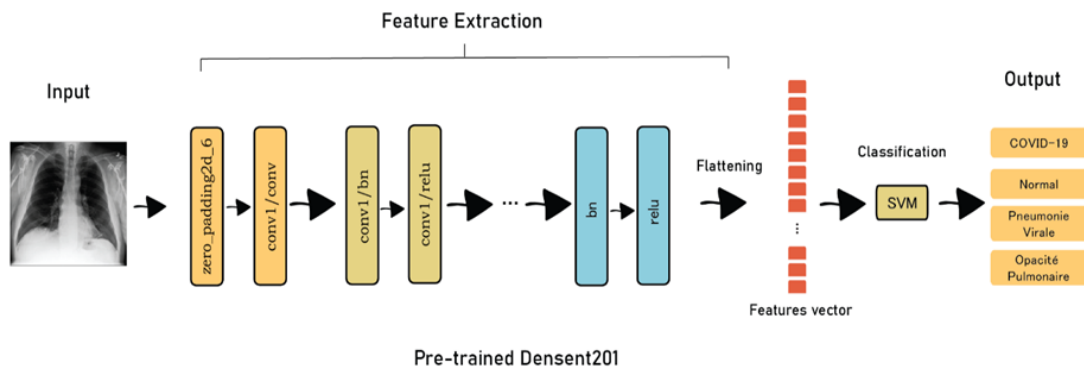


FIGURE 5.17 – Nouvelle architecture du DenseNet201.

### 5.3.2 Extraction de caractéristiques via les CNNs

Après avoir sélectionné les CNNs, nous avons entamé la phase d'extraction des caractéristiques sur notre jeu de données de radiographies COVID-19 en adaptant d'abord nos données aux structures des réseaux de neurones préentraînés choisis pour notre étude. Plus précisément, nous avons redimensionné nos images à une résolution de 224x224 pixels pour les modèles VGG16, VGG19, MobileNet et DenseNet201, tandis que le modèle Inception-V3 a été configuré avec une résolution de 299x299 pixels pour être compatible avec son architecture initiale. Ces modèles ont été entraînés pour la tâche d'extraction des caractéristiques en utilisant des lots de 32 images (appelés "batch size" en anglais), ce qui signifie que 32 images étaient traitées simultanément pour mettre à jour les poids du modèle lors de l'entraînement.

L'examen visuel des couches de convolution au sein des modèles revêt une grande importance. En visualisant ces couches, nous pouvons mieux comprendre les informations que le modèle tente d'extraire et de mettre en évidence, ce qui nous permet de mieux comprendre son fonctionnement. Chaque couche du réseau est dédiée à l'apprentissage de motifs et de détails spécifiques présents dans l'image. La Figure [5.18](#) illustre de manière significative la transformation subie par une image d'entrée lorsqu'elle traverse les différentes couches des CNNs choisis dans l'étude, notamment la première couche de chaque réseau. Ces couches initiales extraient des caractéristiques de bas niveau, qui sont ensuite utilisées pour reconnaître des éléments de haut niveau dans les couches plus profondes. Ces caractéristiques sont essentiellement des représentations abstraites de l'image d'entrée, devenant progressivement plus complexes à mesure que l'on progresse en profondeur dans le réseau.

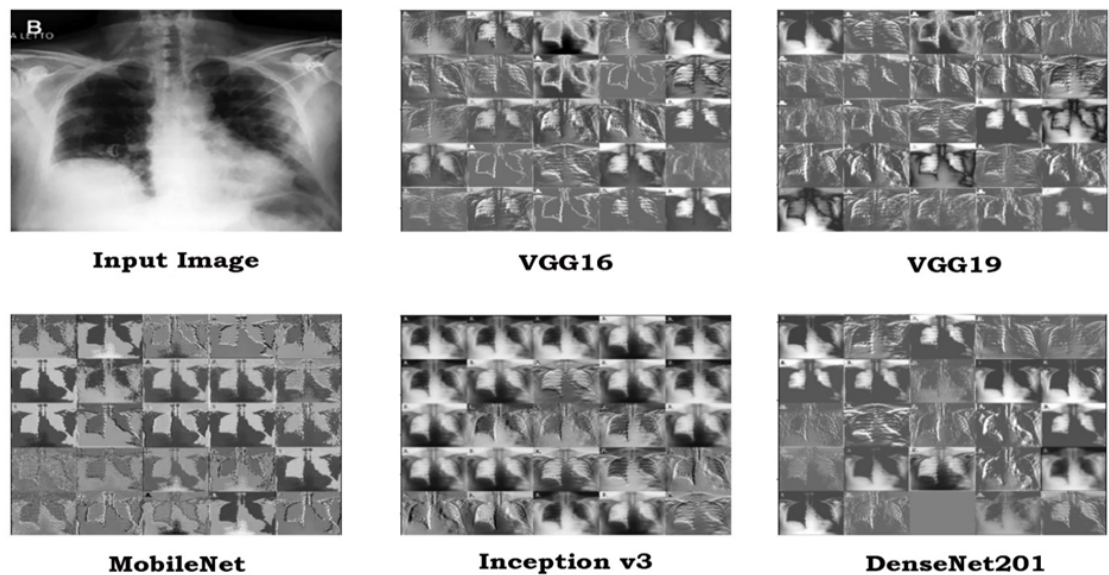


FIGURE 5.18 – Visualisation des premières couches de convolution.

## 5.4 Optimisation des paramètres pour SVM

Les SVM s'appuient sur une variété de paramètres qui influent sur le comportement du modèle au cours de la phase d'apprentissage, comme décrit dans le chapitre 3. Il est essentiel de bien ajuster ces paramètres avant de lancer la phase d'entraînement afin de garantir des performances de prise de décision de haute qualité. Ces paramètres doivent être adaptés en fonction de la nature des données, En particulier, le vecteur de caractéristiques extrait par les CNNs dans notre contexte. Les trois principaux paramètres qui font l'objet d'une optimisation sont le noyau , le paramètre de régularisation  $C$  et le paramètre  $\Gamma$ .

L'importance de chaque paramètre peut être exprimée de la manière suivante :

- **Le noyau** a la capacité de remodeler les espaces de caractéristiques, facilitant ainsi la modélisation des relations complexes et non linéaires entre les données, ce qui le rend essentiel dans les SVM.

- **Le paramètre de régularisation C** régularise le modèle en contrôlant la taille de la marge et le nombre d'observations mal classées. Un C plus grand permet de classer plus d'observations correctement, mais une marge plus étroite. Un C plus petit produit une marge plus large, mais peut permettre davantage d'erreurs de classification.

- **Gamma** contrôle l'ampleur de l'influence de chaque point d'entraînement sur le modèle. Une valeur Gamma plus élevée signifie que les points d'entraînement les plus proches ont un poids plus élevé et vice versa.

L'objectif de l'optimisation des paramètres SVM consiste à déterminer les valeurs optimales qui permettent d'améliorer au maximum la précision, la sensibilité ou d'autres mesures de performance pertinentes pour une tâche de classification spécifique. Au cours de notre recherche, nous avons examiné diverses configurations pour chaque paramètre, dans le but de découvrir les combinaisons de paramètres les plus efficaces en relation avec notre ensemble de caractéristiques extraites par chaque réseau de neurones convolutionnels utilisé dans notre étude. Les configurations testées sont représentées dans le Tableau [5.2](#).

TABLE 5.2 – Échantillons proposés.

Paramètres		Échantillons	
Noyau	Linéaire	Polynomial	RBF
C	0.1	1	10
Gamma	0.1	0.01	0.001



### 5.4.1 Recherche par grille (Grid Search)

La méthode de recherche par grille est une approche fréquemment employée pour améliorer les performances des modèles d'apprentissage automatique en optimisant leurs hyperparamètres. Cette technique implique de définir un ensemble de valeurs potentielles pour chaque hyperparamètre d'un modèle, puis évaluer les performances du modèle sur toutes les combinaisons possibles de ces valeurs. La recherche par grille a pour but de repérer la configuration d'hyperparamètres qui permet d'obtenir les performances les plus optimales sur un ensemble de données de validation donné. Dans leur livre, Yukun Bao et Zhitao Liu [34] approfondissent l'examen de cette méthode visant à automatiser le processus de réglage des hyperparamètres et à choisir ceux qui conduisent aux performances les plus optimales pour une tâche d'apprentissage automatique donnée. Mettons les lignes sur son importance, en particulier pour des modèles d'apprentissage automatique tels que les SVMs. Les résultats obtenus grâce à cette fonction pour la combinaison des échantillons, en utilisant les vecteurs de caractéristiques extraits par les CNNs, sont récapitulés dans le tableau 5.3.

TABLE 5.3 – Paramètres optimaux.

CNN	Échantillons		
	Noyau	C	Gamma
VGG16	RBF	10	0.001
VGG19	RBF	10	0.001
MobileNet	Linéaire	0.1	0.1
InceptionV3	RBF	10	0.01
DenseNet201	Linéaire	0.1	0.1

Il ressort clairement de ces résultats que les paramètres requis pour l'entraînement de notre SVM varient en fonction des caractéristiques spécifiques extraites par les CNNs pré-entraînés, ce qui souligne l'importance fondamentale de cette fonction dans le domaine de l'apprentissage automatique.

## 5.5 Évaluation de la performance CNN-SVM

Une fois que les caractéristiques extraites sont préparées et que les hyperparamètres pour l'entraînement du SVM sont définis, la phase finale d'entraînement du SVM sera lancée, au cours de laquelle le SVM effectuera la classification définitive. Après cette phase d'entraînement, nous mènerons des expérimentations en évaluant les diverses combinaisons de CNN-SVM et en mesurant leurs performances. L'objectif est d'évaluer l'efficacité de notre étude et de la comparer aux autres approches existantes qui utilisent les CNNs avec leur architecture initiale. Les métriques de performance utilisées dans le cadre de notre recherche sont :

- **Accuracy** est une métrique utilisée pour avoir une vue globale de la performance d'un modèle. C'est la capacité d'un modèle à prédire correctement toutes les classes (positives et négatives) parmi tous les exemples donnés, définie comme suit :

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total d'échantillons de données}} \quad (5.1)$$

- **La précision** d'un modèle représente sa capacité à prédire correctement la classe d'un échantillon de données, et elle est calculée à l'aide de la formule suivante :

$$\text{Précision} = \frac{\text{Somme (Vrais Positifs)}}{\text{Somme (Vrais Positifs + Faux Positifs)}} \quad (5.2)$$

• **Le rappel** (Recall en anglais) est une métrique d'évaluation d'un modèle de classification qui mesure sa capacité à identifier tous les échantillons positifs de la classe cible (vrais positifs) dans un ensemble de données. Elle est donnée par la formule suivante :

$$\mathbf{Rappel} = \frac{\text{Somme (Vrais Positifs)}}{\text{Somme (Vrais Positifs + Faux Négatifs)}} \quad (5.3)$$

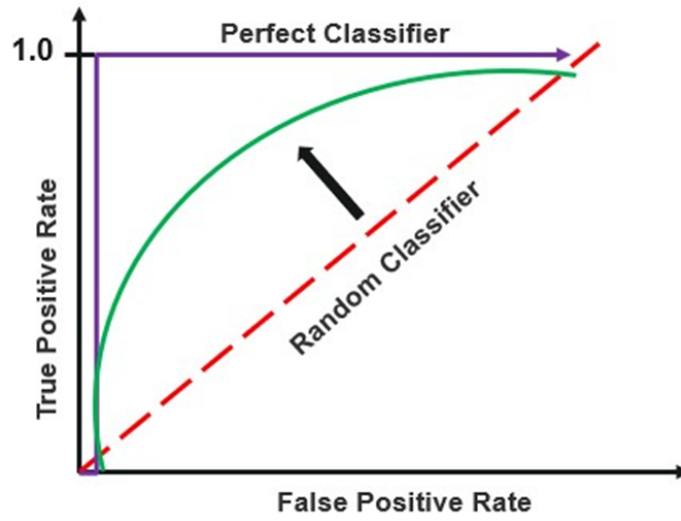
• **Le F1-score** est une mesure de la précision d'un modèle de classification qui combine la précision et le rappel en une seule valeur. Il est particulièrement utile lorsque les classes ne sont pas équilibrées, c'est-à-dire lorsque le nombre d'échantillons positifs et négatifs est différent.

$$\mathbf{F1-score} = \frac{2 \cdot (\text{Précision} \cdot \text{Rappel})}{\text{Précision} + \text{Rappel}} \quad (5.4)$$

Afin de mieux comprendre le comportement de nos modèles et d'identifier les types d'erreurs qu'ils génèrent, nous allons utiliser les métriques suivantes :

• **Matrice de confusion** la matrice de confusion est une méthode importante pour évaluer les performances d'un modèle de classification et pour comprendre les erreurs qu'il commet en comparant les prédictions du modèle aux valeurs réelles des classes d'un ensemble de données.

• **La courbe ROC (Receiver Operating Characteristic)** outil de mesure de la performance d'un modèle de classification qui permet d'évaluer la capacité du modèle à différencier les classes positives et négatives en fonction des seuils de classification. Une courbe ROC qui se rapproche de la diagonale indique une performance moindre du modèle, tandis qu'une courbe éloignée de la diagonale signifie que le modèle est meilleur pour distinguer les classes positives et négatives (voir la Figure [5.19](#)).

FIGURE 5.19 – *Interprétation de la courbe ROC.*

# Chapitre 6

## Expérimentations et discussion

Ce chapitre aborde les aspects liés au matériel, aux logiciels, aux langages de programmation ainsi qu'aux bibliothèques utilisées dans le cadre de notre recherche. Notre objectif principal est d'établir un environnement optimal pour la mise en œuvre de notre approche et la réalisation des expérimentations. Par la suite, nous examinerons les différentes expérimentations menées dans divers scénarios. Enfin, nous procéderons à une comparaison de notre approche par rapport aux études existantes, dans le but de mettre en lumière les avantages de notre recherche.

### 6.1 Environnement matériel

Nous avons mené nos expérimentations en utilisant un processeur **Intel(R) Core (TM) i7-9750H** cadencé à **2,60 GHz (2,59 GHz)** avec une capacité de RAM de **25 Go**, ainsi qu'une carte graphique **NVIDIA GeForce GTX 1660 Ti**. En outre, pour certains calculs, nous avons bénéficié des ressources généreusement fournies par **Google Colab Pro+**, qui mettait à notre disposition une mémoire de **80 Go**, un

espace disque de **166,8 Go**, et une mémoire GPU de **40 Go**. Grâce à ces ressources, nous avons pu mener à bien les phases d'entraînement et d'évaluation requises pour obtenir les résultats que nous présentons dans cette étude.

## 6.2 Environnement logiciel

Nous avons élaboré notre méthode de classification en utilisant l'environnement **Jupyter notebook** ainsi que **Colab Pro+**. Cette approche a été mise en place en utilisant **Python**, un langage de programmation largement utilisé dans les domaines du développement logiciel, de la science des données et de l'apprentissage automatique. Plus précisément, nous avons utilisé la version **3.10.12 de Python** pour mener à bien cette étude. L'utilisation de ces outils et de cette version de Python nous a permis de réaliser notre travail de manière efficace et précise, en tirant pleinement parti des fonctionnalités et des ressources disponibles.

De plus, nous avons réalisé ces expérimentations en utilisant deux bibliothèques open source d'apprentissage automatique élaborées par Google : **TensorFlow** et **Keras**. Leur principal usage consiste à concevoir, entraîner et déployer des modèles d'apprentissage automatique, en particulier ceux basés sur des réseaux de neurones. Ils se distinguent par leur grande flexibilité et leur structure modulaire, ce qui facilite la création de modèles d'apprentissage profond. Nous avons également utilisé la bibliothèque open source **Scikit-learn (sklearn)**, qui propose une variété d'algorithmes supervisés et non supervisés, ainsi que des outils pour la préparation des données et l'évaluation des modèles. Scikit-learn est couramment employée pour mettre en œuvre des techniques telles que les SVM et d'autres méthodes d'apprentissage automatique.

## 6.3 Expérimentations et discussion

L'approche CNN-SVM a été entraînée et testée pour détecter le COVID-19 à partir de trois ensembles de données distincts : (COVID-19 vs. Normal), (COVID-19 vs. Normal vs. Opacité pulmonaire), et (COVID-19 vs. Normal vs. Opacité pulmonaire vs. Pneumonie virale). Nous avons évalué ces résultats en utilisant des métriques que nous avons préalablement définies, et nous allons examiner attentivement les résultats pour chaque combinaison. Par la suite, la combinaison la plus performante parmi celles évaluées a été soumise à un test supplémentaire sur l'ensemble de données COVID-19 Pakistani Patients X-ray, qui se distingue par son indépendance par rapport à l'ensemble d'entraînement ayant servi à la formation des modèles. Ce test a été entrepris dans le but d'évaluer la capacité de notre modèle à maintenir sa robustesse face aux variations de données. Enfin, nous procéderons à une analyse comparative de notre approche par rapport aux travaux antérieurs, en mettant particulièrement l'accent sur les approches qui recourent aux CNN de manière classique.

### 6.3.1 Résultats expérimentaux deux classes

Le tableau [6.1](#) présente les performances des différentes combinaisons de CNN-SVM pour la classification (COVID-19 vs Normale). Selon les données du Tableau [6.1](#), la combinaison VGG16-SVM se distingue en affichant la meilleure précision de 99,03%, suivie de près par la combinaison VGG19-SVM avec une précision d'environ 98,64%. La combinaison MobileNet-SVM obtient également de bons résultats avec une précision de 97,62%. La combinaison DenseNet201-SVM obtient quant à elle des résultats raisonnables avec une précision de 95,64%. Cependant, la combinaison InceptionV3-SVM montre une performance inférieure de 93,46% en termes de précision comparativement avec les autres approches. Ces résultats démontrent l'efficacité des CNN-SVM dans la classification des images COVID-19 par rapport à la

Normale.

TABLE 6.1 – Résultats de la classification problème à deux classes.

Modèle	Global (%)		Pondéré (%)	
	Accuracy	Précision	Rappel	F1-score
<b>VGG16-SVM</b>	<b>99.12</b>	<b>99.03</b>	<b>99.03</b>	<b>99.03</b>
VGG19-SVM	98.5	98.64	98.03	98.32
MobileNet-SVM	97.62	97.62	97.70	97.02
InceptionV3-SVM	92.43	93.46	89.82	91.27
DenseNet201-SVM	96.56	96.71	95.64	96.14

La Figure [6.1](#) illustre les courbes ROC (Receiver Operating Characteristic), offrant une visualisation significative de la performance des modèles de classification pour le problème à deux classes (COVID-19 vs Normale). Dans notre analyse, les courbes ROC de quatre combinaisons (VGG16-SVM, VGG19-SVM, MobileNet-SVM et DenseNet201-SVM) se situent toutes près du coin supérieur gauche, ce qui témoigne d'une capacité remarquable à différencier les classes positives et négatives, avec un taux élevé de vrais positifs et un faible taux de faux positifs. En revanche, la courbe ROC d'InceptionV3-SVM se distingue par son écart par rapport au coin supérieur gauche, suggérant une performance moins optimale dans cette configuration particulière. Cette différence pourrait être attribuée à la minimisation de la taille des vecteurs extraits par ce réseau, ce qui pourrait entraîner une perte d'informations pertinentes.



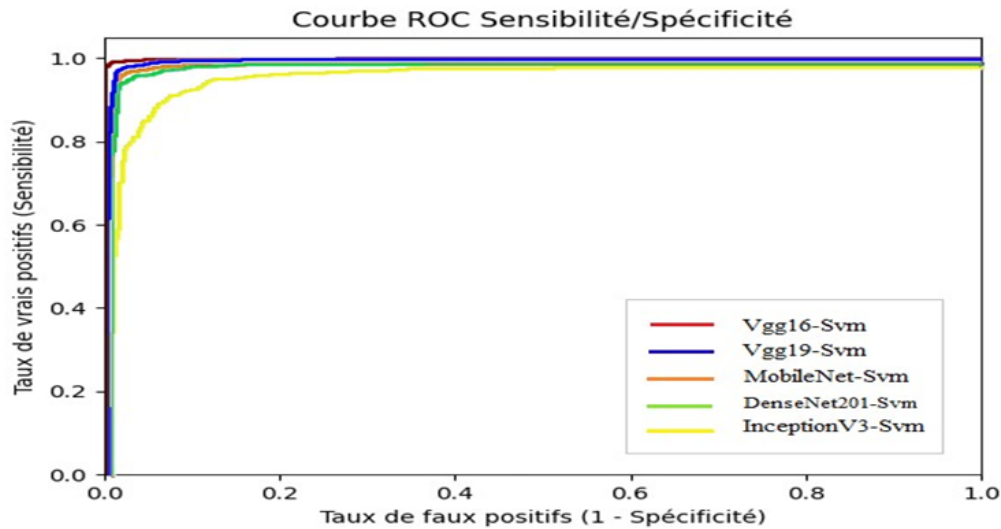


FIGURE 6.1 – Analyse des performances via courbes ROC sur ensemble 1 (2 classes).

Les résultats exposés dans le tableau [6.2](#) démontrent la performance de la combinaison VGG16-SVM dans le contexte de la détection de la Covid-19 par rapport aux cas normaux. Ces performances sont impressionnantes, avec une précision, un rappel et un F1-score de 0,99 pour les deux classes, reflétant une capacité exceptionnelle du modèle à classer avec précision les cas de Covid-19 et les cas normaux. Ces résultats confirment l'efficacité du modèle dans la distinction de ces deux catégories, ce qui est prometteur pour son utilisation potentielle dans le diagnostic de la Covid-19.

TABLE 6.2 – Analyse de la performance par rapport à chaque catégorie.

Classe	VGG16-SVM			
	Précision	Rappel	F1-score	Support
Covid-19	0.99	0.99	0.99	550
Normale	0.99	0.99	0.99	1050

La Figure 6.2 présente la matrice de confusion pour la combinaison VGG16-SVM sur un ensemble de test comprenant 550 cas de COVID-19 et 1050 cas normaux. Sur les 550 cas de COVID-19, le modèle a correctement identifié 543 cas. Cependant, il a commis des erreurs sur 7 échantillons, les classant à tort comme des cas normaux, où le modèle a échoué à détecter correctement la présence du COVID-19. De même, sur les 1050 cas normaux, le modèle a correctement prédit 1043 cas. Cependant, il a également commis des erreurs sur 7 échantillons, les classant à tort comme des cas de COVID-19. Après une analyse approfondie des échantillons mal classés, comme le montre la Figure 6.3, nous avons constaté qu'ils présentaient des caractéristiques très similaires, ce qui les rendait difficiles à distinguer même pour un expert humain. Cela souligne la complexité de la tâche de classification dans ces cas particuliers et met en évidence les limites des modèles et des méthodes d'apprentissage automatique dans des situations où la similarité entre les classes est élevée.

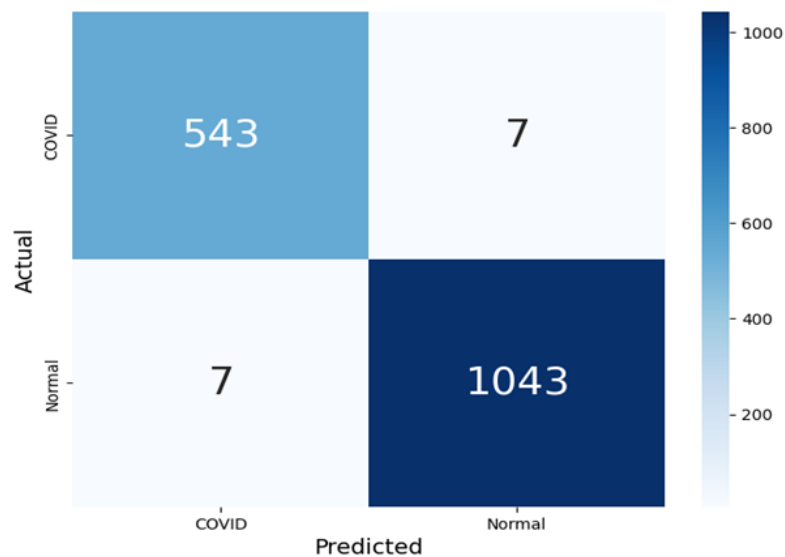


FIGURE 6.2 – Matrice de confusion pour deux classes.

La Figure 6.3 montre des échantillons incorrectement catégorisés. Dans la première ligne, on peut voir des cas de COVID-19 classés à tort comme des cas normaux, tandis que la deuxième ligne montre des cas normaux classés à tort comme des cas de COVID-

19. Comme mentionné précédemment, ces échantillons présentent des caractéristiques similaires, ce qui complique la tâche de classification. Une étude approfondie à l’avenir sera nécessaire pour aborder ces cas particuliers et pour développer une meilleure compréhension des limites de l’apprentissage automatique face à la similitude des données.

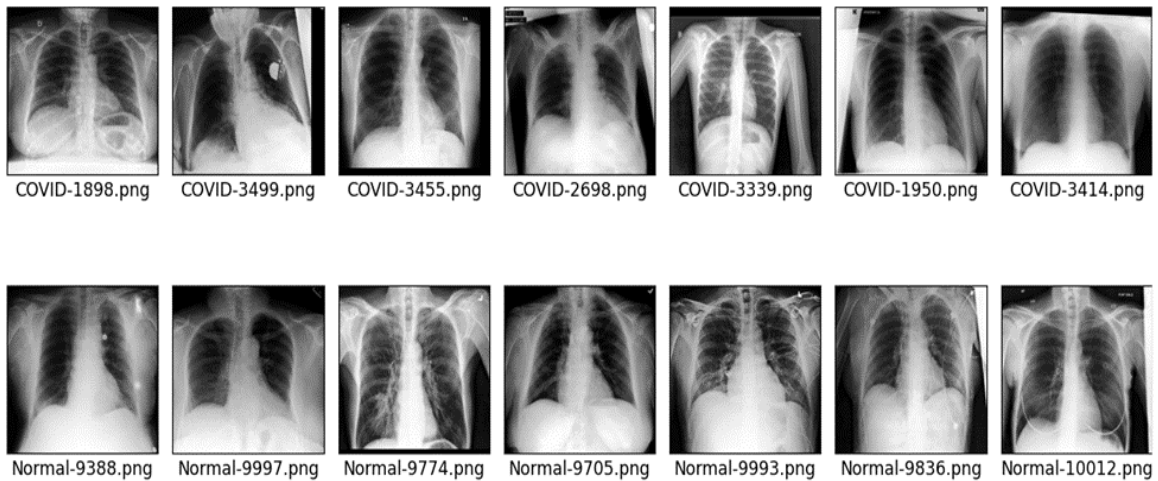


FIGURE 6.3 – Les échantillons mal classifiés.

### 6.3.2 Résultats expérimentaux trois classes

Les performances comparatives des différentes combinaisons CNN-SVM pour la classification (COVID-19 vs. Normale vs. Opacité pulmonaire) sont présentées dans le Tableau [6.3](#). En examinant les résultats, il est clair que la combinaison VGG16-SVM se démarque encore en termes de précision, affichant un impressionnant taux de 95,33%. Cela confirme que cette combinaison est particulièrement performante pour la classification. Parallèlement, les autres combinaisons, à savoir VGG16-SVM, MobileNet-SVM, DenseNet201-SVM et InceptionV3-SVM, ont également obtenu des performances appréciables en termes de précision, avec des taux dépassant largement les 90%. Cette observation atteste de la capacité de ces modèles à réaliser la classification des différentes catégories examinées avec une grande précision.

TABLE 6.3 – Résultats de la classification problème à trois classes.

Modèle	Global (%)		Pondéré (%)	
	Accuracy	Précision	Rappel	F1-score
<b>VGG16-SVM</b>	<b>95.33</b>	<b>95.21</b>	<b>95.92</b>	<b>95.49</b>
VGG19-SVM	93.33	93.32	93.84	93.52
MobileNet-SVM	92.8	92.88	92.96	92.85
InceptionV3-SVM	90.9	90.70	88.90	89.32
DenseNet201-SVM	90.4	90.37	90.66	90.46

La Figure 6.4 présente les courbes ROC, qui illustrent la performance des modèles de classification dans le contexte d'un problème à trois classes (COVID-19 vs. Normale vs. Opacité pulmonaire). Les résultats mettent en évidence que la courbe ROC de VGG16-SVM se rapproche le plus du coin supérieur gauche, indiquant ainsi une performance élevée, caractérisée par une sensibilité et une spécificité élevées. Ensuite, les courbes ROC de VGG19-SVM et de MobileNet-SVM affichent des angles assez similaires et se positionnent légèrement en dessous de celle de VGG16 en termes de performance. Enfin, les courbes ROC de DenseNet201-SVM et d'InceptionV3-SVM révèlent des performances inférieures par rapport aux trois précédemment mentionnées, mais demeurent relativement proches les unes des autres. Ces résultats suggèrent que VGG16-SVM se démarque en tant que modèle le plus performant pour cette tâche de classification, suivi de près par VGG19-SVM et MobileNet-SVM, tandis que DenseNet201-SVM et InceptionV3-SVM présentent des performances légèrement inférieures mais comparables entre elles.

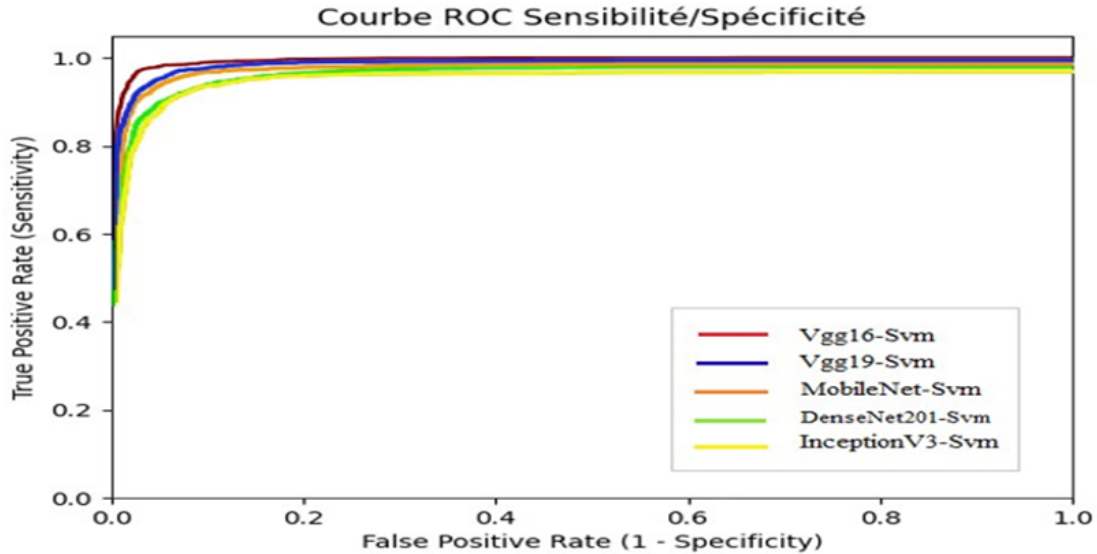


FIGURE 6.4 – Analyse des performances via courbes ROC sur ensemble 2 (3 classes).

Les résultats présentés dans le tableau [6.4](#) de l'analyse de performance du modèle VGG16-SVM sur trois classes différentes sont plutôt intéressants. Pour la classe COVID, le modèle affiche une précision, un rappel et un F1-score remarquablement élevés, tous à 0,98, ce qui indique que le modèle a réussi à détecter cette classe de manière efficace. En ce qui concerne la classe Opacité pulmonaire, bien que la précision soit légèrement plus basse à 0,90, le rappel et le F1-score sont excellents à 0,97 et 0,93 respectivement, suggérant que le modèle est efficace dans la détection de cette classe, même si quelques cas ont été classés à tort. En ce qui concerne la classe Normale, le modèle présente une précision de 0,98, ce qui indique une bonne capacité à identifier les cas normaux. Cependant, le rappel est légèrement plus bas à 0,93, ce qui signifie que certains cas normaux ont été classés à tort dans d'autres catégories.

TABLE 6.4 – Évaluation de la performance par rapport à trois classes.

Classe	VGG16-SVM			
	Précision	Rappel	F1-score	Support
<b>Covid-19</b>	0.98	0.98	0.98	550
<b>Opacité pulmonaire</b>	0.90	0.97	0.93	650
<b>Normale</b>	0.98	0.93	0.95	1050

La Figure [6.5](#) présente la matrice de confusion pour la combinaison VGG16-SVM sur un ensemble de test composé de 550 cas de COVID-19, 650 cas d’opacité pulmonaire et 1050 cas normaux. 537 cas de COVID-19 ont été correctement classés comme COVID-19, 13 cas de COVID-19 ont été mal classés, 7 comme opacité pulmonaire et 6 comme normaux. Pour la classe COVID-19, 537 cas ont été correctement identifiés comme COVID-19, mais il y a eu 13 cas mal classés, dont 7 ont été confondus avec des cas d’opacité pulmonaire et 6 avec des cas normaux. Concernant la classe Opacité pulmonaire, le modèle a réussi à classer avec précision 631 cas sur les 650 présents dans l’ensemble de test. Cependant, il y a eu 19 cas mal classés, dont 3 ont été confondus avec des cas de COVID-19 et 16 avec des cas normaux. En ce qui concerne la classe Normale, le modèle a correctement identifié 977 cas sur les 1050, mais 73 cas ont été mal classés, dont 6 ont été confondus avec des cas de COVID-19 et 67 avec des cas d’opacité pulmonaire. Ces erreurs de classification peuvent être causées par divers facteurs, tels que la similitude des caractéristiques visuelles entre certains cas normaux et les cas d’opacité pulmonaire. Dans le contexte de la radiologie, les images de certaines maladies pulmonaires peuvent présenter des similarités visuelles avec des images de poumons sains, ce qui peut rendre la distinction entre les deux un peu difficile pour certains cas.

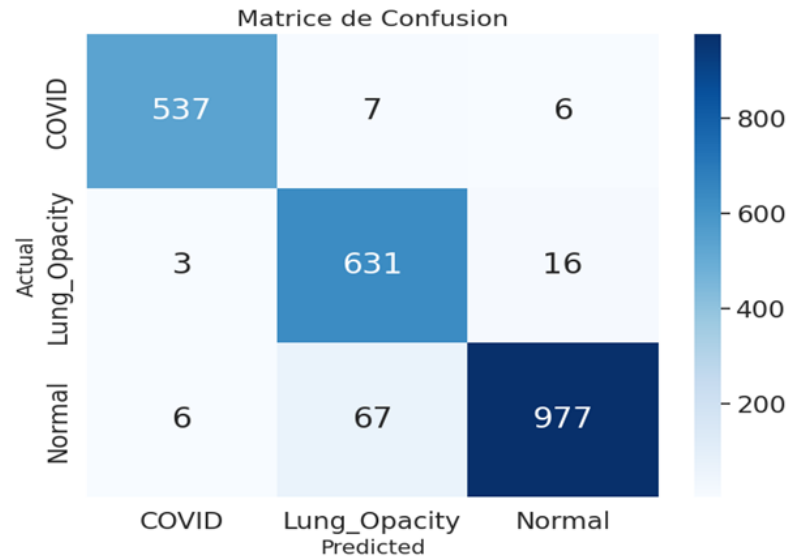


FIGURE 6.5 – Matrice de confusion pour trois classes.

### 6.3.3 Résultats expérimentaux quatre classes

Le tableau [6.5](#) présente une comparaison des performances des diverses combinaisons CNN-SVM pour la classification (COVID-19 vs Normale vs Opacité pulmonaire vs Pneumonie virale). Les résultats révèlent que la combinaison VGG16-SVM obtient la meilleure performance, avec une précision de 96.06%. Cela démontre clairement sa capacité à bien classer les différentes classes du problème de classification à quatre classes. Les modèles VGG19-SVM, MobileNet-SVM, InceptionV3-SVM et DenseNet201-SVM montrent également des résultats raisonnables, avec des taux de précision supérieurs à 92%. Cependant, ils sont légèrement moins performants en comparaison avec la combinaison VGG16-SVM.

TABLE 6.5 – Résultats de la classification problème à quatre classes.

Modèle	Global (%)		Pondéré (%)	
	Accuracy	Précision	Rappel	F1-score
<b>VGG16-SVM</b>	<b>95.46</b>	<b>96.06</b>	<b>96.34</b>	<b>96.15</b>
VGG19-SVM	93.42	94.50	94.18	94.29
MobileNet-SVM	93.10	94.48	93.84	94.10
InceptionV3-SVM	90.08	92.57	89.43	90.58
DenseNet201-SVM	90.73	92.39	91.82	92.05

La Figure 6.6 présente les courbes ROC, qui illustrent la performance des modèles de classification dans le contexte d'un problème à quatre classes (COVID-19 vs Normale vs Opacité pulmonaire vs Pneumonie virale). Les résultats mettent en évidence que la courbe ROC de VGG16-SVM montre la meilleure performance parmi les différentes combinaisons, suivie de près par les autres combinaisons avec une précision presque similaire à celle du problème à trois classes.

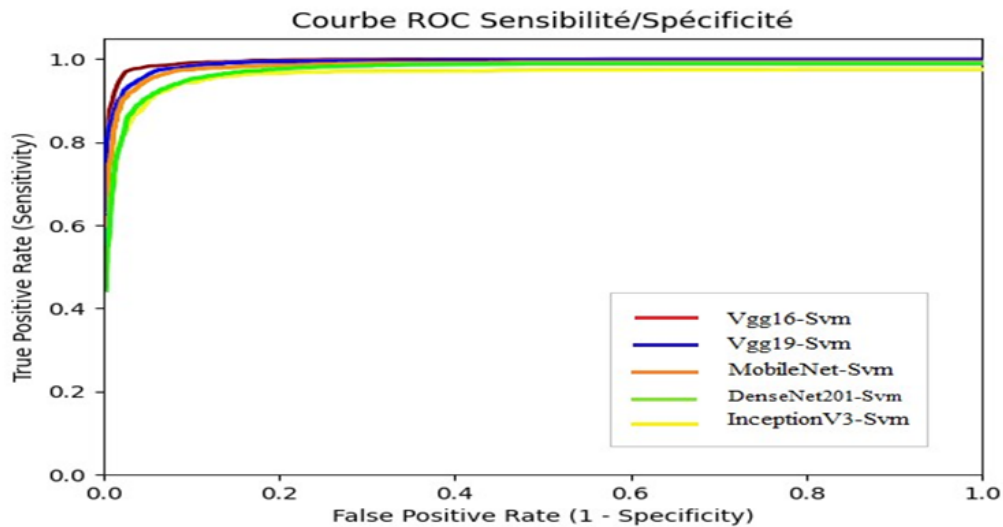


FIGURE 6.6 – Analyse des performances via courbes ROC sur ensemble 3 (4 classes).



Les résultats du tableau 6.6 pour le modèle VGG16-SVM montrent une performance globalement solide sur quatre classes. La classe Covid-19 affiche une précision élevée de 0,98, tout comme le rappel à 0,97, avec un F1-score remarquable de 0,98. Pour la classe Opacité pulmonaire, bien que la précision soit légèrement inférieure à 0,90, le rappel est élevé à 0,97, donnant un F1-score de 0,93. La classe Normale présente une précision de 0,97 avec un rappel légèrement inférieur à 0,93 et un F1-score de 0,95. Enfin, la classe Pneumonie virale montre une précision exceptionnelle de 0,99, un rappel de 0,98 et un F1-score de 0,98. En résumé, le modèle VGG16-SVM démontre une performance robuste, bien qu'il y ait encore de légères marges d'amélioration pour réduire les erreurs de classification dans certaines catégories.

TABLE 6.6 – Évaluation de la performance par rapport à quatre classes.

Classe	VGG16-SVM			
	Précision	Rappel	F1-score	Support
<b>Covid-19</b>	0.98	0.97	0.98	550
<b>Opacité pulmonaire</b>	0.90	0.97	0.93	650
<b>Normale</b>	0.97	0.93	0.95	1050
<b>Pneumonie virale</b>	0.99	0.98	0.98	200

La figure 6.7 présente la matrice de confusion pour la combinaison VGG16-SVM. L'ajout de la classe Pneumonie virale n'a pas eu un impact significatif sur les performances de la classification. Sur un ensemble de test de 200 échantillons, 191 ont été correctement classés, tandis que 9 ont été mal classés. Parmi les cas mal classés, 8 ont été classés comme des cas normaux et 1 a été classé comme un cas de COVID-19. Ces résultats indiquent que la combinaison VGG16-SVM continue de bien performer même avec l'ajout de la classe Pneumonie virale, ce qui est un aspect positif pour la classification médicale complexe.

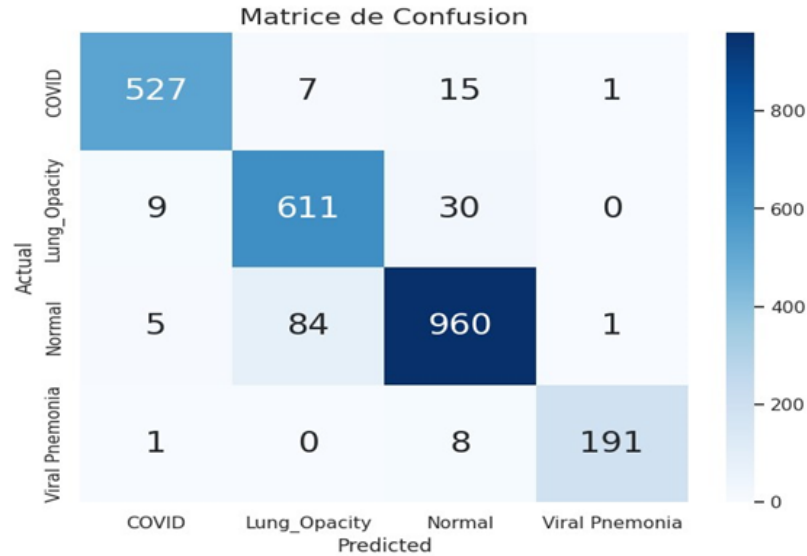


FIGURE 6.7 – Matrice de confusion pour quatre classes.

## 6.4 Évaluation de VGG16-SVM sur de nouvelles données de test

Dans le cadre de notre étude visant à évaluer la robustesse de l’approche VGG16-SVM face aux changements de données, nous avons utilisé un ensemble de quatre cent cinquante nouvelles données pour la phase de test. Ces données sont complètement distinctes de l’ensemble d’entraînement et de test précédemment utilisé. Elles ont été collectées auprès d’hôpitaux pakistanais et sont réparties en deux classes : ”COVID-19” et ”Normale”. Muhammad Umair et al. [11] ont été responsables de la collecte de ces données. Nous avons effectué un prétraitement sur ces données en appliquant la technique d’égalisation d’histogramme, tout en conservant uniquement les données pertinentes et clairement visibles. L’approche VGG16-SVM a obtenu un accuracy de 80.38%. Parmi les trois cent soixante-treize cas de COVID, le modèle a réussi à identifier trois cent onze cas de COVID, mais a mal classé soixante-deux d’entre eux comme étant des cas normaux. De même, parmi les quarante cas normaux, le modèle

a correctement classé vingt et un images, mais a mal classé dix-neuf d'entre elles comme étant des cas de COVID-19. Le principal facteur à l'origine de ces erreurs semble être la qualité des images. La Figure [6.8](#) montre les résultats des prédictions sur les nouvelles données.

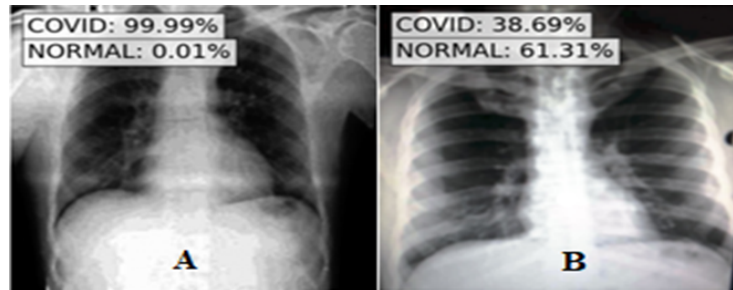


FIGURE 6.8 – Résultats de la prédiction : A : cas de COVID, B : cas Normaux.

## 6.5 Etude comparative

Le Tableau [6.7](#) présente une comparaison entre notre approche performante VGG16-SVM et d'autres méthodes existantes dans la littérature pour résoudre le même problème de classification des images radiographiques de la COVID-19. En revanche, les approches concurrentes dans la littérature se concentrent exclusivement sur l'utilisation de réseaux de neurones convolutionnels pré-entraînés (CNN) pour classer les images radiographiques de la COVID-19 en distinguant différentes formes de pneumonie. Les résultats sur les trois schémas différents (COVID-19 vs Normale), (COVID-19 vs Normale vs Opacité pulmonaire) et (COVID-19 vs Normale vs Opacité pulmonaire vs Pneumonie virale) ont montré une amélioration remarquable par rapport aux études existantes.

Pour la prédiction de virus à partir de deux classes, notre combinaison VGG16-SVM, exposée dans notre étude, a démontré une impressionnante précision de 99,03%. Cette performance la positionne clairement en tête par rapport à plusieurs études de référence antérieures. En comparaison avec l'étude d'Asif Iqbal Khan et al. [\[14\]](#), qui

a atteint un taux de précision de 98,3%, notre approche a affiché une amélioration notable de 0,73%. De manière similaire, en comparaison avec l'étude d'Apostolopoulos et al. [15], qui a obtenu une précision de 98,75%, notre recherche a enregistré une amélioration de 0,28%. Lorsqu'on la compare à d'autres études, la nôtre a également surpassé Fátima et al. [12] (94,92%), Muhammad Umair et al. [11] (96,49%) et Cagin Polat et al. [10] (97,10%), avec des différences significatives de 4,11% par rapport à Fátima et al., 2,54% par rapport à Muhammad Umair et al., et 1,93% par rapport à Cagin Polat et al. En ce qui concerne la classification à trois classes, notre approche basée sur VGG16-SVM a obtenu une précision de 95,21%. Cette performance se traduit par une amélioration de 0,21% par rapport à l'étude d'Asif Iqbal Khanf et al. [14], ainsi qu'une hausse de 1,73% par rapport à l'étude similaire d'Apostolopoulos et al. [15]. Passant au problème à quatre classes, notre approche a atteint une précision de 96,06%, représentant une amélioration significative de 6,06% par rapport à l'étude d'Asif Iqbal Khan et al. [14] dans une situation similaire.

TABLE 6.7 – Comparaison de notre approche avec d'autres études.

Dataset	Articles	Architecture	Performance
<b>2 classes</b>	Khan et al. [14]	CoroNet	98.3%
	Fátima et al. [12]	VGG16	94.92%
	Apostolopoulos et al. [15]	VGG19	98.75%
	Umair et al. [11]	DenseNet-121	96.49%
	Cagin Polat et al. [10]	nCOV-NET	97.10%
	<b>Notre approche</b>	<b>VGG16-SVM</b>	<b>99.03%</b>
<b>3 classes</b>	Khan et al. [14]	CoroNet	95%
	Apostolopoulos et al. [15]	VGG19	93.48%
	<b>Notre approche</b>	<b>VGG16-SVM</b>	<b>95.21%</b>
<b>4 classes</b>	Khan et al. [14]	CoroNet	90%
	<b>Notre approche</b>	<b>VGG16-SVM</b>	<b>96.06%</b>

# Chapitre 7

## Conclusion et perspectives

L'apparition de la pandémie de COVID-19 et l'émergence de ses variants ont plongé l'ensemble du secteur médical dans une situation d'urgence sans précédent. Dans ce contexte particulièrement critique, la nécessité d'obtenir un diagnostic précis pour le virus, dans les délais les plus brefs, revêt une importance vitale. Cette approche est cruciale pour prévenir une détérioration rapide de l'état du patient, anticiper les complications potentielles et mettre en place un traitement spécifiquement adapté à la souche virale en question.

La radiographie est une technique d'imagerie médicale qui est utilisée depuis de nombreuses années pour le diagnostic de diverses maladies. Cependant, en raison de la pandémie de COVID-19, les radiographies thoraciques ont acquis une importance particulière dans le dépistage précoce de la maladie. En outre, ces radiographies peuvent fournir des informations cruciales sur la gravité de la maladie et son évolution au fil du temps. Toutefois, avec l'approche de la saison grippale, le diagnostic de l'épidémie deviendra plus complexe en raison des similitudes entre différents types de pneumonie et le COVID-19, ce qui ajoutera une charge supplémentaire au personnel de santé.

L'objectif de notre travail de recherche est de développer une approche hybride CNN-SVM pour classifier la COVID-19 à partir d'images radiographiques et la distinguer des autres pneumonies, afin d'aider les médecins dans leur processus de diagnostic. Nous avons créé une nouvelle combinaison basée sur les algorithmes d'apprentissage automatique. Pour ce faire, nous avons utilisé cinq CNNs préentraînés couramment cités dans la littérature (VGG16, VGG19, MobileNet, Inception-v3 et DenseNet201) pour extraire des caractéristiques complexes à partir de nos images. Ensuite, nous avons classé ces caractéristiques en utilisant un classifieur SVM, pour lequel nous avons optimisé les paramètres spécifiques à chaque vecteur extrait. De plus, nous avons appliqué une nouvelle technique d'amélioration d'images basée sur l'égalisation d'histogramme à l'ensemble de données avant de former les modèles.

L'étude a été menée en utilisant une base de données publique récente contenant des images radiographiques de la COVID-19. Les expérimentations menées sur trois ensembles de données différents, à savoir (COVID-19 vs. Normale), (COVID-19 vs. Normale vs. Opacité pulmonaire) et (COVID-19 vs. Normale vs. Opacité pulmonaire vs. Pneumonie virale), ont révélé des résultats prometteurs en termes de précision. Ces résultats étaient nettement supérieurs à ceux obtenus dans des études qui se fondaient uniquement sur des CNNs. La combinaison VGG16-SVM s'est avérée être la plus performante dans les trois ensembles. De plus, cette méthodologie a été validée sur un ensemble de données distinct lié à la population au Pakistan, afin d'évaluer sa capacité à maintenir ses performances face aux variations des données, et elle a produit des résultats satisfaisants.

Nous avons l'intention de poursuivre notre recherche en élargissant le champ d'application de cette méthode. Pour ce faire, nous prévoyons de valider davantage notre approche en utilisant un ensemble de données plus étendu, comprenant une plus grande variété de cas de COVID-19 et d'autres affections pulmonaires. Cette expansion de la base de données nous permettra de renforcer la fiabilité de notre méthode et de l'adapter pour une utilisation en temps réel, ce qui serait particulièrement utile

dans des contextes cliniques où des diagnostics rapides sont nécessaires.

De plus, nous envisageons d'étendre notre approche à d'autres types de maladies, notamment les tumeurs cancéreuses. Cette extension de notre recherche nous permettra d'explorer comment notre méthode peut être appliquée à la détection précoce et au diagnostic de diverses affections médicales, ouvrant ainsi la voie à de nouvelles avancées dans le domaine de l'imagerie médicale et de l'intelligence artificielle en santé.

# Bibliographie

- [1] A. KHAN, A. SOHAIL, U. ZAHOORA et A. S. QURESHI, « A survey of the recent architectures of deep convolutional neural networks », *Artificial intelligence review*, vol. 53, p. 5455–5516, 2020.
- [2] M. CONSTANTINO, T. EXARCHOS, A. G. VRAHATIS et P. VLAMOS, « Covid-19 classification on chest x-ray images using deep learning methods », *International Journal of Environmental Research and Public Health*, vol. 20, no. 3, p. 2035, 2023.
- [3] S. SAEEDI, S. REZAYI, H. KESHAVARZ et S. R. NIAKAN KALHORI, « Mri-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques », *BMC Medical Informatics and Decision Making*, vol. 23, no. 1, p. 16, 2023.
- [4] V. VAPNIK et R. IZMAILOV, « Knowledge transfer in svm and neural networks », *Annals of Mathematics and Artificial Intelligence*, vol. 81, no. 1-2, p. 3–19, 2017.
- [5] T. ADAMS, J. DÖRPINGHAUS, M. JACOBS et V. STEINHAGE, « Automated lung tumor detection and diagnosis in ct scans using texture feature analysis and svm. », in *FedCSIS (Communication Papers)*, p. 13–20, 2018.
- [6] A. BENKADJA, I. BISKRI et N. GHAZZALI, « Hybride cnn-svm : vers une meilleure classification des données manuscrites », In *JADT 2022-Proceedings of the 16th International Conference on Statistical Analysis of Textual Data (Vol. 1, pp. 119-126)*. *Vadistat press Edizioni Erranti. July 2022*.
- [7] X.-X. NIU et C. Y. SUEN, « A novel hybrid cnn-svm classifier for recognizing



- handwritten digits », *Pattern Recognition*, vol. 45, no. 4, p. 1318–1325, 2012.
- [8] T. AI, Z. YANG, H. HOU, C. ZHAN, C. CHEN, W. LV, Q. TAO, Z. SUN et L. XIA, « Correlation of chest ct and rt-pcr testing for coronavirus disease 2019 (covid-19) in china : a report of 1014 cases », *Radiology*, vol. 296, no. 2, p. E32–E40, 2020.
- [9] A. ABOUGAZIA, A. ALNUAIMI, A. MAHRAN, T. ALI, A. KHEDR, B. QADOURAH, A. SHAREEF, S. ZITOUNI, S. KAHVECI, B. ALQUDAH *et al.*, « Chest x-ray findings in covid-19 patients presenting to primary care during the peak of the first wave of the pandemic in qatar : their association with clinical and laboratory findings », *Pulmonary medicine*, vol. 2021, 2021.
- [10] Ç. POLAT, O. KARAMAN, C. KARAMAN, G. KORKMAZ, M. C. BALCI et S. E. KELEK, « Covid-19 diagnosis from chest x-ray images using transfer learning : Enhanced performance by debiasing dataloader », *Journal of X-ray Science and Technology*, vol. 29, no. 1, p. 19–36, 2021.
- [11] M. UMAIR, M. S. KHAN, F. AHMED, F. BAOTHMAN, F. ALQAHTANI, M. ALIAN et J. AHMAD, « Detection of covid-19 using transfer learning and grad-cam visualization on indigenously collected x-ray dataset », *Sensors*, vol. 21, no. 17, p. 5813, 2021.
- [12] F. SAIZ et I. BARANDIARAN, « Covid-19 detection in chest x-ray images using a deep learning approach », 2020.
- [13] A. ABBAS, M. M. ABDELSAMEA et M. M. GABER, « Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network », *Applied Intelligence*, vol. 51, p. 854–864, 2021.
- [14] A. I. KHAN, J. L. SHAH et M. M. BHAT, « Coronet : A deep neural network for detection and diagnosis of covid-19 from chest x-ray images », *Computer methods and programs in biomedicine*, vol. 196, p. 105581, 2020.
- [15] I. D. APOSTOLOPOULOS et T. A. MPESIANA, « Covid-19 : automatic detection from x-ray images utilizing transfer learning with convolutional neural networks », *Physical and engineering sciences in medicine*, vol. 43, p. 635–640, 2020.

- [16] R. JAIN, M. GUPTA, S. TANEJA et D. J. HEMANTH, « Deep learning based detection and analysis of covid-19 on chest x-ray images », *Applied Intelligence*, vol. 51, p. 1690–1700, 2021.
- [17] Y. LECUN, Y. BENGIO et G. HINTON, « Deep learning », *nature*, vol. 521, no. 7553, p. 436–444, 2015.
- [18] D. BHATT, C. PATEL, H. TALSANIA, J. PATEL, R. VAGHELA, S. PANDYA, K. MODI et H. GHAYVAT, « Cnn variants for computer vision : History, architecture, application, challenges and future scope », *Electronics*, vol. 10, no. 20, p. 1, 2021.
- [19] D.-N. LE, V. S. PARVATHY, D. GUPTA, A. KHANNA, J. J. RODRIGUES et K. SHANKAR, « Iot enabled depthwise separable convolution neural network with deep support vector machine for covid-19 diagnosis and classification », *International journal of machine learning and cybernetics*, p. 1–14, 2021.
- [20] D. HENDRYCKS et K. GIMPEL, « A baseline for detecting misclassified and out-of-distribution examples in neural networks », *arXiv preprint arXiv :1610.02136*, 2016.
- [21] Y. LECUN, B. BOSER, J. DENKER, D. HENDERSON, R. HOWARD, W. HUBBARD et L. JACKEL, « Handwritten digit recognition with a back-propagation network », *Advances in neural information processing systems*, vol. 2, 1989.
- [22] M. E. CHOWDHURY, T. RAHMAN, A. KHANDAKAR, R. MAZHAR, M. A. KADIR, Z. B. MAHBUB, K. R. ISLAM, M. S. KHAN, A. IQBAL, N. AL EMADI *et al.*, « Can ai help in screening viral and covid-19 pneumonia? », *Ieee Access*, vol. 8, p. 132665–132676, 2020.
- [23] W. ALHAZMI et T. TURKI, « Applying deep transfer learning to assess the impact of imaging modalities on colon cancer detection », *Diagnostics*, vol. 13, no. 10, p. 1721, 2023.
- [24] M. EL-ABED, B. HEMERY, C. CHARRIER et C. ROSENBERGER, « Evaluation de la qualité de données biométriques », *Revue des Nouvelles Technologies de l'Information*, p. 1–22, 2011.

- [25] J. SERIN, M. I. D. IS, V. EBENEZER, A. JENEFA *et al.*, « Gender classification from fingerprint using hybrid cnn-svm », *Journal of Artificial Intelligence and Technology*, 2023.
- [26] Y. WU, Y. HE et Y. WANG, « Multi-class weed recognition using hybrid cnn-svm classifier », *Sensors*, vol. 23, no. 16, p. 7153, 2023.
- [27] T. RAHMAN, A. KHANDAKAR, Y. QIBLAWAY, A. TAHIR, S. KIRANYAZ, S. B. A. KASHEM, M. T. ISLAM, S. AL MAADEED, S. M. ZUGHAIER, M. S. KHAN *et al.*, « Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images », *Computers in biology and medicine*, vol. 132, p. 104319, 2021.
- [28] S. I. de Radiologie Médicale et INTERVENTIONNELLE ( SIRM ), « La radiologia medica ». <https://sirm.org/category/senza-categoria/covid-19/>, 2021.
- [29] M. imaging databank of the valencia REGION, « Bimcv-covid19 ». <https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/1590858128006-9e640421-6711>, 2020.
- [30] J. P. COHEN, P. MORRISON et L. DAO, « Covid-19 image data collection », *arXiv 2003.11597*, 2020.
- [31] D. S. KERMANY, M. GOLDBAUM, W. CAI, C. C. VALENTIM, H. LIANG, S. L. BAXTER, A. MCKEOWN, G. YANG, X. WU, F. YAN *et al.*, « Identifying medical diagnoses and treatable diseases by image-based deep learning », *cell*, vol. 172, no. 5, p. 1122–1131, 2018.
- [32] M. ANOUK STEIN, « Rсна pneumonia detection challenge ». <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data>, 2018.
- [33] N. ADALOGLOU, « Best deep cnn architectures and their principles : From alexnet to efficientnet », *AI Summer*, 2021.
- [34] Y. BAO et Z. LIU, « A fast grid search method in support vector regression forecasting time series », in *Intelligent Data Engineering and Automated Learning—IDEAL 2006 : 7th International Conference, Burgos, Spain, September 20-23, 2006. Proceedings 7*, p. 504–511, Springer, 2006.

## **Annexe A**

# **Detection of coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine**

# Detection of coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine\*

Mohamed Elamine Khoudour  
 Laboratoire d'Intelligence Artificielle Appliquée (LI2A)  
 University of Quebec at Trois-Rivières  
 Trois-Rivières-QC-Canada  
 Mohamed.Elamine.Khoudour@uqtr.ca

Ismail Biskri  
 Laboratoire d'Intelligence Artificielle Appliquée (LI2A)  
 University of Quebec at Trois-Rivières  
 Trois-Rivières-QC-Canada  
 Ismail.Biskri@uqtr.ca

**Abstract**— In the emergency context of COVID-19 and its variants, rapid and accurate diagnosis based on radiographic images is of paramount importance. This avoids confusion with other types of pneumonia and ensures appropriate treatment. This paper presents a hybrid model combining five pre-trained CNNs (VGG16, VGG19, MobileNet, Inception-v3 and DenseNet201) with the SVM classifier. The study was conducted on a recent public database of radiographic images of COVID-19. Experiments on 3 different patterns (COVID-19 vs Normal), (COVID-19 vs Normal vs Lung Opacity) and (COVID-19 vs Normal vs Lung Opacity vs Viral Pneumonia) showed encouraging accuracies, with higher recognition rates compared to studies using CNNs alone. In addition, the approach was validated on a separate dataset linked to Pakistani population, achieving acceptable results.

**Keywords**—CNN, SVM, Classification, Radiography.

## I. INTRODUCTION

The COVID-19 pandemic, caused by the SARS-CoV-2 coronavirus, has profoundly disrupted our lives since 2019. Despite efforts to contain its spread, it's clear that COVID-19 is here to stay, due to the different variants of the virus. We need to adapt to this reality and take the necessary steps to protect ourselves against a new, potentially deadly wave of the virus. There are various methods for detecting this virus in the human body, mainly using biological and chemical approaches.

PCR tests are widely recognized as the gold standard for diagnosing infection, however, PCR tests can give false-negative results. This means that some COVID-19 patients may test negative when they are actually infected. What's more, these tests often take a long time to provide results, which can increase the time to diagnosis [1]. Antigen testing is a faster alternative to PCR testing, but may be slightly less sensitive. Serological tests help identify past infections, but are not suitable for early detection.

It is well known that radiography is an imaging technique that has been used for many years to diagnose various diseases. However, due to the COVID-19 pandemic, chest X-rays have taken on particular importance in the early diagnosis of the disease [2]. In addition, chest X-rays can provide important information on the severity of the disease and its evolution over time. But, with the approach of the flu season, diagnosis of the epidemic will become difficult due to the many similarities

between different types of pneumonia and Covid-19, adding an extra burden to healthcare staff.

Advances in artificial intelligence have enabled machines to autonomously solve complex problems. Deep convolutional neural networks and support vector machines are effective machine learning techniques for intricate image recognition, particularly in medicine. In our paper, we demonstrate the benefits of combining these applications for improved classification performance. The remainder of this article is organized as follows: the second section describes related work in the detection of COVID-19 using radiographic imaging. The third and fourth sections are dedicated to CNN and SVM, respectively. The fifth section details the hybrid CNN-SVM approach. The sixth section delves deeply into our own method, the seventh section presents and discusses the experimental results, and finally, the eighth section addresses the conclusion and future prospects to consider.

## II. STATE OF THE ART

Recently, a great deal of research has been carried out to demonstrate the detection of COVID-19 using radiographic images, using various techniques based on artificial intelligence. Approaches such as transfer learning and new network architectures have been proposed to improve model performance and enable classification of COVID-19, normal lung disease and other pulmonary conditions. Cagin Polat et al. [3] achieved 97.1% accuracy using a convolutional neural network called nCoV-NET, which was optimized using the DenseNet-161 architecture. Their model was trained and evaluated on a dataset comprising 299 COVID-19 cases and 1,522 non-COVID-19 cases. Muhammad Umair et al. [4] employed four pre-trained deep neural networks, namely VGG16, ResNet-50, DenseNet-121 and MobileNet, using transfer learning on a dataset consisting of 7232 chest X-ray images (COVID-19 and normal). The accuracy rates obtained were 83.27%, 92.48%, 96.49% and 96.48% respectively.

Ioannis D Apostolopoulos et al. [5] retrained five deep neural networks (VGG19, MobileNet v2, Inception, Xception, Inception ResNet v2) using 1427 X-ray images, including 224 Covid-19 images, 700 bacterial pneumonia images, and 504 normal images. VGG19 and MobileNet v2 demonstrated superior classification accuracy among CNNs, achieving 98.75 % and 96.78% for two-class, and 93.48% and 94.72% for three-

class scenarios. Fátima A. Saiz et al. [6] developed an object detection architecture that was trained and tested on a dataset consisting of 1,500 images of uninfected, COVID-19-infected and pneumonia patients. To improve the data, a contrast adjustment technique was applied to the images. For this study, VGG16 was used as a template and achieved a sensitivity of 94.92% and a specificity of 92.00% in the detection of COVID.

Muhammad E et al. [7] established a database comprising 423 cases of COVID-19, 1485 cases of viral pneumonia and 1579 normal cases. They used transfer learning and image augmentation to train deep pre-trained convolutional neural networks (CNNs). The networks classified the cases into two categories: normal vs. COVID-19 and normal vs. viral vs. COVID-19, achieving accuracies of 99.7% and 97.94% respectively with DenseNet201 for two-class and three-class classification. Asif Iqbal Khan et al. [8] developed a convolutional neural network model, CoroNet, to detect COVID-19 on radiographs. CoroNet demonstrated exceptional accuracy of 89.6% in overall detection, with precision rates of 93% and recall of 98.2% for COVID-19 cases in a four-category classification, as well as 95% accuracy in a three-category classification.

The studies presented indicate that deep CNNs are well suited for the detection of COVID-19 from X-ray images. However, most of these studies are impeded by the limited amount of data available, forcing the authors to use data augmentation techniques to feed their networks. This constraint can make it difficult to validate results on real samples or on a large number of data. In addition, it is important to note that some of these studies consider pneumonia cases as non-COVID cases, without specifying the type of pneumonia in question. As a result, the precise identification of different pulmonary conditions may be limited. Another aspect to consider is that these studies have been validated on a small number of images from a single source, which may introduce a potential bias in the results. The contributions of this study can be clearly defined as follows:

- The use of a large dataset from the latest available public update.
- A noise elimination process was applied to the entire dataset using a histogram equalization-based enhancement technique.
- A novel combinatorial approach was adopted, involving the extraction of features from each class using pre-trained convolutional neural networks, followed by their classification using Support Vector Machines (SVM).
- This study was validated on a large number of data and used other data sources that differ from the initial training source.

### III. CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNNs) are widely recognized as the best-performing models for medical image classification, including the detection of Covid-19 from X-rays and CT scans, as well as the detection of cancerous tumors from CT scans [7,9]. Their success lies mainly in their ability to

automatically extract relevant features using convolutional layers, which leads to a significant improvement in their classification capability. Convolution involves applying a filter representing features such as edges or textures to the input image. The filter is moved progressively across the image, calculating a convolution product at each position, generating an activation map or feature map highlighting the detected features. The feature vectors obtained are then passed to a fully connected layer, or dense layer, usually the last layer in the network. This layer adjusts the connection weights between extracted features and output classes for accurate classification.

### IV. SUPPORT VECTOR MACHINE (SVM)

SVMs are a family of machine learning algorithms developed by Vladimir Vapnik [10], used to solve classification, regression and anomaly detection problems. They aim to separate data into classes using a boundary or hyperplane, while maximizing the distance between the different groups of data and the separating boundary. This feature also makes them suitable for medical applications. Tim Adams et al. [11] have proposed an automated SVM-based system for the diagnosis of lung tumors in CT scans, achieving remarkable accuracy with respect to the minimal data set used.

#### A. Optimization function:

SVMs use optimization techniques to adjust parameters in order to find the best decision frontier that maximizes the margin of separation between classes. This enables SVM to be an effective model for classification, even in cases where the data are not linearly separable.

$$\text{Minimize } \frac{1}{2} \|W\|^2 + C \sum_{i=1}^n \xi_i \quad (1)$$

$$\text{Sejut a } Y_i(w_0 + X_i w) \geq 1 - \xi_i \quad (2)$$

$$\xi_i \geq 0 \quad (3)$$

Where:  $w$  is the weight vector that defines the decision hyperplane,  $C$  a regularization parameter that controls the trade-off between maximizing margin and minimizing classification errors,  $\xi_i$  deviation variables to manage classification errors,  $X_i$  are the training vectors and  $Y_i$  are corresponding class labels.

### V. HYBRID CNN-SVM

The CNN-SVM architecture combines the capabilities of Convolutional Neural Networks (CNN) to extract hierarchical features at different scales from data through their convolutional layers, while incorporating the classification skills of Support Vector Machines (SVM). To achieve this, the last layer of the CNN, which is the fully connected layer dedicated to classification, is replaced by an SVM that seeks to find an optimal hyperplane to separate different classes in the feature space, as illustrated in Fig. 1. This combination has demonstrated superior performance compared to the individual use of each method in various domains. In their paper, Benkadja, A. et al. [12], demonstrated the effectiveness of this approach for handwritten character classification compared to the isolated use of each method. Serin J. et al. [13] developed a hybrid CNN-SVM model for gender classification based on fingerprints. This hybrid model achieved an accuracy of 99.25%, surpassing the CNN alone, which achieved an accuracy of 98%.

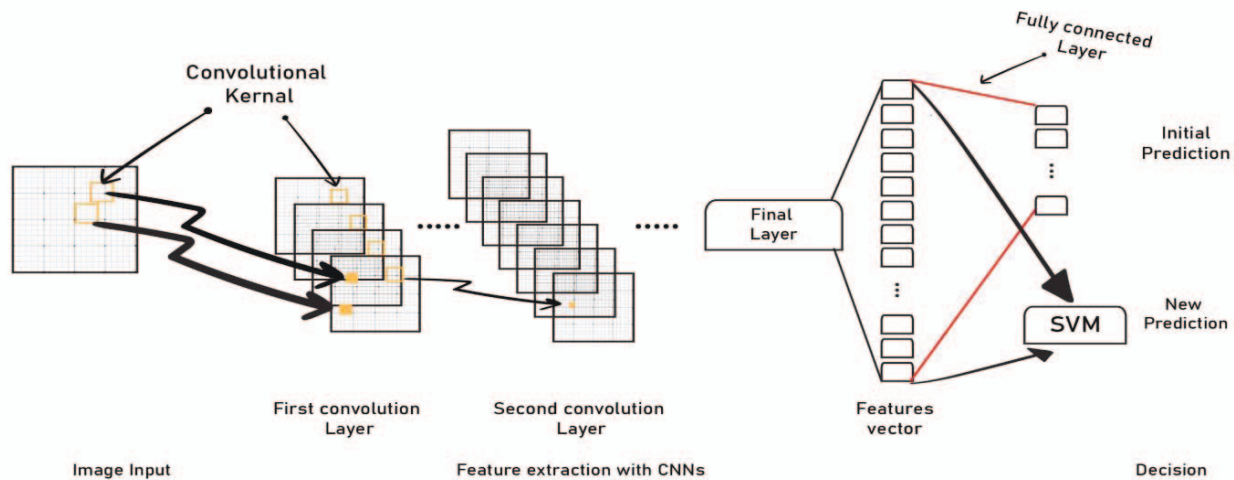


Fig. 1. CNN-SVM architecture.

## VI. METHODOLOGIE

Our approach to X-ray image classification is to use pre-trained CNNs to extract features from X-ray images and classify them using the SVM classifier. To do this, we first improve the quality of our data. Then we load the pre-trained CNNs and train them to extract features from each class. This takes advantage of pre-trained CNNs architectures, which can extract complex features from images. The outputs of the last convolution layer are then flattened into a one-dimensional vector, yielding a set of features for each image. The extracted vector is then used as input to train an SVM with parameters specifically chosen to optimize model performance for classification purposes.

### A. Database

Researchers from Qatar University, the University of Dhaka, and collaborators in Pakistan and Malaysia have created a database of chest X-ray images for COVID-19 cases, validated through PCR and other tests, along with images of normal and viral pneumonia [7,14]. This database was compiled from various sources, including the COVID-19 database of the Italian Society of Medical and Interventional Radiology, the PadChest dataset from the Image Bank of the Region of Valencia, the New Coronavirus 2019 dataset on GitHub, and the "Chest X-Ray Images (pneumonia)" databases on Kaggle and the Radiological Society of North America (RSNA) Kaggle.

### B. Data pre-processing

Tawsifur Rahman et al.[14] have demonstrated in their paper the importance of improving data quality on the performance of convolutional neural networks. Among the various image processing techniques available, histogram equalization (HE) is widely recommended for improving image quality. This technique works by adjusting the gray level distribution of the image by stretching or compressing the histogram to optimize use of the full dynamic range of available gray levels. Fig. 2 and Fig. 3 illustrate the transformation of COVID-19 (A), lung opacity (B), normal (C) and viral pneumonia (D) images before and after the application of histogram equalization, respectively.

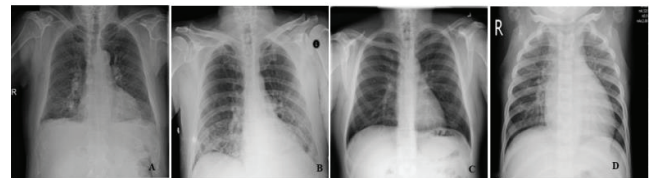


Fig. 2. Radiographic images before histogram equalization.



Fig. 3. Radiographic images after histogram equalization.

We created a new dataset comprising enhanced images. We then divided this dataset into three distinct sets: (COVID-19 vs Normal), (COVID-19 vs Normal vs Lung Opacity) and (COVID-19 vs Normal vs Lung Opacity vs Viral Pneumonia). This will enable us to evaluate the performance of our method against that of other researchers under similar conditions. Table I illustrates the dataset used in the study.

TABLE I. DATASET USED IN THE STUDY.

Type	Total	Training	Validation
COVID-19	3614	3064	550
Normal	10192	9142	1050
Viral pneumonia	1323	1123	200
Lung opacity	6011	5361	650

### C. Feature extraction with CNNs

Deep CNNs are a machine learning technique particularly suited to medical image recognition. In our approach, we used five pre-trained convolutional neural networks to extract features from our data. These networks are : VGG16, VGG19, MobileNet, Inception-V3 and DenseNet201. We chose them because of the excellent results obtained with these networks in state-of-the-art work for image classification of COVID-19. We resized our data to make it compatible with the structure of the pre-trained neural networks chosen for our study: (224x224 pixels) for the VGG16, VGG19, MobileNet and DenseNet201 models, and (299x299 pixels) for the Inception-V3 model. The models are trained to extract features with a batch size equal to 32. In our approach, we made a modification by replacing the fully connected layer of these networks with an SVM classifier. Once we had extracted the features from the images, we flattened them into a one-dimensional vector so that we could pass them on to our SVM classifier.

### D. Parameter optimization for SVM training

SVMs rely on a series of parameters that influence the model's behavior during the learning phase. The three key parameters are the choice of kernel, the regularization parameter (C), and the Gamma parameter. The aim of SVM parameter optimization is to determine optimal values that maximize accuracy, sensitivity, or other relevant performance measures for a specific classification task. In our study, we examined different combinations of these parameters, including the choice of kernel (linear, polynomial, RBF), C values (0.1, 1, 10), and Gamma values (0.1, 0.01, 0.001), using the Grid Search method. This method aims to identify the hyperparameter configuration that achieves optimal performance with respect to the dataset in question, i.e., the feature vector extracted by the CNNs in our case. The results of this approach are shown in table II. It is clear from these results that the parameters required for training our SVM vary according to the specific features extracted by the CNNs, thus underlying the crucial importance of this step in the machine learning process.

TABLE II. OPTIMAL PARAMETERS

CNNs	Parameters		
	Kernel	C	Gamma
VGG16	RBF	10	0.001
VGG19	RBF	10	0.001
MobileNet	Linear	0.1	0.1
InceptionV3	RBF	10	0.01
DenseNet201	Linear	0.1	0.1

### E. CNN-SVM performance evaluation

After training our SVM classifier with the features extracted by the CNNs we mentioned earlier, we'll evaluate its performance using the following metrics in order to assess the effectiveness of our combined CNN-SVM approach :

$$\text{Accuracy} = \frac{\text{Number of correctly classified images}}{\text{Total number of images}} \quad (4)$$

$$\text{Precision} = \frac{\text{Sum of all True Positives (TP)}}{\text{Sum of all True Positives (TP) + All False Positives (FP)}} \quad (5)$$

$$\text{Recall} = \frac{\text{Sum of all True Positives (TP)}}{\text{Sum of all True Positives (TP) + All False Negatives (FN)}} \quad (6)$$

$$\text{F - measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

## VII. IMPLEMENTATION AND EXPÉRIMENTATION

In this study, we carried out various experiments using the TensorFlow and Keras libraries with Python 3.10.12. We ran the experiments on an Intel(R) Core (TM) i7-9750H processor running at 2.60 GHz (2.59 GHz) with 25 GB RAM, and an NVIDIA GeForce GTX 1660 Ti graphics card. In addition, Google Colab Pro+ was used for some calculations, offering 80 GB memory, 166.8 GB disk and 40 GB GPU RAM.

### A. Experimental results two class problem

Table III presents the performance of different CNN-SVM combinations for classification (COVID-19 vs. Normal). According to the data in table III, the VGG16-SVM combination stands out with the highest precision of 99.03%, closely followed by the VGG19-SVM combination with a precision of around 98.64%. The MobileNet-SVM combination also achieved good results, with a precision of 97.62%. The DenseNet201-SVM combination achieves reasonable results, with a precision of 96.71%. However, the InceptionV3-SVM combination obtained a lower precision of 93.46% compared to the other approaches. This decrease could be due to the use of a global average pooling layer at the end of the networks to reduce the size of the feature vectors, making them compatible with the available memory (RAM).

TABLE III. RESULTS OF CLASSIFICATION TWO CLASS PROBLEM

Model	Overall			
	A	P	R	F
<b>VGG16-SVM</b>	<b>99.12</b>	<b>99.03</b>	<b>99.03</b>	<b>99.03</b>
VGG19-SVM	98.5	98.64	98.03	98.32
MobileNet-SVM	97.62	97.62	97.70	97.02
InceptionV3-SVM	92.43	93.46	89.82	91.27
DenseNet201-SVM	96.56	96.71	95.64	96.14

Fig. 4 (A) shows the confusion matrix for the VGG16-SVM combination on a test set comprising 550 COVID-19 cases and 1050 normal cases. Of the 550 COVID-19 cases, the model correctly identified 543 cases. However, it made errors on 7 samples, misclassifying them as normal cases, where the model failed to correctly detect the presence of COVID-19. Similarly, of the 1050 normal cases, the model correctly predicted 1043 cases. However, it also erred on 7 samples, misclassifying them as COVID-19 cases. After analyzing these misclassified samples, we found that they had very similar characteristics, making them difficult to distinguish even for a human expert. The complexity of classification underlines the importance of



collaboration between machine learning and medicine to better understand errors. Medical experts can analyze misclassified samples in detail, which could significantly improve the classification of similar cases.

### B. Experimental results three class problem

The comparative performances of different CNN-SVM combinations for classification (COVID-19 vs Normal vs Lung Opacity) are presented in Table IV. Upon examining the results, it is evident that the VGG16-SVM combination stands out once again in terms of precision, displaying an impressive rate of 95.21%. This confirms that this combination is particularly adept at classification. As for the other combinations, they also achieved good results in terms of precision, with rates exceeding 90%. This demonstrates that these models are also capable of accurately classifying the different classes.

TABLE IV. RESULTS OF CLASSIFICATION THREE CLASS PROBLEM

Model	Overall		Weighted	
	A	P	R	F
<b>VGG16-SVM</b>	<b>95.33</b>	<b>95.21</b>	<b>95.92</b>	<b>95.49</b>
VGG19-SVM	93.33	93.32	93.84	93.52
MobileNet-SVM	92.8	92.88	92.96	92.85
InceptionV3-SVM	90	90.70	88.90	89.32
DenseNet201-SVM	90.4	90.37	90.66	90.46

Fig. 4 (B) presents the confusion matrix for the VGG16-SVM combination on a test set consisting of 550 COVID-19 cases, 650 cases of lung opacity, and 1050 normal cases. Among the COVID-19 cases, 537 were correctly classified as COVID-19, while 13 were misclassified, with 7 as lung opacity and 6 as normal cases. For the COVID-19 class, 537 cases were correctly identified as COVID-19, but there were 13 cases misclassified, out of which 7 were confused with lung opacity cases and 6 with normal cases. Regarding the lung opacity class, the model achieved accurate classification for 631 out of 650 cases in the test set. However, there were 19 cases misclassified, including 3 confused with COVID-19 cases and 16 with normal cases. Concerning the Normal class, the model correctly identified 977 out of 1050 cases, but 73 cases were misclassified, with 6 confused with COVID-19 cases and 67 with lung opacity cases. These classification errors could stem from various factors, such as visual feature similarities between certain normal cases and cases of lung opacity.

### C. Experimental results four class problem

Table V provides a comprehensive comparison of the performances exhibited by various CNN-SVM combinations for classification (COVID-19 vs Normal vs Lung Opacity vs Viral Pneumonia). Notably, the findings underscore the exceptional performance of the VGG16-SVM combination, boasting an impressive precision score of 96.06%. This resounding success is a testament to its remarkable capacity to adeptly classify the distinct classes within the intricate four-class classification problem. Intriguingly, while the VGG16-SVM combination takes the lead, the other models VGG19-SVM, MobileNet-

SVM, InceptionV3-SVM, and DenseNet201-SVM also exhibit commendable outcomes. Although they slightly lag behind the benchmark set by VGG16-SVM in terms of precision, their results still stand as a testament to their efficacy in tackling the complex challenges posed by multi-class classification scenarios.

TABLE V. RESULTS OF CLASSIFICATION FOUR CLASS PROBLEM.

Model	Overall		Weighted	
	A	P	R	F
<b>VGG16-SVM</b>	<b>95.46</b>	<b>96.06</b>	<b>96.34</b>	<b>96.15</b>
VGG19-SVM	93.42	94.50	94.18	94.29
MobileNet-SVM	93.10	94.48	93.84	94.10
InceptionV3-SVM	90.08	92.57	89.43	90.58
DenseNet201-SVM	90.73	92.39	91.82	92.05

Fig. 4 (C) shows the confusion matrix for the VGG16-SVM combination. The addition of the Viral Pneumonia class had no significant impact on classification performance. Out of a test set of 200 samples, 191 were correctly classified, while 9 were misclassified. Of the misclassified cases, 8 were classified as normal and 1 as COVID-19. These results indicate that the VGG16-SVM combination continues to perform well even with the addition of the Viral Pneumonia class, which is a positive aspect for complex medical classification.

### D. VGG16-SVM evaluation on new test data

As part of our study to assess the robustness of the VGG16-SVM approach against data variations, we utilized a new set of 450 data points for the testing phase. These data points are entirely distinct from the previously used training and testing sets. They were sourced from Pakistani hospitals and are categorized into two classes: "Covid19" and "Normal.". Muhammad Umair et al [4] were responsible for data collection. We pre-processed the data using the histogram equalization technique, retaining only relevant and clearly visible data. The VGG16-SVM approach achieved an accuracy of 80.38%. Among the 373 Covid cases, the model successfully identified 311 Covid cases, but misclassified 62 of them as normal cases. Similarly, among the 40 normal cases, the model correctly classified 21 images, but misclassified 19 of them as Covid19 cases. The main factor behind these errors appears to be image quality. Fig. 5 shows the prediction results on the new data.

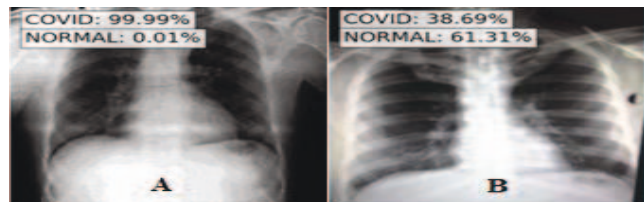


Fig. 5. Prediction results: A: COVID cases, B: normal cases.

Table VI presents a comparison between our approach and other existing methods in the literature to address the same problem of classifying COVID-19 radiographic images. In

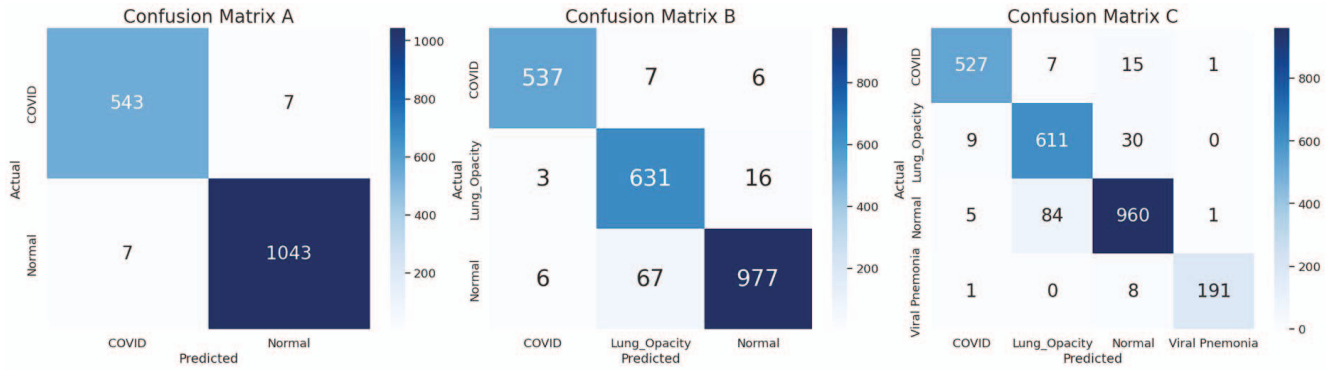


Fig. 4. Confusion Matrix for Two-Class (A), Three-Class (B), and Four-Class (C) classification using VGG16-SVM.

contrast, competing approaches in the literature focus exclusively on using pre-trained CNNs for classifying COVID-19 radiographic images by distinguishing different forms of pneumonia.

TABLE VI. COMPARISON OF OUR APPROACH WITH OTHER METHODS.

Dataset	Articles	Architecture	Performance
Two class	Asif et al. [8]	CoroNet	98.3 %
	Fátima et al. [6]	VGG16	94.92 %
	Apostolopoulos et al. [5]	VGG19	98.75 %
	Muhammad et al. [4]	DenseNet-121	96.49 %
	Cagin Polat et al. [3]	nCOV-NET	97.10 %
	<b>Our approach</b>	<b>VGG16-SVM</b>	<b>99.03 %</b>
Three class	Asif et al. [8]	CoroNet	95 %
	Apostolopoulos et al. [5]	VGG19	93.48 %
		<b>Our approach</b>	<b>VGG16-SVM</b>
Four class	Asif et al. [8]	CoroNet	90 %
		<b>Our approach</b>	<b>VGG16-SVM</b>

## VIII. CONCLUSION

In order to assist physicians in diagnosing COVID-19 from radiographic images and distinguishing it from other pneumonias, we developed a new approach based on machine learning algorithms. We employed five commonly used pre-trained CNNs in the literature to extract features from our data, which were then classified using the SVM classifier with specific parameters optimized for each extracted vector. A novel image enhancement technique based on histogram equalization was applied to the dataset before model training. The VGG16-SVM combination was found to perform the best in all three scenarios. This high-performance approach was also tested on a new dataset separate from the training and validation dataset, and the results obtained were satisfactory. We intend to continue our work by further refining this approach to better distinguish complex cases and exploring other data enhancement techniques, with the aim of making it perfectly suited for real-time use. Additionally, we plan to extend it to other types of diseases, including cancer tumors. This initiative has the

potential to save human lives by enabling faster and more precise medical intervention, which can greatly enhance the prospects for patient treatment and recovery.

## REFERENCES

- [1] T. Ai et al. Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases. *Radiology*. 2020 Aug;296(2):E32-E40.
- [2] A. Abougazia et al. Chest X-Ray Findings in COVID-19 Patients Presenting to Primary Care during the Peak of the First Wave of the Pandemic in Qatar: Their Association with Clinical and Laboratory Findings. *Pulm Med*. 2021 Oct 27.
- [3] Ç. Polat , O. Karaman , C. Karaman, G. Korkmaz , MC. Balcı , SE. Kelek. COVID-19 diagnosis from chest X-ray images using transfer learning: Enhanced performance by debiasing dataloader. *J Xray Sci Technol*. 2021;29(1):19-36.
- [4] M. Umair et al. Detection of COVID-19 Using Transfer Learning and Grad-CAM Visualization on Indigenously Collected X-ray Dataset. *Sensors* 2021, 21, 5813.
- [5] I. D. Apostolopoulos, T. A. Mpesiana. Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Phys Eng Sci Med* 43, 635–640 (2020).
- [6] F. A. Saiz, I. Barandiaran. (2020). COVID-19 Detection in Chest X-ray Images using a Deep Learning Approach. Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Donostia – San Sebastián (Spain).
- [7] M. E. H. Chowdhury et al., "Can AI Help in Screening Viral and COVID-19 Pneumonia?," in *IEEE Access*, vol. 8, pp. 132665-132676, 2020.
- [8] AI. Khan , JL. Shah , MM.Bhat. CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Comput Methods Programs Biomed*. 2020 Nov.
- [9] M. Constantinou, T. Exarchos, A.G. Vrahatis et P. Vlamos. COVID-19 Classification on Chest X-ray Images Using Deep Learning Methods. *Int. J. Environ. Res. Public Health* 2023, 20, 2035.
- [10] V. Vapnik, R. Izmailov. Knowledge transfer in SVM and neural networks. *Ann Math Artif Intell* 81, 3–19 (2017).
- [11] T. Adams, J. Dörpinghaus, M. Jacobs, et V. Steinhage. (2018). Automated lung tumor detection and diagnosis in CT Scans using texture feature analysis and SVM.
- [12] A. Benkadja, A. Ben Ayed, I. Biskri,, et N. Ghazzali. Statistical profiling of Hybride CNN-SVM effectiveness. In *New Frontiers in Textual Data Analysis* (M. Misuraca & G. Giordano, editors) 2023.
- [13] J. Serin, T. Vidhya Keren, S. Mary Ivy, V. Ebenezer, A. Jeneffa, Gender Classification from Fingerprint Using Hybrid CNN-SVM, *Journal of Artificial Intelligence and Technology* (2023).
- [14] T. Rahman, et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Computers in Biology and Medicine*, 132, 104319. 2021.