

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة مصطفى اسطمبولي معسكر
Université Mustapha Stambouli de Mascara



Faculté des Sciences Exactes
Département Informatique

Thèse de Doctorat 3^{ème} cycle

Spécialité : Informatique

Intitulée

Deep learning pour l'aide au diagnostic des lésions cutanées

Présentée par : Benyahia Samia

Devant le jury :

Président	Teggar Hamza	MCA	Université de Mascara
Directeur de thèse	Meftah Boudjelal	Professeur	Université de Mascara
Co-Directeur de thèse	Lézoray Olivier	Professeur	Université Caen Normandie, France
Examineur	Adjoudj Réda	Professeur	Université de Sidi Belabbes
Examinatrice	Boufera Fatma	MCA	Université de Mascara

Année universitaire 2021-2022

Dédicace

A mon cher père

A ma chère mère

A mes chers frère et sœurs

En témoignage de leur soutien indéfectible et inconditionnel ainsi que des sacrifices consentis pour ma réussite.

*A mon directeur de thèse le professeur Meftah Boudjelal à qui je rends
hommage*

Pour ses qualités morales et intellectuelles

Pour sa rigueur scientifique et ses conseils avisés

Pour sa disponibilité permanente et sa grande confiance en moi.

Remerciements

Je tiens à exprimer mes plus sincères remerciements à mon directeur de thèse, Monsieur Meftah Boudjelal, professeur à l'université Mustapha Stambouli de Mascara, pour avoir encadré et dirigé ma thèse. De plus, je le remercie profondément pour son soutien permanent, sa grande disponibilité et sa qualité d'encadrement dont j'ai eu l'honneur de bénéficier tout au long de ce travail de recherche.

Je tiens aussi à remercier chaleureusement mon co-directeur de thèse, Monsieur Olivier Lezoray, professeur à Université Caen Normandie France, pour son suivi, son appui technique et ses conseils avisés prodigués.

Je tiens également à remercier les membres du jury d'avoir accepté d'évaluer ma thèse :

- Monsieur Hamza Teggat Maître de conférence classe A à l'université Mustapha Stambouli de Mascara pour m'avoir fait l'honneur de faire partie du jury et de le présider.
- Monsieur Réda Adjoudj professeur à l'Université Djillali Liabes Sidi-Bel-Abbès pour avoir accepté de faire partie du jury et consacré du temps pour la lecture de la thèse et son enrichissement.
- Madame Fatma Boufera Maître de conférence classe A à l'université Mustapha Stambouli de Mascara pour l'intérêt qu'elle a porté à ce travail en acceptant de faire partie du jury.

Sommaire

La peau subit des agressions externes, physiques, chimiques et bactériologiques. Elle y répond par des réactions vasculaires et tissulaires non spécifiques, suivant un mécanisme parfois allergique. Les lésions cutanées se rapportent à un terme général qui caractérise une partie de la peau ayant, en comparaison de la peau l'entourant, une croissance/apparence ou structure qui diffère de tissus considérés comme sains. Pour diagnostiquer une lésion cutanée, une variété d'indices visuels peuvent être utilisés tels que la morphologie lésionnelle individuelle, la distribution du site corporel, la couleur, la desquamation et la disposition des lésions. Lorsque les éléments individuels sont analysés séparément, le processus de reconnaissance peut être assez complexe. Pour certaines maladies, leurs lésions sont si similaires qu'elles ne peuvent être distinguées visuellement. Les limites du dépistage visuel des lésions cutanées peuvent être surmontées grâce à l'utilisation du diagnostic assisté par ordinateur (DAO). Les systèmes DAO tentent d'imiter les performances des dermatologues pour déterminer la lésion cutanée en tant que normale ou anormale. Ces systèmes utilisent des algorithmes d'extraction de caractéristiques conçus par l'homme en combinaison avec un classifieur. Cependant, lorsque nous étendons les maladies de la peau à une gamme plus large, où les caractéristiques sont si complexes que la conception de caractéristiques artisanales devient irréalisable, l'approche traditionnelle échoue. C'est pour cette raison, qu'on fait recours aux techniques d'apprentissage profond.

La percée de l'apprentissage profond a conduit à des progrès dans de nombreuses tâches de vision par ordinateur. Le problème fondamental abordé par cette thèse peut être énoncé comme une question : comment utiliser les derniers développements en apprentissage profond pour mettre en œuvre différents classifieurs capable d'examiner une image contenant une lésion cutanée et de prédire un résultat (malin ou bénin).

Notre contribution s'oriente vers l'utilisation d'un pipeline en deux étapes (extraction de caractéristiques avec apprentissage profond + classification avec apprentissage automatique conventionnel) au lieu d'un pipeline d'apprentissage profond unifié de bout en bout.

Mots clés : Lésions cutanées, Apprentissage profond, Réseau de neurones convolutionnel, Diagnostic assisté par ordinateur, Extraction de caractéristiques, Classification

ملخص

يتعرض الجلد لهجمات خارجية وفيزيائية وكيميائية وبكتريولوجية. يستجيب مع تفاعلات الأوعية الدموية والأنسجة غير المحددة، وفي بعض الأحيان يتبع آلية الحساسية. تشير الآفات الجلدية إلى مصطلح عام يميز جزءاً من الجلد له، مقارنة بالجلد المحيط، نمو/ مظهر أو بنية تختلف عن الأنسجة التي تعتبر صحية. لتشخيص الآفة الجلدية، يمكن استخدام مجموعة متنوعة من القرائن البصرية مثل مورفولوجيا الآفة الفردية، وتوزيع موقع الجسم، واللون، والتفشر، وترتيب الآفات. عندما يتم تحليل العناصر الفردية بشكل منفصل، يمكن أن تكون عملية التعرف معقدة للغاية. بالنسبة لبعض الأمراض، تكون آفاتهم متشابهة لدرجة أنه لا يمكن تمييزها بصرياً. يمكن التغلب على قيود فحص آفات الجلد المرئية من خلال استخدام التشخيص بمساعدة الكمبيوتر (CAD).

تحاول أنظمة CAO محاكاة أداء أطباء الجلد في تحديد الآفة الجلدية على أنها طبيعية أو غير طبيعية. تستخدم هذه الأنظمة خوارزميات استخراج الميزات من تصميم الإنسان جنباً إلى جنب مع المصنف. ومع ذلك، لكن عندما نوسع الأمراض الجلدية إلى نطاق أوسع، حيث تكون الميزات معقدة للغاية بحيث يصبح تصميم الميزات المصنوعة يدوياً غير ممكن، فإن النهج التقليدي يفشل. هذا هو سبب استخدام تقنيات التعلم العميق.

يمكن طرح المشكلة الأساسية التي تتناولها هذه الأطروحة على شكل سؤال: كيفية استخدام أحدث التطورات في التعلم العميق لتنفيذ مصنقات مختلفة قادرة على فحص صورة تحتوي على آفة جلدية والتنبؤ بنتيجة (خبيثة أو حميدة).

تتجه مساهمتنا نحو استخدام نهج من خطوتين (استخراج الميزات مع التعلم العميق + التصنيف مع التعلم الآلي التقليدي) بدلاً من نهج التعلم العميق الموحد الشامل.

الكلمات المفتاحية: الآفة الجلدية، التعلم العميقة، الشبكة العصبية التلافيفية، التشخيص بمساعدة الكمبيوتر، استخراج الميزات، تصنيف.

Summary

The skin is subject to external, physical, chemical and bacteriological attacks. It responds with non-specific vascular and tissue reactions, sometimes following an allergic mechanism. Cutaneous lesions refer to a general term that characterizes a part of the skin having, in comparison to the surrounding skin, a growth/appearance or structure that differs from tissue considered healthy. To diagnose a skin lesion, a variety of visual clues can be used such as individual lesion morphology, body site distribution, color, desquamation, and arrangement of lesions. When the individual elements are analyzed separately, the recognition process can be quite complex. For some diseases, their lesions are so similar that they cannot be visually distinguished. The limitations of visual skin lesion screening can be overcome through the use of computer-aided diagnosis (CAD). DAO systems attempt to mimic the performance of dermatologists in determining the skin lesion as normal or abnormal. These systems use human-designed feature extraction algorithms in combination with a classifier. However, when we extend skin diseases to a wider range, where the features are so complex that designing handmade features becomes unfeasible, the traditional approach fails. This is why deep learning techniques are used.

The breakthrough in deep learning has led to advancements in many computer vision tasks. The fundamental problem addressed by this thesis can be stated as a question: how to use the latest developments in deep learning to implement different classifiers capable of examining an image containing a skin lesion and predicting an outcome (malignant or benign).

Our contribution is moving towards the use of a two-step pipeline (feature extraction with deep learning + classification with conventional machine learning) instead of a unified end-to-end deep learning pipeline.

Keywords: Skin lesion, Deep learning, Convolutional Neural network, Computer-aided diagnosis, Feature extraction, Classification,

Table des matières

Table des matières.....	I
Liste des Figures.....	VII
Liste des Tables.....	XII
Liste des Abréviations.....	XIV
Valorisations scientifiques.....	XVI
Introduction générale.....	1
Chapitre 1 Etat de l'art sur les lésions cutanées	
1 Introduction.....	9
2 Anatomie de la peau	9
2.1 Présentation de la peau.....	9
2.2 Structure de la peau.....	10
2.2.1 Epiderme	10
2.2.2 Jonction dermo-épidermique	12
2.2.3 Derme.....	12
2.2.4 Hypoderme.....	13
2.3 Fonctions de la peau.....	13
2.3.1 Fonction protectrice	13
2.3.2 Fonction de régulation thermique.....	13
2.3.3 Fonction de sensation.....	14
2.4 Types de la peau	14
3 Lésions cutanées.....	16
3.1 Eczéma.....	16
3.2 Rosacée.....	17
3.3 Psoriasis.....	17
3.4 Zona.....	17

3.5	Acné.....	18
3.6	Cancer de la peau.....	18
3.6.1	Lésions Mélanocytaires.....	19
3.6.2	Lésions non- mélanocytaires: les carcinomes.....	21
3.6.3	Diagnostic du cancer de la peau.....	21
4	Etat de l'art sur la classification des lésions cutanées	24
5	Conclusion	27

Chapitre 2 Intelligence artificielle, Apprentissage machine, Apprentissage profond

1	Introduction.....	29
2	Intelligence artificielle, Apprentissage machine et Apprentissage profond.....	29
2.1	Intelligence artificielle.....	29
2.2	Apprentissage Machine	30
2.3	Apprentissage profond	31
1.4	Différence entre apprentissage machine traditionnel et apprentissage profond.....	33
3	Classification, régression et clustering	34
3.1	Régression	34
3.2	Classification	35
3.3	Clustering	35
4	Types d'apprentissage	35
4.1	Apprentissage supervisé.....	36
4.2	Apprentissage non supervisé	37
4.3	Apprentissage par renforcement	37
4.4	Apprentissage semi-supervisé.....	37
5	Classifieurs supervisés	37
5.1	Support Vector Machine.....	38
5.2	K-NearestNeighbor	40
5.3	Arbres de décision	42
5.4	Analyse Discriminante Linéaire	44

5.5	Naive Bayesian	45
5.6	Apprentissage d'ensemble.....	46
5.6.1	Pourquoi un apprentissage d'ensemble ?.....	48
5.6.2	Bagging	51
5.6.3	Boosting	52
5.6.4	Forêt aléatoire	53
6	Transfert Learning.....	54
7	Conclusion	56

Chapitre 3 Réseaux de Neurones Convolutionnels

1	Introduction.....	59
2	Réseaux de neurones artificiels	59
3	Présentation des réseaux de neurones convolutifs	60
4	Architecture générale d'un réseau de neurones convolutif.....	61
4.1	Couche de convolution	62
4.2	Couche de pooling.....	63
4.3	Fonction d'activation	64
4.4	Couche entièrement connectée.....	67
4.5	Fonctions de perte	67
4.5.1	Entropie croisée.....	68
4.5.2	Erreur quadratique moyenne.....	68
4.5.3	Fonction de perte hinge.....	69
5	Régularisation des réseaux de neurones convolutifs.....	69
5.1	Dropout.....	71
5.2	Drop-Weight.....	71
5.3	Weight decay.....	71
5.4	Augmentation des données.....	72
5.5	Normalisation par lots.....	72
6	Apprentissage et optimisation des réseaux de neurones convolutifs....	73
6.1	Descente de gradient.....	73
6.2	Descente de gradient stochastique.....	76

6.3	Descente de gradient stochastique avec Momentum.....	77
6.4	Descente de gradient accélérée de Nesterov.....	78
6.5	Descente de gradient par lots.....	78
6.6	Descente de gradient en mini-lots.....	79
6.7	Algorithmes avec taux d'apprentissage adaptatifs.....	80
6.7.1	AdaGrad.....	80
6.7.2	RMSprop.....	81
6.7.3	Estimation adaptative de moment.....	82
6.8	Choisir le bon algorithme d'optimisation.....	82
6.9	Apprentissage par transfert learning.....	83
7	Conclusion.....	85

Chapitre 4 Méthodologie et modèles des réseaux de Neurones convolutionnels

1	Introduction.....	87
2	Prétraitement.....	87
3	Augmentation de données.....	89
4	Extraction des caractéristiques des lésions cutanées.....	90
4.1	Hand-crafted	91
4.2	Caractéristique locale et globale	92
4.3	Caractéristiques cliniques	93
4.4	Approche CNN.....	93
5	Modèles des réseaux de neurones convolutionnels	93
5.1	LeNet-5.....	94
5.2	AlexNet	95
5.3	VGGNet	95
5.4	Google Net.....	96
5.5	InceptionV2.....	97
5.6	InceptionV3	98
5.7	ResNet	99
5.8	SqueezeNet.....	100
5.9	DarkNet.....	101

5.10	DenseNet	102
5.11	Xception	103
3.12	Inception-ResNet	104
5.13	ShuffleNet	105
5.14	MobileNet..	106
5.15	Nasnet	107
5.16	EfficientNet	108
6	Conclusion.....	109

Chapitre 5 Expérimentation et Discussion

1	Introduction	113
2	Environnement.....	113
2.1	Keras	113
2.2	Numpy	114
2.3	Pandas.....	114
2.4	SciKit Learn	114
2.5	TensorFlow	114
2.6	OpenCV	115
3	Métriques d'évaluation	115
4	Bases d'images dermoscopiques.....	117
4.1	Bases de données ISIC.....	117
4.2	Base PH2.....	120
5	Expérimentation et discussion.....	121
5.1	Approche pipeline en deux étapes avec la base ISIC 2019.....	122
5.1.1	Méthode proposée en utilisant le jeu de données original.....	124
5.1.2	Méthode proposée avec prétraitement de la base ISIC 2019.....	127
5.1.3	Méthode proposée avec augmentation de la base ISIC 2019.....	129
5.2	Réduction des caractéristiques de base ISIC 2019.....	133
5.3	Comparaison.....	137
5.4	Approche pipeline en deux étapes avec la base PH2.....	142
5.5	Approche pipeline de bout en bout avec augmentation de données	148

5.6	Approche pipeline hiérarchique.....	150
6	Conclusion.....	154
	Conclusion générale	156
	Références bibliographiques.....	159

Liste des Figures

Figure 0.1	Probabilité de survie au cancer de la peau.....	2
Figure 0.2	Quelques maladies de peau visuellement similaires.....	3
Figure 1.1	Représentation schématique de la structure de la peau.....	10
Figure 1.2	Schéma représentatif des différentes couches épidermiques..	12
Figure 1.3	Échelle de Fitzpatrick	15
Figure 1.4	Classification hiérarchique du cancer de la peau.....	18
Figure 1.5	Différents types de mélanomes	19
Figure 1.6	Niveaux d'évolution du mélanome	20
Figure 1.7	Modèle de diagnostic ABCD	23
Figure 2.1	Distinction entre IA, ML et DL	33
Figure 2.2	Différence entre apprentissage machine et apprentissage profond.....	34
Figure 2.3	Types d'apprentissage machine	36
Figure 2.4	Pipeline de l'apprentissage supervisé	36
Figure 2.5	Pipeline de l'apprentissage non supervisé	37
Figure 2.6	Classification des algorithmes supervisés utilisés	38
Figure 2.7	Illustration du principe du SVM	39
Figure 2.8	Projection des données dans un nouvel espace	40
Figure 2.9	K-Nearest Neighbor avec K=5.....	41
Figure 2.10	Structure d'un arbre de décision	43
Figure 2.11	Distribution gaussienne pour deux classe.....	47
Figure 2.12	Schéma général d'apprentissage d'ensemble	48

Figure 2.13	Principe de l'approche Bagging	52
Figure 2.14	Principe de l'approche Boosting	53
Figure 2.15	Exemple d'une forêt constituée de 6 arbres de décision	54
Figure 2.16	Différence entre approche traditionnelle et approche de transfert learning	55
Figure 3.1	Architecture générale d'un réseau de neurones artificiel.....	60
Figure 3.3	Illustration d'une convolution sur une matrice	61
Figure 3.4	Exemple d'une convolution	62
Figure 3.5	Exemple d'application du pooling	63
Figure 3.6	Illustration des opérations de convolution, de ReLU et de pooling	67
Figure 3.7	Couche entièrement connectée	67
Figure 3.8	Relation typique entre la capacité et l'erreur	70
Figure 3.9	Différentes formes d'apprentissage	70
Figure 3.10	Exemple illustratif du dropout	71
Figure 3.11	Minimum local et minimum global	74
Figure 3.12	Diagramme de la descente du gradient	74
Figure 3.13	Comparaison Adam avec d'autres optimiseurs.....	83
Figure 4.1	Méthodologie globale des approches adoptées.....	87
Figure 4.2	Exemples de deux images dermoscopiques montrant différents types de bruit.....	88
Figure 4.3	Exemple d'élimination des poils par Dull Razor.....	89
Figure 4.4	Image originale (première à gauche en haut) et quelques exemples d'augmentation de données.....	90
Figure 4.5	Différentes approches d'extraction des caractéristiques	91

Figure 4.6	Evolution des réseaux de neurones convolutifs	94
Figure 4.7	Structure du réseau LeNet	94
Figure 4.8	Structure du réseau AlexNet	95
Figure 4.9	Structure du réseau VGGNet.....	96
Figure 4.10	Structure d'un bloc de démarrage	97
Figure 4.11	Structure globale de GoogleNet	97
Figure 4.12	Structure de base du module InceptionV2.....	98
Figure 4.13	Structure de l'architecture Inception V3.....	98
Figure 4.14	Concept des blocs résiduels	99
Figure 4.15	Structure globale de l'architecture ResNet	99
Figure 4.16	Structure du bloc SE (Fire Module).....	100
Figure 4.17	Structure de l'architecture SqueezeNet.....	101
Figure 4.18	Structure de l'architecture DarkNet 19 et DarkNet 53.....	102
Figure 4.19	Structure globale de l'architecture Densenet.....	103
Figure 4.20	Structure d'une convolution séparable en profondeur.....	103
Figure 4.21	Structure de l'architecture Xception.....	104
Figure 4.22	Structure d'un bloc Inception-ResNet.....	105
Figure 4.23	Structure de l'architecture Inception-Resnet	105
Figure 4.24	Différence entre ShuffleNetV1 et ShuffleNetV2.....	106
Figure 4.25	Structure de l'architecture ShuffleNet	106
Figure 4.26	Structure de l'architecture MobileNet.....	107
Figure 4.27	Architecture d'une cellule convolutionnelle.....	108
Figure 4.28	Structure de l'architecture Nasnet	108
Figure 4.29	Structure de l'architecture EfficientNet B0.....	109

Figure 5.1	Matrice de confusion.....	116
Figure 5.2	Quelques images de la base ISIC2019.....	120
Figure 5.3	Quelques images de la base PH2.....	121
Figure 5.4	Méthodologie de l'approche pipeline en deux étapes avec la base ISIC2019.....	122
Figure 5.5	Taux de classification par DenseNet201 sans prétraitement de la base.....	126
Figure 5.6	Taux de classification par DenseNet201 après prétraitement de la base.....	129
Figure 5.7	Taux de classification par DenseNet201 après augmentation de la base.....	133
Figure 5.8	Méthodologie de l'approche réduction des caractéristiques	134
Figure 5.9	Taux de classification par DenseNet201 après réduction des caractéristiques.....	137
Figure 5.10	Taux de classification par DenseNet201 avec Cubic SVM.....	139
Figure 5.11	Taux de classification par DenseNet201 Fine KNN.....	139
Figure 5.12	Taux de classification de DenseNet201 avec différents classifieurs.....	140
Figure 5.13	Matrice de confusion de l'architecture DenseNet201 avec Cubic SVM.....	140
Figure 5.14	Matrice de confusion de l'architecture DenseNet201 avec Fine KNN.....	141
Figure 5.15	Méthodologie de l'approche pipeline en deux étapes avec la base PH2.....	143
Figure 5.16	Précision selon DenseNet201, ResNet50 et EfficientB0 avec quatre classifieurs pour mélanome ou non-mélanome.....	144
Figure 5.17	Précision selon DenseNet201, ResNet50 et EfficientB0 sans augmentation.....	145

Figure 5.18	Précision selon DenseNet201, Resnet50 et Efficient 0 avec augmentation.....	146
Figure 5.19	Architecture DenseNet201 affiné de bout en bout.....	149
Figure 5.20	Matrice de confusion de l'architecture DenseNet201 affiné de bout en bout avec augmentation de données.....	149
Figure 5.21	Organigramme de la méthode proposée.....	151
Figure 5.22	Résultats d'évaluation obtenus avec DenseNet201.....	153
Figure 5.23	Résumé des résultats obtenus avec ResNet201.....	154

Liste des Tableaux

Tableau 2.1	Différents noyaux des SVM.....	40
Tableau 4.1	Comparaison des architectures CNN.....	109
Tableau 5.1	Résumés des bases ISIC.....	118
Tableau 5.2	Distribution des différentes classes de la base ISIC 2019.....	119
Tableau 5.3	Répartition des classes en apprentissage et test.....	121
Tableau 5.4	Informations sur les différentes architectures CNN utilisées.	123
Tableau 5.5	Paramètre des classifieurs.....	124
Tableau 5.6	Résultat d'évaluation par l'utilisation de 8 architectures.....	125
Tableau 5.7	Résultat d'évaluation par l'utilisation de 9 architectures.....	125
Tableau 5.8	Résultat d'évaluation par l'utilisation de 8 architectures avec prétraitement de la base.....	127
Tableau 5.9	Résultat d'évaluation par l'utilisation de 9 architectures avec prétraitement de la base.....	128
Tableau 5.10	Différents paramètres des opérations d'augmentation utilisées.....	130
Tableau 5.11	Répartition des données avant et après l'augmentation.....	131
Tableau 5.12	Résultat d'évaluation par l'utilisation de 8 architectures avec augmentation de la base.....	131
Tableau 5.13	Résultat d'évaluation par l'utilisation de 9 architectures avec augmentation de la base.....	132
Tableau 5.14	Nombre de caractéristiques avant et après l'application de l'ACP.....	135
Tableau 5.15	Résultat d'évaluation par l'utilisation de 8 architectures avec réduction de la base.....	135

Tableau 5.16	Résultat d'évaluation par l'utilisation de 9 architectures avec réduction de la base.....	136
Tableau 5.17	Résultat d'évaluation par plusieurs classifieurs	138
Tableau 5.18	Résultat d'évaluation de notre approche avec l'état de l'art...	142
Tableau 5.19	Résultats d'évaluation du mélanome et non-mélanome.....	143
Tableau 5.20	Résultats d'évaluation du mélanome, naevus atypique et naevus commun sans augmentation.....	144
Tableau 5.21	Résultats d'évaluation du mélanome, naevus atypique et naevus commun avec augmentation.....	145
Tableau 5.22	Comparaison d'évaluation de 2 classes avec l'état d'art.....	147
Tableau 5.23	Comparaison d'évaluation de 3 classes avec l'état d'art.....	147
Tableau 5.24	Résumé des résultats sur PH2.....	147
Tableau 5.25	Résultat d'évaluation du modèle DenseNet201 affiné de bout en bout avec augmentation de données	150
Tableau 5.26	Synthèse des résultats obtenus avec Resnet séquentiel et hiérarchique50 etArchitectures DenseNet201.....	152
Tableau 5.27	Différentes augmentations effectuées sur l'ensemble d'apprentissage.....	152
Tableau 5.28	Synthèse des résultats obtenus avec ResNet50 séquentiel et hiérarchique et l'architectures DenseNet201 avec augmentation des données.....	153

Liste des Abréviations

DAO	Diagnostic Assisté par Ordinateur
ABCD	Asymétrie Bords Couleur Diamètre
CNN	Convolutional Neural Network
UV	Ultra Violet
GLCM	Gray-Level Co-Occurrence Matrix
kNN	K-Nearest Neighbor
ISIC	International Skin Imaging Collaboration
HAM10000	Human Against Machine with 10000 training images
PH2 dataset	Pedro Hispano dataset
ANN	Artificial Neural Network
SVM	Support Vector Machine
ENN	Ensemble Neural Networks
MLP	Multi Layer Perceptron
CLAHE	Contrast Limited Adaptive Histogram Equalization
SWT	Stationary wavelet Transform
GPU	Graphics Processing Unit
ML	Machine Learning
DL	Deep Learning
RVB	Rouge, vert, bleu
PCA	Analyse en Composantes Principales
MDP	Markov decision process
ADL	Analyse Discriminante Linéaire
NLP	Natural Language Processing
DRL	Deep Reinforced Learning
RL	Reinforcement learning

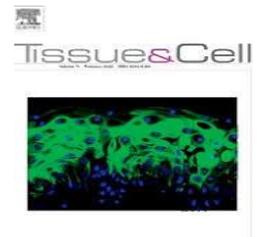
IA	Intelligence Artificielle
RNA	Réseau De Neurones Artificiels
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
ReLU	Rectified Linear Unit
PReLU	Parametric ReLU
ELU	Exponential Linear Unit
SeLU	Scaled ELU
FC	Fully Connected
SGD	Stochastic gradient descent
BGD	Batch Gradient Descent
MGD	Mini Batch Gradient Descent
RMSProp	Root Mean Square Propagation
MNIST	Mixed National Institute of Standards and Technology
RMS	Root Mean Square
BoF	Bug or Feature
SE-Network	Squeeze and Expand Network
SE	Squeeze-and-Excitation
NAS	Neural Architecture Search
API	Application Programming Interface
ACP	Analyse en Composantes Principales
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Valorisations scientifiques

Avant de commencer, mes travaux de recherche ont donné lieu à de multiples valorisations scientifiques par la publication d'articles de revue, par la participation à des conférences internationales mais également mon intégration à des projets de recherches (Projet PRFU, Projet Tassili).

Articles de revue

- Samia Benyahia, Boudjelal Meftah, Olivier Lézoray, Multi-Features Extraction Based on Deep Learning for Skin Lesion Classification, *Tissue and Cell*, Vol 74, 101701, 2022
<https://www.sciencedirect.com/science/article/abs/pii/S0040816621002172>
- Samia Benyahia et Boudjelal Meftah, Automatic diagnosis system for classification, segmentation, detection and tracking of Skin lesion based on Deep convolutional neural networks, *Models & Optimisation and Mathematical Analysis Journal*, Vol. 6 (1), pp. 23-27, 2018
<https://www.asjp.cerist.dz/en/article/71121>



Conférences internationales

- Samia Benyahia, Boudjelal Meftah, Olivier Lézoray. Hierarchical Approach for the Classification of Multi-class Skin lesions Based on Deep Convolutional Neural Networks. Third International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI 2022), Paris, France, 1-3 Juin, 2022.
<https://icprai2022.sciencesconf.org/>
- Samia Benyahia, Boudjelal Meftah, Olivier Lézoray, Skin lesion classification using convolutional neural networks based on Multi-Features Extraction, The 19th International Conference on Computer Analysis of Images and Patterns (CAIP 2021), Cyprus, Grèce, 27 Septembre-01 Octobre 2021.
<http://cyprusconferences.org/caip20211>



Projets de recherche

Projet de recherche PRFU

- **Deep learning et bio-inspirée pour l'aide au diagnostic du cancer de la peau**, C00L07UN290120220001, agréé le 01 Janvier 2022.
- Traitement des images médicales par des nouvelles approches bio-inspirées, C00L07UN290120180001, du 01 janvier 2018 au 31 Décembre 2021.

Projet PHC Tassili

- Dermato.ai, 19MDU2012, du 01 janvier 2019 au 31 Décembre 2022.

Introduction générale

La peau subit, plus que tout autre tissu, des agressions externes, physiques, chimiques et bactériologiques. Elle y répond par des réactions vasculaires et tissulaires non spécifiques, suivant un mécanisme parfois allergique. En second lieu, la peau est souvent le siège de manifestations qui dépendent de désordres internes, d'ordre infectieux, métabolique ou immunologique ; ces symptômes cutanés sont fréquemment les signes révélateurs de la maladie générale en cause (Dervaux, 2011). Les lésions cutanées se rapportent à un terme général qui caractérise une partie de la peau ayant, en comparaison de la peau l'entourant, une croissance/apparence ou structure qui diffère de tissus considérés comme sains. Ces lésions peuvent être prénatales ou post-natales, avoir de nombreuses formes et peuvent entraîner des risques différents. Le cancer de la peau est la maladie la plus fréquente et la plus connue.

À l'origine de près de 10 millions de décès en 2020, le cancer est une des principales causes de mortalité dans le monde¹. Le cancer de la peau compte 1.20 million de cas. Pour l'année 2020, le registre national du cancer estime que le nombre de nouveaux cas en Algérie atteindra la barre des 49 000 dont plus de 23000 chez l'homme et plus de 26 000 chez la femme.

Le cancer de la peau peut être classé en mélanome et non mélanome. Le mélanome est une tumeur maligne des cellules qui donnent sa couleur à la peau (mélanocytes) et peut envahir les tissus voisins. De plus, il peut se propager dans tout le corps humain et entraîner la mort. À l'inverse, le non-mélanome se propage rarement à d'autres parties du corps. Le mélanome malin est considéré comme l'une des formes les plus mortelles de cancers de la peau chez l'homme, ce qui a entraîné une augmentation du taux de mortalité.

Le mélanome est considéré comme la septième tumeur maligne la plus fréquente chez les femmes et la sixième chez les hommes. Néanmoins, c'est aussi le type de cancer de la peau le plus traitable s'il est détecté ou diagnostiqué à un stade précoce (Adamson, 2019). Avec un diagnostic précoce, le mélanome peut souvent être guéri par une simple excision ; réduisant ainsi le taux de mortalité.

En tant que type mortel de cancer de la peau, le mélanome a une incidence plus élevée avec un nombre croissant de personnes touchées. Au fil du temps, le nombre de personnes qui développeront un mélanome à un moment donné de leur vie est estimé à environ un homme sur 36 et une femme sur 55 (Adamson, 2019). Les personnes qui dépistent un mélanome au stade le plus précoce ont une probabilité

¹<https://www.who.int/fr/news-room/fact-sheets/detail/cancer>

plus élevée d'y survivre. La figure ci-dessous montre la raison et l'importance du dépistage précoce du cancer de la peau.

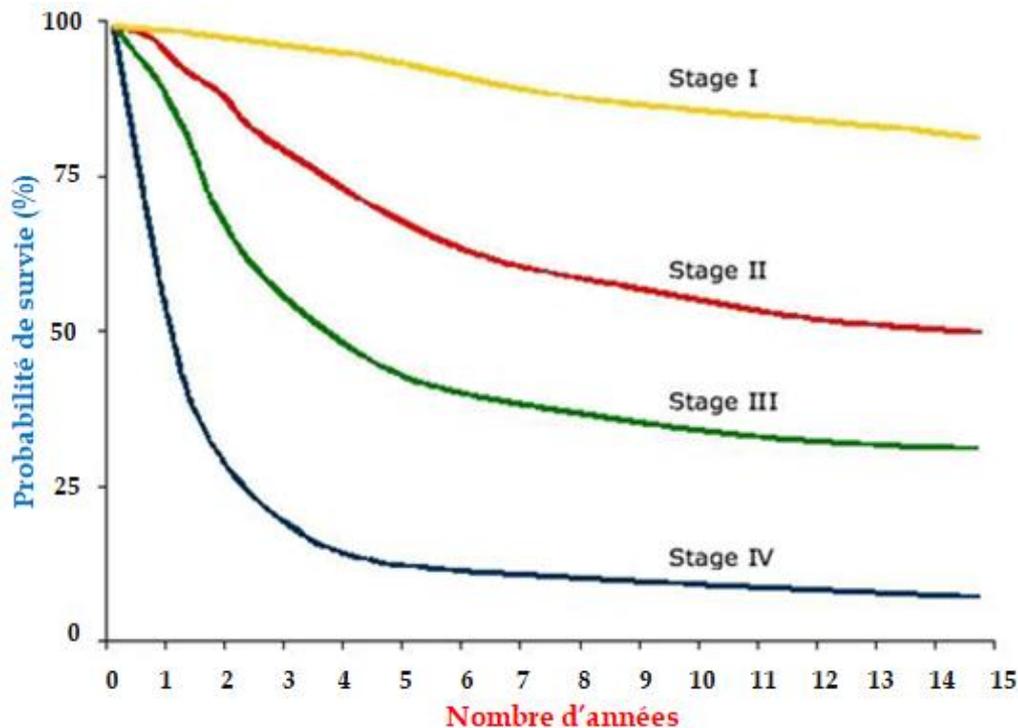


Figure 0.1 : Probabilité de survie au cancer de la peau (Travel, 2015)

La Figure 0.1 montre, ce que le dermatologue appelait, la courbe de survie ou probabilité de survie. Le cancer de la peau peut être divisé en quatre stades, et les stades plus graves peuvent être plus difficiles à traiter. Chaque couleur de la ligne de la figure présente un stade différent du cancer de la peau ; il est également montré comment les personnes précoces pourraient trouver un cancer de la peau. Ainsi, le premier stade du mélanome peut être facilement éliminé. C'est la forme la plus ancienne, qui ne pousse que sur la couche supérieure de la peau. S'il est trouvé au plus tôt, un taux de guérison de 100% peut être atteint. Certains mélanomes commencent un peu plus tard, par exemple dans 10 ans ou plus tard dans le stade, par conséquent, si le mélanome est détecté dans la période la plus précoce, les chances d'élimination, de guérison et de survie seront considérablement augmentées (Sung et al., 2021).

Énoncé du problème

Pour diagnostiquer une lésion cutanée une variété d'indices visuels peuvent être utilisés tels que la morphologie lésionnelle individuelle, la distribution du site

corporel, la couleur, la desquamation et la disposition des lésions. Lorsque les éléments individuels sont analysés séparément, le processus de reconnaissance peut être assez complexe (Cox, 2008). Par exemple, le mélanome a quatre principales méthodes de diagnostic clinique : les règles ABCD, l'analyse des modèles, la méthode de Menzies et la liste de contrôle en 7 points. Pour utiliser ces méthodes et atteindre une précision diagnostique satisfaisante, un haut niveau d'expertise est requis car la différenciation des lésions cutanées demande une grande expérience (Whited & Grichnik, 1998).

Pour certaines maladies, leurs lésions sont si similaires qu'elles ne peuvent être distinguées visuellement. La Figure 0.2 montre les images dermatologiques de huit maladies de peau différentes. Nous pouvons voir que les deux maladies dans chaque colonne ont des apparences visuelles très similaires. Ainsi, il est très difficile de porter un jugement entre les deux maladies avec uniquement les informations visuelles.



Figure 0.2 : Quelques maladies de peau visuellement similaires. Première rangée (de gauche à droite) : mélanome malin, dermatofibrome, carcinome basocellulaire et kératose séborrhéique. Deuxième rangée (de gauche à droite) : naevus composé, naevus intradermique, kératose bénigne et maladie de Bowen².

Les limites du dépistage visuel des lésions cutanées peuvent être surmontées grâce à l'utilisation du diagnostic assisté par ordinateur (DAO). Les systèmes DAO tentent d'imiter les performances des dermatologues pour déterminer la lésion cutanée en tant que normale ou anormale.

²<https://dermnet.com/>

Habituellement, on peut utiliser des algorithmes d'extraction de caractéristiques conçus par l'homme en combinaison avec un classificateur pour accomplir cette tâche. Pour certaines maladies de la peau, telles que le mélanome et le carcinome basocellulaire, cette solution est envisageable car leurs caractéristiques sont régulières et prévisibles. Cependant, lorsque nous étendons les maladies de la peau à une gamme plus large, où les caractéristiques sont si complexes que la conception de caractéristiques artisanales devient irréalisable, l'approche traditionnelle échoue. C'est pour cette raison, qu'on fait recours aux techniques d'intelligence artificielle. Le premier réflexe des techniques IA porte sur les algorithmes d'apprentissage machine (ML).

Toutefois, la plupart des algorithmes ML fonctionnent bien sur des datasets contenant jusqu'à quelques centaines de caractéristiques. Cependant, un ensemble de données non structuré, comme celui d'une image par exemple, a un grand nombre de caractéristiques. Une seule image de 800×1000 pixels en couleur RVB comporte 2,4 millions de caractéristiques. Le processus devient lourd ou complètement irréalisable par les algorithmes d'apprentissage machine traditionnels (Fleury, 2021). L'apprentissage machine traditionnel fait référence au processus d'extraction de caractéristiques à partir d'un grand ensemble de données chargé dans la machine.

Par contre, les algorithmes d'apprentissage profond apprennent progressivement plus sur l'image au fur et à mesure qu'elle traverse chaque couche du réseau de neurones.

Pourquoi l'apprentissage profond ?

Ces dernières années, le développement des techniques d'apprentissage et les progrès du matériel pour le calcul parallèle ont rendu possible l'apprentissage de modèles profonds, tels que les réseaux de neurones profonds (Bengio et al., 2018). La percée de l'apprentissage en profondeur a conduit à des progrès sans précédent dans de nombreuses tâches de vision par ordinateur telles que la classification, la détection d'objets et la segmentation. De même, il y a aussi un énorme élan de la communauté informatique d'imagerie médicale passant des approches traditionnelles à l'apprentissage en profondeur.

Plusieurs points de performance peuvent répondre à cette question, nous pouvons citer :

1. Approche d'apprentissage universel : étant donné que l'apprentissage profond a la capacité de fonctionner dans presque tous les domaines d'application
2. Robustesse : en général, des caractéristiques conçues avec précision ne sont pas requises dans les techniques d'apprentissage profond. Au lieu de cela, les caractéristiques optimisées sont apprises de manière automatisée en fonction de la tâche considérée. Ainsi, la robustesse aux changements habituels des données d'entrée est atteinte.
3. Généralisation : différents types de données ou différentes applications peuvent utiliser la même technique d'apprentissage profond, une approche fréquemment appelée apprentissage par transfert. De plus, c'est une approche utile dans les problèmes où les données sont insuffisantes.
4. Évolutivité : l'apprentissage profond est hautement évolutif. ResNet (He et al., 2016), qui a été inventé par Microsoft, comprend 1202 couches et est fréquemment appliqué à une échelle de calcul intensif.

Approche adoptée

Le problème fondamental abordé par cette thèse peut être énoncé comme deux questions

Q1 : comment utiliser les derniers développements en apprentissage profond pour mettre en œuvre différents classifieurs capable d'examiner une image contenant une lésion cutanée et de prédire un résultat (malin ou bénin) ? avec un degré de confiance suffisamment élevé pour améliorer les méthodes actuelles de détection précoce du mélanome ?

Plus précisément, il est souhaitable de disposer d'un modèle intelligent de la façon dont les lésions cutanées malignes diffèrent des lésions bénignes et d'utiliser ce modèle pour prédire - sur la base d'une photo d'un grain de beauté ou d'un patch suspect - l'apparition de lésions cutanées malignes et d'autres types de maladies qui nécessiterait une assistance médicale.

Q2 : Pourquoi faire un pipeline en deux étapes (extraction de caractéristiques avec apprentissage profond + classification avec apprentissage automatique conventionnel) au lieu d'un pipeline d'apprentissage profond unifié de bout en bout ?

Les réseaux de neurones convolutionnels (CNN) sont considérés comme une solution d'extraction de caractéristiques à partir d'images. La principale raison de considérer CNN comme le meilleur extracteur de caractéristiques est que le CNN obtient plus de caractéristiques par rapport aux autres méthodes. Il a la capacité de capturer la qualité supérieure et les fonctionnalités puissantes et d'un grand volume de données brutes ainsi que l'amélioration de la précision en moins de temps.

L'avantage d'utiliser un extracteur de caractéristiques est la possibilité de choisir le classifieur qui correspond le mieux aux données en tenant compte du problème, ainsi que la possibilité de combiner le CNN avec les différents classifieurs et l'intégration d'innombrables caractéristiques.

De nombreux chercheurs ont démontré l'efficacité de l'apprentissage profond en tant que méthode d'extraction de caractéristiques ces dernières années. De plus, de nombreux travaux dans différentes tâches (Deepika & Yamini, 2020 ; Barata et al., 2019), montrent l'efficacité de l'utilisation du classifieur de l'apprentissage automatique pour classer les données en fonction des caractéristiques extraites via le CNN par rapport à l'apprentissage profond de bout en bout.

Choisir le bon classifieur et la bonne architecture CNN comme extracteurs de caractéristiques a toujours été une tâche difficile. C'est à partir de cet objectif que nous avons développé notre thèse et contribuer à trouver la solution adéquate (CNN+Classifieur).

Notre contribution s'oriente vers l'utilisation d'un pipeline en deux étapes (extraction de caractéristiques avec apprentissage profond + classification avec apprentissage automatique conventionnel) au lieu d'un pipeline d'apprentissage profond unifié de bout en bout.

Limites de l'approche bout en bout (Fine tuning)

Nombreuses sont les raisons qui font de l'approche bout en bout une option irréalisable dans différents cas. L'apprentissage de bout en bout peut être très inefficace pour apprendre des modèles de réseaux de neurones composés de plusieurs modules non communs. L'apprentissage de bout en bout peut même s'effondrer complètement ; dans le pire des cas aucun des modules ne parvient à apprendre. En revanche, chaque module est capable d'apprendre si les autres modules sont déjà entraînés (Glasmachers, 2017).

Organisation de la thèse

Cette thèse débute par une présentation de la peau, l'organe majeur de cette étude, réalisée au sein du Chapitre 1. Ainsi, divers éléments de base de la peau sont décrits à savoir les aspects qui permettent une bonne compréhension de la physiologie de la peau, de son fonctionnement et de ses rôles. Dans ce chapitre exclusivement dédié à cet organe vital sont abordées les multiples pathologies pouvant l'affecter et l'altérer. Le chapitre se termine par un état de l'art sur la classification des lésions cutanées par les algorithmes d'apprentissage automatique.

Puis, s'ensuit dans le Chapitre 2 l'ensemble des connaissances d'intelligence artificielle sollicitées dans ce manuscrit en abordant le machine learning et le deep learning. Ce chapitre dresse une description théorique de six classifieurs issus de l'apprentissage machine avec leurs variantes. Enfin, les aspects liés au principe de transfert de connaissances sont proposés dans la dernière section.

Ensuite, le Chapitre 3 présente au lecteur l'ensemble des connaissances des réseaux de neurones convolutifs sollicitées dans ce manuscrit. Dans le même contexte la régularisation ainsi que l'apprentissage et l'optimisation des réseaux de neurones convolutifs sont discutées.

Le chapitre 4 présente la méthodologie globale des approches adoptées et mise en œuvre dans la partie expérimentale. Plus précisément, il fournit l'architecture des réseaux de neurones convolutifs (CNN) comme extracteur de caractéristiques associé à de nombreux classifieurs d'apprentissage automatique pour évaluer la classification des lésions cutanées à partir des images dermoscopiques.

Le Chapitre 5 expose la partie expérimentale associée aux différents classifieurs et aux différentes architectures des réseaux de neurones convolutionnels. La mise en œuvre de l'approche est appliquée sur deux bases d'images dermoscopiques ISIC 2019 et PH2. Une discussion et des comparaisons avec les états de l'art sont effectuées.

Enfin, le chapitre 6 fournit des conclusions, résume les contributions à la recherche et identifie les orientations pour les travaux futurs.

Chapitre 1 : Etat de l'art sur les lésions cutanées

1.	Introduction.....	9
2.	Anatomie de la peau	9
3.	Lésions cutanées.....	16
4.	Etat de l'art sur la classification des lésions cutanées.....	24
5.	Conclusion	27

1. Introduction

Ce chapitre est consacré à la description de l'organe sujet de cette thèse, à savoir la peau et ses maladies. Ainsi, divers éléments de base de la peau sont décrits pas à pas à savoir les aspects qui permettent une bonne compréhension de la physiologie de la peau, de son fonctionnement et de ses rôles. Dans ce chapitre exclusivement dédié à cet organe vital sont abordées les multiples pathologies pouvant l'affecter et l'altérer.

Dans un premier temps, une description des principales couches et composantes de la peau est réalisée par profondeur croissante. Dans un second temps, ce travail consacre une section dédiée à la présentation de quelques-unes de ses principales lésions.

Le chapitre se termine par un état de l'art sur la classification des lésions cutanées par les algorithmes d'apprentissage automatique.

2. Anatomie de la peau

2.1. Présentation de la peau

La peau, appelée aussi « tégument » du latin « tegumentum, couverture » (Mélissopoulos, 2012), est souvent présentée comme étant le plus grand et le plus multifonctionnel organe de l'organisme humain avec une moyenne de 16 % de la masse totale d'un individu adulte, elle constitue l'organe le plus lourd et le plus étendu avec un poids de 5kg, une surface de 2m² et une épaisseur qui varie entre 1 et 5 mm selon les endroits du corps. Elle se compose de 70% d'eau, de 27.5% de protéines, de 2% de matières grasses, de 0.5% de sels minéraux et d'oligo-éléments. La peau n'est pas qu'une simple enveloppe qui recouvre notre corps, elle est en effet une barrière entre l'environnement extérieur et l'intérieur du corps, dont le rôle principal est d'assurer la protection du corps contre les agressions extérieures, qu'elles soient : physiques, lumineuses, thermiques, chimiques ou microbiennes et infectieuses (Mélissopoulos, 2012). Elle assure en parallèle d'autres fonctions vitales biologiques et biochimiques telles que les fonctions : de thermorégulation, sensorielle, métabolique, immunitaire, d'échanges, d'autoréparation ou de cicatrisation, de réduction de la pénétration des rayons ultraviolets nocifs, de protection contre la déshydratation ainsi que de synthèse de la vitamine D (Someya & Amagai, 2019).

2.2. Structure de la peau

Fondamentalement, la structure de la peau est une structure hétérogène constituée essentiellement de trois tissus superposés de l'extérieur vers l'intérieur (Figure 1.1). Le tissu le plus externe est l'épiderme (la peau) suivi par la jonction dermo-épidermique (le tissu intermédiaire), par le derme et le tissu le plus profond, l'hypoderme. A ces trois composantes s'ajoutent les annexes cutanées représentées par les glandes et les phanères (Gallo, 2017).

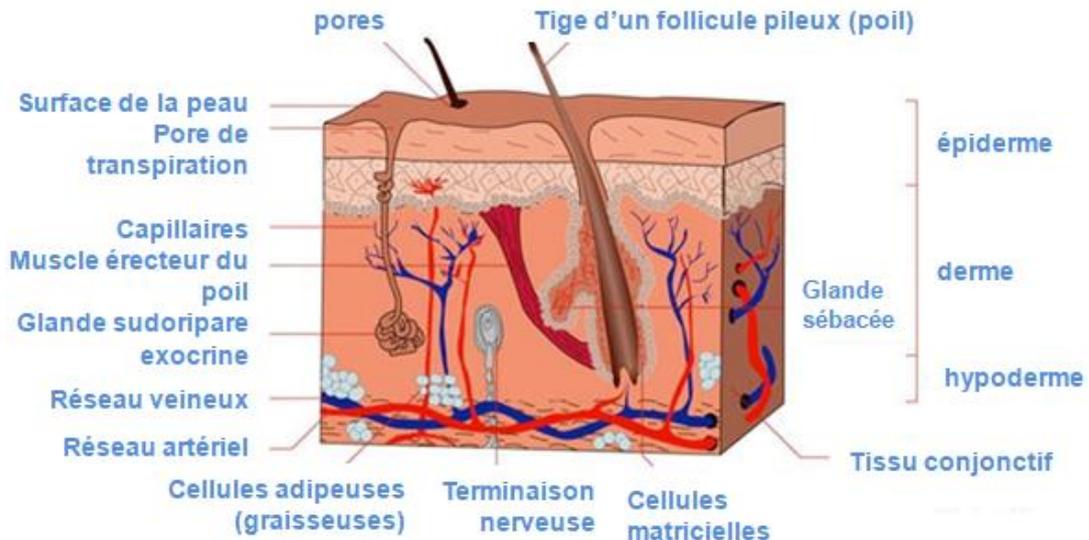


Figure 1.1 : Représentation schématique de la structure de la peau³.

2.2.1. Epiderme

L'épiderme est la partie superficielle la plus externe de la peau (van Smeden et al., 2014), c'est un tissu épithélium, pluristratifié kératinisé, non vascularisé, sa surface comporte des orifices pilo-sébacés, des pores et des dépressions (empreintes digitales, rides, plis). L'épiderme est responsable de la couleur de la peau, de sa texture, il maintient une hydratation satisfaisante de la peau et assure la protection et la régénération de celle-ci.

L'épiderme est constitué, de quatre types de cellules avec différentes fonctions (Figure 1.2):

- Les kératinocytes : cellules majoritaires primaires avec un taux de 80% de l'ensemble des cellules épidermiques, responsable de la kératinisation et de la protection extérieure. Les 20% de cellules restantes, sont dispersés entre les kératinocytes. Ce sont : les mélanocytes, les cellules de Langerhans et les cellules de Merkel.

³<https://infovisual.info/fr/corps-humain/peau>

- Mélanocytes : cellules situées dans la couche basale, elles sont responsables de la pigmentation, leur fonction principale est la production de mélanine par les mélanosomes, le mélanocyte processus dendritiques permet la sécrétion de mélanine qui est ensuite captée par les cellules kératinocytes adjacentes, réparties sur les noyaux cellulaires pour protéger l'ADN des rayons ultraviolets.
- Cellules de Langerhans ou macrophage : cellules ayant un rôle immunitaire local, caractérisées par des processus dendritiques similaires aux mélanocytes. Les cellules de Langerhans sont très sensibles aux rayons UV ce qui provoque la réduction des cellules et par la suite la diminution conséquente de l'immunologie cutanée qui peut développer un environnement plus permissif pour le développement d'un carcinome.
- Cellules de Merkel : cellules situées dans la couche basale, contiennent des granules de sécrétions associées aux terminaisons nerveuses périphériques, son rôle est la perception sensorielle.

Parmi les principales caractéristiques de la couche épidermique c'est le fonctionnement varié de ces cellules, le processus de kératinisation qui se subdivise en sous-couches aux propriétés diverses afin d'assurer la maturation des kératinocytes de la couche inférieure vers la couche supérieure : la couche basale (stratum basale), la couche d'épineuse (stratum spinosum), la couche granuleuse (stratum granulosum), la couche claire (stratum lucidum, uniquement dans la peau épaisse), la couche cornée (stratum corneum).

En fin de parcours, les kératinocytes perdent leur cohésion et leur noyau ce qui provoque leur mort, en subissant le processus de desquamation, la peau se régénère au fur et à mesure.

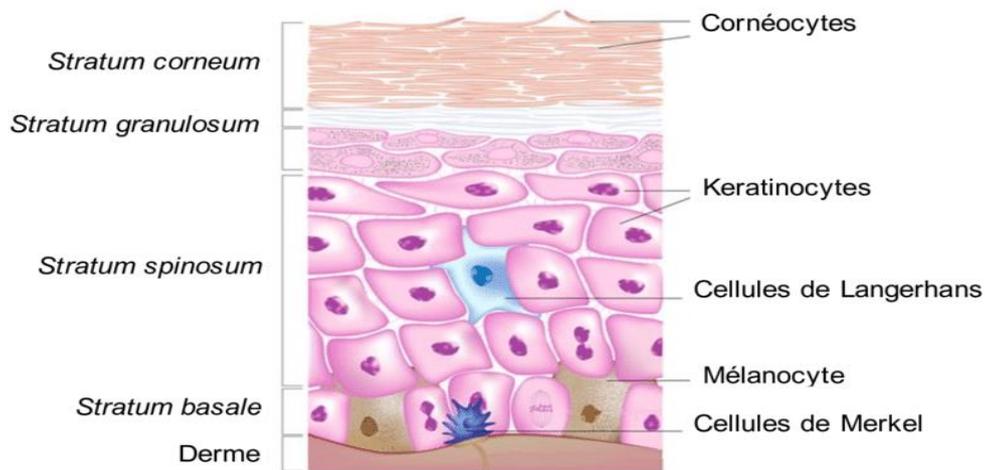


Figure 1.2: Schéma représentatif des différentes couches épidermiques (van Smeden et al., 2014).

2.2.2. Jonction dermo-épidermique¹

La Jonction dermo-épidermique ou membrane basale se situe à la jonction de l'épiderme et du derme, dont le rôle est d'assurer la cohésion entre ces deux couches, la nutrition des cellules de base de l'épiderme et la ré-épidermisation essentielle à la cicatrisation. La jonction dermo-épidermique est scindée en deux sous couches : la lame basale et la lame réticulaire.

2.2.3. Derme

Le derme est un tissu conjonctif riche en vaisseaux sanguins qui se situe entre l'épiderme et l'hypoderme, il est considéré comme étant un tissu de soutien de l'épiderme. Il est responsable de la variation régionale de l'épaisseur de la peau, assure l'approvisionnement en oxygène et nutriments à toutes les structures du système tégumentaire et restitue les déchets résultant de leur activité. Il contient également, les glandes annexes cutanées et les récepteurs sensoriels qui détectent la douleur, la pression, les vibrations et les sensations thermiques.

Le derme est composé principalement de cellules de type fibroblaste responsable de la production du collagène et de l'élastine ainsi que d'autres protéines. Il contient également les vaisseaux sanguins, les nerfs et les glandes sudoripares. (Mélissopoulos, 2012). Le collagène est une famille de protéines présentes dans la peau et le tissu conjonctif du corps humain, ses propriétés structurales et biologiques lui confèrent un rôle élémentaire dans la construction et l'architecture de nombreux tissus. Il contribue à la résistance de la peau, à sa fermeté et à son aspect lisse. Les fibres d'élastine confèrent à la peau son élasticité.

Le derme est divisé en deux régions qui diffèrent par leur composition et organisation :

- Le derme papillaire est une couche sous épithéliale qui se situe sous la jonction dermo-épidermique et contient un mélange serré de fibres élastiques, de collagène et de vaisseaux sanguins.
- le derme réticulaire se situe en dessous du derme papillaire, il contient moins de fibres élastiques mais une collection plus dense de collagène en faisceaux.

2.2.4. Hypoderme

L'hypoderme est la couche grasseuse sous-cutanée la plus profonde sous le derme, constitué de tissu conjonctif très riche en lobules graisseux, vaisseaux sanguins et nerfs. Il assure la mobilité de la peau sur les structures sous-jacentes telles que les muscles et les os, comme il assure l'isolation thermique.

2.3. Fonctions de la peau

2.3.1. Fonction protectrice

La peau exerce d'abord une fonction protectrice, vis à vis de toutes les agressions venant de l'extérieur (Mélissopoulos, 2012).

- Contre les agressions mécaniques : les mouvements du corps, les chocs. La peau assure une protection souple et efficace.
- Contre les agressions chimiques. La couche cornée et le film formé par la sueur et le sébum forment une barrière, mais qui n'est pas absolument infranchissable. Certains produits peuvent pénétrer lentement à travers la peau pour gagner le système circulatoire en général. A l'inverse, elle représente une protection vis à vis de produits toxiques comme le mercure.
- Contre les microbes, la défense de la peau est assurée par la couche cornée, les cellules immunitaires du derme, et aussi par la flore microbienne normale qui existe à la surface de la peau.
- Contre les rayons du soleil, la peau bénéficie de la double protection de la couche cornée, qui s'épaissit, et de la mélanine, dont la sécrétion augmente colore naturellement la peau et la protège.

2.3.2. Fonction de régulation thermique

La peau est un organe essentiel de la régulation thermique. Elle contribue efficacement au maintien de la température constante de notre corps. Contre le chaud, l'évacuation de l'excès de chaleur est assurée par la dilatation active des

petits vaisseaux du derme, et par la sudation. Contre le froid, le resserrement des petits vaisseaux de la peau, le coussin graisseux de l'hypoderme, la contraction des petits muscles des poils (chair de poule) s'opposent au refroidissement de l'organisme (Someya&Amagai, 2019).

2.3.3. Fonction de sensation

La peau, par sa richesse en fibres sensibles, informe l'organisme sur quatre grands groupes de sensations : le toucher, la douleur, la température et la pression. Premier organe sensoriel à apparaître au cours de l'évolution, le sens tactile est aussi le dernier à disparaître au cours du vieillissement.

2.4. Types de la peau

Il existe quatre types de peaux : peau normale, sèche, grasse ou mixte (Mélissopoulos, 2012). Le type de peau est déterminé par notre patrimoine génétique. L'état de santé de notre peau peut cependant considérablement varier en fonction des divers facteurs internes et externes auxquels elle est exposée.

- **Peau normale** : Le terme « peau normale » est largement utilisé pour décrire une peau équilibrée. Le terme scientifique désignant une peau saine est « eudermique ».
- **Peau sèche** : Le terme « peau sèche » est utilisé pour décrire un type de peau qui produit moins de sébum qu'une peau normale. À cause de cette carence en sébum, une peau sèche ne dispose pas des lipides dont elle a besoin pour retenir l'humidité et construire une barrière protectrice contre les agressions extérieures.
- **Peau grasse** : Le terme « peau grasse » est utilisé pour décrire un type de peau dont la production de sébum est accrue. Une surproduction s'appelle une séborrhée.
- **Peau mixte** : Comme son nom le suggère, une peau mixte représente un mélange de deux types de peaux.

Thomas Fitzpatrick (Fitzpatrick, 1988) a développé une échelle de six phototypes pour la classification des profils type de peau et décrivant la réaction typique à l'exposition au soleil et le risque de cancer. Par conséquent, certaines conditions sont plus susceptibles de se produire sur certains types de peau, de plus, une même pathologie peut avoir des caractéristiques différentes.

Les six types de peau différents selon l'échelle de Fitzpatrick sont les suivants:

- **Type 1** (type roux) correspond à une peau très blanche, des cheveux blonds ou roux et des yeux bleu-vert avec souvent des taches de rousseur ou éphélides. Les coups de soleil sont systématiques, la peau ne bronze jamais et rougit toujours.
- **Type 2** (type nordique) correspond à une peau claire, des cheveux blond roux à châtain et des yeux clairs à bruns avec parfois des taches de rousseur. Les coups de soleil sont fréquents, la peau bronze à peine ou très lentement.
- **Type 3** (type le plus répandu) correspond à une peau intermédiaire, des cheveux châtain à bruns et des yeux bruns. Les coups de soleil sont occasionnels et la peau bronze, d'un hâle brun clair.
- **Type 4** (type méditerranéen) correspond à une peau claire mate et des cheveux et yeux bruns ou noirs. Les coups de soleil sont rares. La peau bronze bien, d'un brun foncé.
- **Type 5** (type latino) correspond à une peau claire mate, des cheveux et des yeux noirs. Les coups de soleil sont rares, la peau bronze bien, d'un brun foncé.
- **Type 6** (type africain) correspond à une peau très foncée, des cheveux et des yeux noirs. Les coups de soleil sont très exceptionnels.

La **Figure 1.3** fournit un résumé visuel des types de peau identifiés par Fitzpatrick, ainsi que la résistance aux UV.

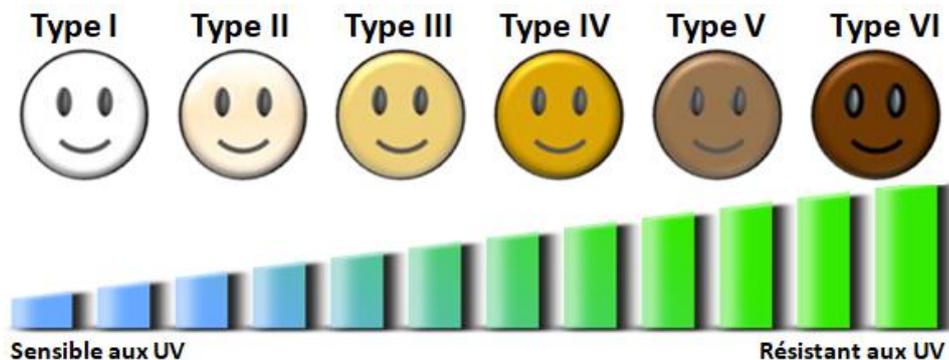


Figure 1.3 : Échelle de Fitzpatrick.

Selon l'échelle de Fitzpatrick, ceux avec un type de peau de I ou II ont besoin de plus de protection UV et peuvent être plus à risque de développer un cancer de la peau, ils devraient subir des examens cutanés réguliers. Tandis que le type VI par exemple y est moins sensible.

3. Lésions cutanées

La peau subite, plus que tout autre tissu, des agressions externes, physiques, chimiques et bactériologiques. Elle y répond par des réactions vasculaires et tissulaires non spécifiques, suivant un mécanisme parfois allergique. En second lieu, la peau est souvent le siège de manifestations qui dépendent de désordres internes, d'ordre infectieux, métabolique ou immunologique ; ces symptômes cutanés sont fréquemment les signes révélateurs de la maladie générale en cause (Dervaux, 2011).

Une lésion est une modification pathologique d'une cellule ou d'un tissu d'un organisme provoquée par une maladie, un accident ou tout autre dommage tissulaire. Les lésions cutanées se rapportent à un terme général qui caractérise une partie de la peau ayant, en comparaison de la peau l'entourant, une croissance/apparence ou structure qui diffère de tissus considérés comme sains. Ces lésions peuvent être prénatales ou post-natales, avoir de nombreuses formes et peuvent entraîner des risques différents.

Le domaine de la dermatologie est trop vaste pour que l'on puisse ici passer en revue toutes les affections de la peau. L'eczéma, la rosacée, le psoriasis, le zona, l'acné et le cancer de la peau sont les maladies de la peau les plus fréquentes et les plus connues (Sellam, 2008).

3.1. Eczém

L'eczéma constitue la maladie de la peau la plus répandue (Pomaredé, 2016). Cette affection peut toucher n'importe quelle zone de la peau. Il existe différentes formes d'eczéma mais la plus courante est l'eczéma atopique.

Les symptômes les plus courants sont les démangeaisons, les rougeurs, la présence de vésicules ou cloques ainsi que la peau sèche. La peau enflammée peut devenir rouge sur une peau claire, et brun foncé, violet ou gris sur une peau foncée. L'eczéma atopique peut toucher n'importe quelle partie du corps mais le plus souvent ce sont les mains, l'intérieur des coudes, l'arrière des genoux, le visage et le cuir chevelu.

Les poussées d'eczéma sont favorisées par la sécheresse cutanée, qui empêche la peau de jouer son rôle de barrière, et laisse les allergènes pénétrer plus facilement. Il peut également y avoir une composante génétique et elle se développe souvent en même temps que d'autres affections telles que l'asthme et le rhume des foins.

3.2. Rosacée

La rosacée est une maladie de peau courante qui touche principalement le visage. Elle est plus fréquente chez les femmes et les personnes à la peau claire, bien que les symptômes puissent être plus graves chez les hommes. Elle provoque des rougeurs sur le nez, les joues, le front et le menton. Il peut également y avoir de minuscules vaisseaux sanguins brisés et de petits bosses rosés remplies de pus. Les autres symptômes peuvent être un gonflement autour des yeux, des paupières douloureuses et un épaissement de la peau du nez. De nombreux cas sont bénins et tout le monde ne présente pas tous les symptômes (Sellam, 2008).

Certains facteurs déclenchants peuvent aggraver les symptômes, en particulier l'alcool et les aliments épicés, mais aussi le fromage, la caféine, les boissons chaudes, l'exercice vigoureux, le stress ou encore le soleil.

3.3. Psoriasis

Le psoriasis est une inflammation chronique de la peau. Il existe plusieurs formes de psoriasis, mais en général, cette maladie se caractérise par l'apparition d'une ou plusieurs plaques rouges bien délimitées au niveau des zones de frottement de la peau comme les coudes ou les genoux, recouvertes de fragments de peau blanchâtre et sèche qui se détachent (squames). Lorsque l'on gratte ou décape ces plaques, il ne subsiste que la rougeur de la peau (Pagano, 2022).

Le psoriasis est une maladie auto-immune, qui pousse l'organisme à attaquer ses propres tissus, entraînant une augmentation de la production de cellules cutanées. Cela crée une accumulation de plaques de peau sèche et squameuse.

3.4. Zona

Le zona est une infection qui provoque une éruption cutanée douloureuse. La fréquence et la gravité de la maladie augmente très nettement avec l'âge (Sellam, 2008).

Le premier signe du zona est une sensation de picotement douloureux et des maux de tête. Quelques jours plus tard, une éruption cutanée rouge, prurigineuse et tachetée apparaît sur un seul côté du corps, généralement sur la poitrine et l'estomac, mais elle peut également toucher le visage, les yeux. L'éruption se transforme alors en cloques qui démangent et qui suintent.

3.5. Acné

L'acné, ou l'apparition de boutons, est un phénomène lié à l'inflammation des glandes sébacées sous la peau. Ces lésions peuvent atteindre toutes les parties du corps humains mais préférentiellement le visage, la poitrine et le dos. L'acné est un problème commun à la puberté en raison du bouleversement hormonal, mais il est possible d'avoir de l'acné à tout âge (Sellam, 2008).

L'acné a aussi de nombreuses causes, le plus souvent due à des variations hormonales mais elle peut aussi se déclencher à cause de l'alimentation, l'hérédité, la prise de médicaments ou encore d'autres maladies.

3.6. Cancer de la peau

Le terme « cancer de la peau » correspond à la présence de cellules anormales au sein de la peau, cellules qui se multiplient de façon anarchique pour former une tumeur (un « carcinome »). Selon le type de cancer, ces cellules peuvent rester localisées dans la peau (cancers dits « in situ » ou « localisés ») ou envahir les ganglions proches de la tumeur, voire d'autres organes (mélanomes dits « invasifs »).

On peut diviser les cancers de la peau en deux catégories principales selon leur origine (Boonen, 2016): les non-mélanomes et les mélanomes, ou selon leur degré de malignité : bénigne ou maligne (Figure 1.4).

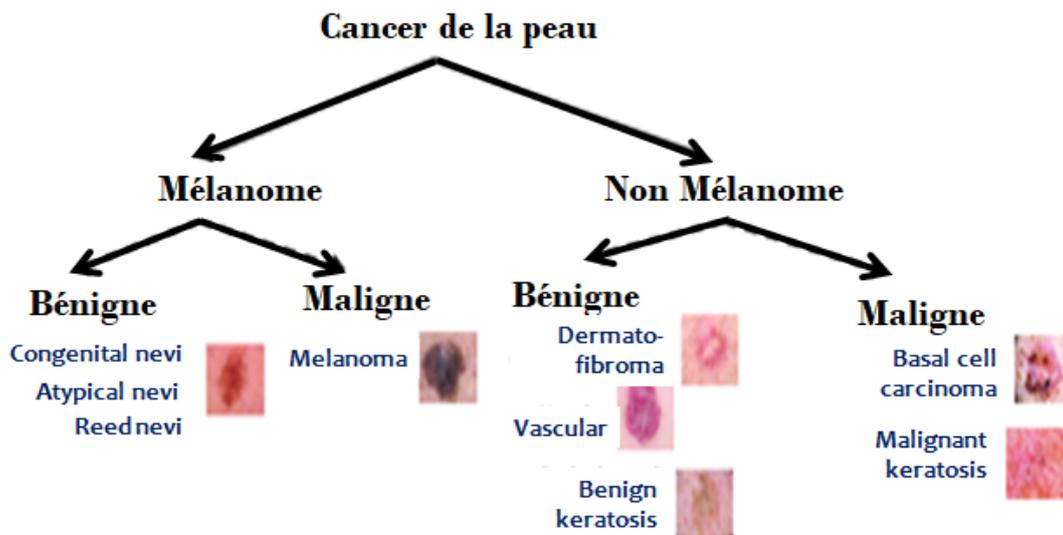


Figure 1.4 : Classification hiérarchique du cancer de la peau.

3.6.1. Lésions Mélanocytaires

Ce sont des lésions constituées de cellules mélanocytaires qui élaborent le pigment mélanique, elles se réfèrent à des lésions qui sont de couleur brune, noire ou bleue, la prolifération de ces mélanocytes est responsable des tumeurs bénignes (naevus) et malignes (mélanomes). La **Figure 1.5** présente des exemples de mélanomes.



Figure 1.5 : Différents types de mélanomes.

Lésions mélanocytaires bénignes

Ces lésions sont considérées comme sans conséquences graves sur la vie des individus.

Les lésions mélanocytaires bénignes à fort risque de transformation en lésion maligne, ainsi que leur aspect inhabituel, peuvent faire porter à tort le diagnostic. L'une des lésions bénignes les plus fréquentes du système mélanocytaires est le naevus qui se présente sous forme circulaire ou ovale, il se développe à partir des mélanocytes qui se regroupent dans l'épiderme et /ou le derme. Parmi ces lésions, peuvent être cités :

- Le naevus commun acquis (grain de beauté) est l'une des lésions les plus fréquentes composées de mélanocytes qui se regroupent en amas dans l'épiderme ou le derme (naevus épidermiques, jonctionnels, dermiques).
 - Le naevus atypique caractérisé par une pigmentation variable et des bordures irrégulières ce qui pose le problème du diagnostic différentiel avec les tumeurs malignes.
- Le naevus congénital est caractérisé par des lésions (sous forme de taches) de couleur brun clair à foncé, parfois bleutées, plus ou moins pileuses.

- Le nævus de Spitz est une variante rare qui s'apparente souvent à un mélanome de par sa couleur ainsi que sa croissance rapide.

Lésions mélanocytaires malignes

Ce sont des mélanomes, des tumeurs malignes résultant d'une croissance cancéreuse des cellules cutanées pigmentées. Les mélanomes représentent la forme la plus mortelle de cancer de la peau, avec plus de 75% de tous les décès par cancer de la peau (Leiter et al., 2020). Le mélanome peut évoluer généralement en deux phases (Figure 1.6), horizontalement avec prolifération des mélanocytes dans l'épiderme de façon anarchique, puis verticalement où les cellules tumorales envahissent le derme progressivement (Clark et al., 1969).



Figure 1.6 : Niveaux d'évolution du mélanome.

La Figure 1.6 montre différents stades du mélanome. La tumeur commence comme une lésion brune confinée dans la couche supérieure de la peau. Au fil du temps, le mélanome commence à pénétrer de plus en plus profondément dans la peau au-delà de la couche supérieure de l'épiderme et se développe dans le derme réticulaire. En fin de compte, les cellules tumorales pénètrent plus profondément dans la peau et envahissent les vaisseaux lymphatiques ou se développent dans les vaisseaux de la motte de la peau. Lorsqu'il atteint le stade quatre, le mélanome peut avoir commencé à se propager au-delà de la peau, d'abord aux ganglions lymphatiques, puis il peut se propager dans tout le corps, comme le cerveau, le foie et les poumons. En ce qui concerne le stade quatre, il s'agit du mélanome le plus évolutif, et c'est là que les chances de survie deviennent plus faibles.

3.6.2. Lésions non- mélanocytaires : les carcinomes

Le terme « carcinome » désigne les tumeurs malignes d'origine épithéliale⁴. Le carcinome basocellulaire et le carcinome épidermoïde sont les deux formes les plus courantes de non-mélanomes. Ils surviennent habituellement chez les personnes âgées de plus de 50 ans.

Le carcinome basocellulaire, qui représente 70 % des cancers de la peau, se forme le plus souvent sur le visage, chez les personnes à peau claire et de plus de 50 ans. Il peut également apparaître sur la peau du cou, des bras ou du dos des mains. C'est le cancer de la peau le moins grave, son évolution est lente et il n'est jamais invasif (il ne produit jamais de métastases). Le carcinome basocellulaire est lié à l'exposition aux rayons du soleil. Il se traduit par une lésion rosée de quelques millimètres qui ne guérit pas.

Le carcinome épidermoïde cutané représente 20% des cancers de la peau. Il se développe à partir de lésions dites « précancéreuses » (également appelées « kératoses actiniques »), en particulier au niveau des cicatrices de brûlures et des plaies chroniques. Favorisé par l'exposition au soleil, le carcinome épidermoïde cutané peut se traduire par une plaie, un bouton ou une croûte qui ne guérit pas. Ces cancers de la peau ne deviennent invasifs que très rarement, dans moins de 1% des cas.

3.6.3. Diagnostic du cancer de la peau

Le cancer de la peau peut être diagnostiqué de plusieurs façons :

- **Auto-examen visuel** : La plupart des lésions cutanées sont visibles pour la plupart des gens, et le patient peut faire un auto-examen en regardant autour de son corps pour trouver quoi que ce soit d'anormal en utilisant un miroir ou quelqu'un qui pourrait l'aider avant de consulter un médecin. Le dermatologue recommande de faire ce contrôle régulièrement, afin qu'une lésion évolutive puisse être détectée au plus tôt. De nombreuses études ont montré que la plupart des mélanomes cancéreux de la peau sont en fait détectés pour la première fois par le patient (McGuire et al., 2011), et des études ont montré que les auto-examens cutanés, y compris la vérification de la lésion (taille et profondeur du mélanome) peuvent augmenter le taux de

⁴Epithélium est la structure histologique constitutive de la peau et de certaines muqueuses.

survie. Il s'est avéré que les personnes qui effectuaient les auto-examens 1 à 11 fois par an avaient un faible risque statistiquement significatif de former des mélanomes (Titus et al., 2013). Par conséquent, c'est une bonne preuve que l'auto-examen de la peau fonctionne.

- **Règle ABCD** : La règle ABCD développée par Stolz (Stolzet al., 1994), à la fois simple à mémoriser et à appliquer, a été conçue pour promouvoir l'auto-examen. Sur la base des caractéristiques cliniques du mélanome précoce, la règle ABCD évalue les lésions cutanées en fonction de 4 caractéristiques (Figure 1.7) : (A) symétrie, (B) Bords, (C) Couleur inhomogène et (D) diamètre.
 - **A** signifie asymétrie. Les deux moitiés du mélanome n'ont pas la même forme.
 - **B** signifie bord. Le contour du mélanome est inégal (irrégulier). Il peut sembler dentelé, échancré ou flou.
 - **C** signifie couleur. La couleur du mélanome n'est pas la même partout variant d'une zone à l'autre. Elle peut présenter des teintes de brun clair, de brun et de noir.
 - **D** signifie diamètre. La taille du mélanome est supérieure à 6 mm de diamètre.

Si la lésion devient plus grande que l'apparence initiale, alors cette lésion doit être évaluée par un dermatologue (Griffiths et al., 2016).

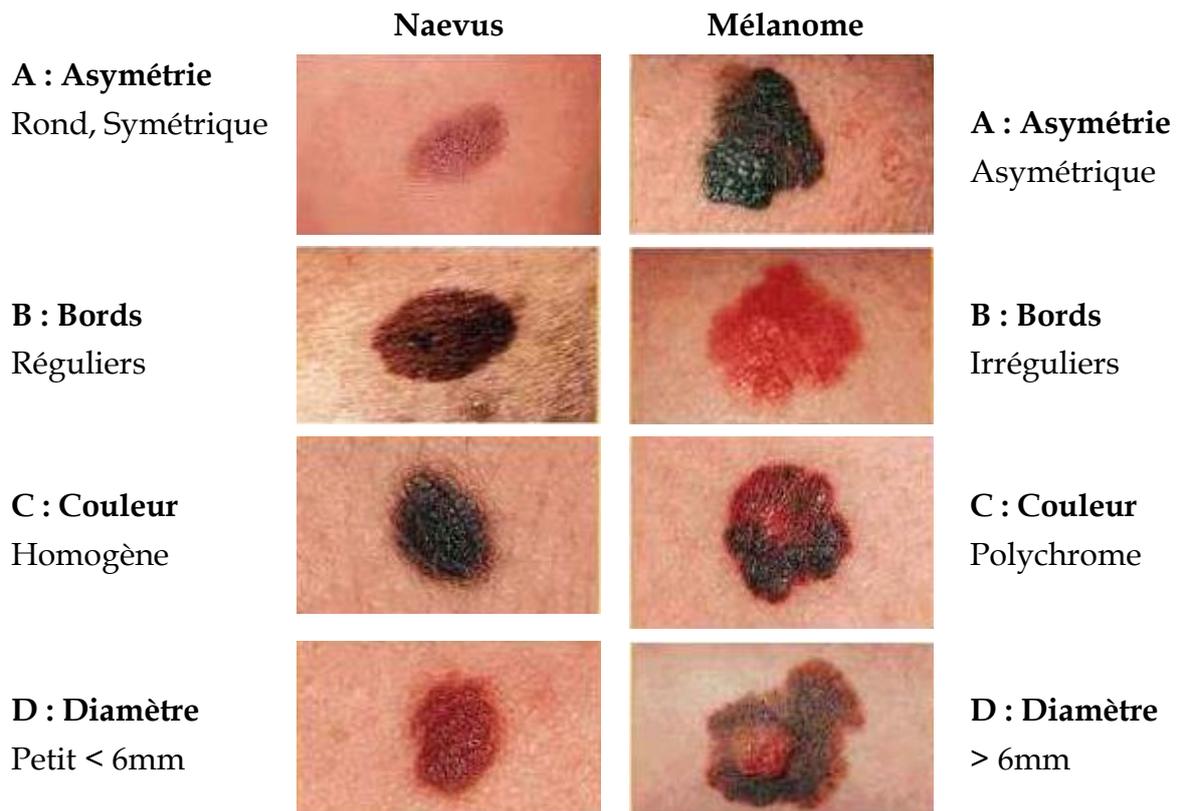


Figure 1.7 : Modèle de diagnostic ABCD.

- **Liste de 7 points à vérifier** : La liste de 7 points à vérifier peut permettre aux professionnels de la santé de détecter des changements susceptibles d'être un cancer de la peau (Argenziano et al, 1998). Ils attribuent à chaque caractéristique présente 1 ou 2 points selon son type. Un score final est calculé en additionnant la valeur en points de chaque critère présent. Si le score est supérieur ou égal à 3, alors la lésion est classée comme mélanome.

Les caractéristiques sont divisées en 2 groupes, soit les caractéristiques principales et les caractéristiques secondaires.

Caractéristiques principales (2 points chacune):

- changement de taille
- couleur inégale (irrégulière)
- forme inégale

Caractéristiques secondaires (1 point chacune):

- diamètre supérieur à 7 mm
- sensation différente, comme des démangeaisons, une sensibilité ou une douleur
- écoulement ou encroûtement
- inflammation ou saignement

- **Notation de Menzies** : La méthode de notation de Menzies a été proposée pour la première fois en 1996 (Menzies et al., 1996). La méthode de Menzies a été créée pour aider à différencier les mélanomes des autres lésions cutanées pigmentées. Les lésions sont évaluées en fonction de la présence de critères positifs et de l'absence de critères négatifs. Les critères négatifs incluent une lésion symétrique et la présence d'une seule couleur. Les lésions qui possèdent les deux critères négatifs sont essentiellement négatives pour le mélanome, car ces deux caractéristiques ont une sensibilité de zéro pour cent pour le mélanome.

Les critères positifs incluent des caractéristiques dermoscopiques telles qu'un voile bleu-blanc, de multiples points bruns, des pseudopodes, un flux radial, un réseau large, une dépigmentation semblable à une cicatrice, des points noirs périphériques ou des globules, de multiples points bleus/gris et cinq couleurs ou plus dans la lésion. La présence de l'un de ces critères positifs fait suspecter un mélanome.

4. Etat de l'art sur la classification des lésions cutanées

Au cours de la dernière décennie, plusieurs approches de classification automatique des lésions cutanées ont été proposées. Les approches convolutionnelles CNN ont complètement dominé le processus de classification de ces dernières. Nous présentons un état de l'art non exhaustif des approches les plus couramment utilisées.

Yu et al. ont proposé un framework en deux étapes pour la reconnaissance du mélanome en utilisant un réseau de neurones convolutifs (CNN) pour acquérir des caractéristiques discriminantes (Yu et al., 2017). Tout d'abord, un réseau résiduel entièrement convolutif intègre des représentations d'entités multi-échelles pour l'étape de segmentation, puis les derniers résultats de segmentation sont intégrés dans un réseau résiduel pour l'étape de classification.

Mahbod et al. ont proposé une étude comparative du diagnostic des lésions mélanocytaires pigmentées (Mahbod et al., 2019). Le CNN proposé combine les sorties de deux CNN entraînés sur deux ensembles de données différents, l'un avec 7895 images dermoscopiques et l'autre avec 5829 images cliniques de lésions cutanées. L'étude prouve l'efficacité du CNN pour la classification des images dermoscopiques.

Une approche hybride pour la classification des lésions cutanées est proposée (Tschandl et al., 2019) en utilisant un générateur de caractéristiques profondes à partir de trois modèles CNN pré-entraînés : AlexNet, VGG16 et ResNet-18. Une

combinaison de plusieurs classifieurs SVM est utilisée dans la dernière étape de classification.

Jadhav et al. ont extrait les caractéristiques des lésions à l'aide de CNN, puis un SVM est utilisé pour la classification (Jadhav et al., 2019),.

Trois classificateurs sont utilisés dans le travail de Patil et Dongre pour détecter les mélanomes à partir d'une collection d'images dermoscopiques PH2 (Patil&Dongre, 2020). Les classificateurs KNN, Naive Bayes et SVM sont utilisés dans l'étape de classification. Le plus grand taux de la classification est obtenue par un SVM avec augmentation des données.

Pour extraire les caractéristiques des images de cancer de la peau, les auteurs ont utilisé trois approches et classificateurs différents (Murugan et al., 2019). La forme, la règle ABCD et le GLCM sont les caractéristiques extraites. Les classificateurs utilisés sont : kNN, forêt aléatoire et SVM dans l'étape de classification.

Dans (Arora et al., 2020b), les auteurs ont extrait les caractéristiques de forme, de texture et de couleur des lésions cutanées. La classification a été faite par un SVM. Les patients atteints de mélanome ont été classés et différenciés des cas sans mélanome.

Dans (Melbin& Raj, 2021), les auteurs présentent une approche intégrée basée sur des caractéristiques ABCD modifiées, et un SVM a été utilisée pour détecter et classer les images de lésions cutanées. Les auteurs ont pris trois maladies de la peau, à savoir le mélanome, la kératose séborrhéique et le lupus érythémateux, à partir de trois ensembles de données différents.

Surowka et Ogorzalek ont extrait des caractéristiques d'ondelettes appropriées pour la classification des images dermoscopiques en deux classes : mélanome et atypique bénigne (Surowka&Ogorzalek, 2019). La classification est faite avec Naive Bayes.

Dans (Mporas et al., 2020), les auteurs présentent une architecture pour classer les lésions cutanées pigmentées à partir d'images dermatoscopiques. Les caractéristiques basées sur la couleur ont été extraites et le classificateur utilisé est AdaBoost avec forêt aléatoire.

Dans (Dhivyaa et al., 2020), les auteurs ont utilisé des arbres de décision et des algorithmes de forêts aléatoires pour la classification des lésions cutanées. Ils ont testé leur approche sur HAM10000 et ISIC 2017. Les algorithmes d'arbre de décision et les forêts aléatoires aident à générer un meilleur raisonnement sur les résultats, ce qui en fait un bon objectif pour traiter les problèmes qui affectent également d'autres régions. Les résultats obtenus sont considérés comme supérieurs à ceux qui sont similaires dans la littérature.

Perez et al. ont proposé une étude d'évaluation des facteurs qui affectent le choix de la meilleure architecture CNN pour l'analyse des lésions cutanées (Perez et al., 2019). L'évaluation des performances de modèles d'ensemble simples contrastant avec des modèles uniques, basée sur l'évaluation de 13 facteurs sur neuf architectures sur l'ensemble de données ISIC 2017. Les résultats obtenus se situaient entre 84% et 91%.

Dans (Valle et al., 2020), les auteurs ont exploré 10 choix auxquels sont confrontés les chercheurs : utilisation de l'apprentissage par transfert, architecture de modèle, apprentissage des données, résolution d'image, type d'augmentation de données, normalisation des entrées, utilisation de la segmentation, durée de l'apprentissage, utilisation supplémentaire de SVM et augmentation des données de test . Ils ont conclu que l'avancement de la recherche sur l'analyse automatisée des lésions cutanées nécessite la conservation d'ensembles de données publics plus importants. Les ensembles de modèles sont une alternative rentable aux conceptions factorielles complètes coûteuses et aux conceptions séquentielles instables.

Dans (Ozkan&Koklu, 2017), les auteurs ont classé un ensemble de données PH2 en trois groupes comme normal, anormal et mélanome à l'aide de quatre classifieurs d'apprentissage automatique différents : ANN, SVM, KNN et arbre de décision. Ils ont atteint respectivement 92,50%, 89,50%, 82,00% et 90,00% pour ANN, SVM, KNN et arbre de décision.

Ghalejoogh et al. ont proposé une méthode d'ensemble d'empilement basée sur l'algorithme de méta-apprentissage pour classer les lésions cutanées (Ghalejoogh et al., 2020a). Le modèle a utilisé quatre classifieurs : KNN, SVM, ENN et MLP. L'étape de segmentation utilisée pour différentes extractions de caractéristiques est basée sur la forme, la couleur et la texture. Ils ont atteint 98,5 % pour les classes de mélanome et de non-mélanome sur l'ensemble de données PH2.

Chakravorty et al. ont proposé une combinaison de trois vecteurs d'extraction de caractéristiques, basée sur la segmentation pour capturer la distribution asymétrique de la forme, de la couleur et de la structure (Chakravorty et al. 2016). Les caractéristiques géométriques pour capturer les différentes formes liées à la lésion, les caractéristiques de couleur utilisant Kullback - Leibler pour capturer la divergence de la distribution des couleurs et les caractéristiques structurelles pour capturer le contraste et la différence de luminance. Les résultats obtenus étaient de 83 % pour l'ensemble de données PH2.

Salido et Ruiz ont utilisé une technique de prétraitement pour supprimer les artefacts indésirables et un réseau neuronal convolutif AlexNet pour la classification (Salido& Ruiz, 2018). Ils ont testé le classifieur en utilisant des images prétraitées et

non traitées sur l'ensemble de données PH2. Ils ont obtenu une précision de 93 % pour deux classes et une précision de 67,5% pour trois classes.

AshaDeepika et Yamini ont utilisé un prétraitement suivi d'une segmentation et enfin de deux classifieurs d'apprentissage automatique SVM et KNN pour la classification (AshaDeepika&Yamini (2020)). De plus, ils ont utilisé l'amélioration d'image, CLAHE pour le domaine spatial et SWT pour le domaine fréquentiel, puis l'architecture CNN comme extracteur de caractéristiques suivi du classifieur SVM. La précision obtenue était de 92,80 % sur l'ensemble de données PH2.

Khalid et al. ont proposé une architecture AlexNet affinée en remplaçant la dernière couche par un softmax pour classer la lésion sur trois classes différentes à partir du jeu de données PH2 (Hosny et al., 2018). Le résultat obtenu était de 98,61 % sur l'ensemble de données PH2.

Ahmed et al. ont proposé une architecture affinée Xception, Inception-ResNet-V2 et NasNetLarge sur l'ensemble de données ISIC2019 (Ahmed et al., 2020). Ils ont atteint un taux de classification de 93,70%.

5. Conclusion

Dans ce chapitre, ont été présentés la composition et le fonctionnement de la peau ainsi que ses pathologies. En outre, une attention particulière a été portée sur l'une de ces pathologies : le cancer de la peau et les caractéristiques permettant d'établir son diagnostic.

Ce chapitre se termine par un état de l'art non exhaustif des approches les plus couramment utilisées en classification automatique des lésions cutanées.

Chapitre 2 : Intelligence artificielle, Apprentissage machine et Apprentissage profond

1	Introduction.....	29
2	Intelligence artificielle, Apprentissage machine et Apprentissage profond.....	29
3	Classification, régression et clustering	34
4	Types d'apprentissage	35
5	Classifieurs supervisés	37
6	Transfert Learning.....	54
7	Conclusion	56

1. Introduction

L'intelligence artificielle et le machine learning sont deux concepts visant à améliorer les compétences d'un système informatique. L'intelligence artificielle est un concept visant à donner aux machines la capacité de raisonner comme un être humain.

Le machine learning est une sous-catégorie de l'intelligence artificielle. Le principe est donc qu'une machine apprenne d'une manière automatique et regroupe en son sein plusieurs concepts, dont le deep learning. L'ordinateur va pouvoir apprendre afin d'apporter, par la suite, une réponse adaptée à une situation dite complexe.

Ce chapitre dresse une panoplie conceptuelle des différentes approches de l'intelligence artificielle, du machine learning et du deep learning. Dans cette thèse nous étudions six classifieurs pour l'apprentissage machine avec leurs variantes. Enfin, les aspects liés au principe de transfert de connaissances sont proposés dans une dernière section.

2. Intelligence artificielle, Apprentissage machine et Apprentissage profond

2.1. Intelligence artificielle

L'intelligence artificielle (IA) est un domaine de recherche visant à reproduire par des systèmes artificiels les diverses capacités cognitives de l'être humain (et plus largement les comportements "intelligents" du vivant), en particulier en ce qui concerne leur capacités à résoudre des problèmes complexes.

Marvin Minsky a défini l'intelligence artificielle comme étant une science dont le but est de faire réaliser par une machine des tâches que l'homme accomplit en utilisant son intelligence (Pickover, 2021).

Dans la pratique cela peut recouvrir actuellement des choses aussi variées qu'identifier des images de chats dans des vidéos, permettre à un robot de naviguer de façon autonome dans son environnement, ou poser un diagnostic médical à partir de diverses informations sur un patient.

Les premiers systèmes d'IA consistaient en des programmes informatiques qui s'appuyaient sur des règles (moteur de règles), capables de résoudre des problèmes assez complexes. Plutôt que d'intégrer au code chaque décision que le logiciel était censé prendre, le programme se divisait en une base de connaissances et un moteur d'inférence. Un moteur de règles a de nombreux atouts. Le processus est totalement

visible et auditable. L'algorithme peut gérer plusieurs scénarios complexes. Sous sa forme la plus évoluée, il examine un très grand nombre de branches dans un arbre décisionnel et trouve celle qui fournit les meilleurs résultats. Tout d'abord, une liste de règles est créée, souvent par un développeur interne, puis un moteur d'inférence ou un moteur de raisonnement sémantique effectue un cycle de correspondance, de résolution et d'action, en mesurant les informations à prendre en compte en fonction de ces règles. Les connaissances humaines sont encodées sous forme d'arbre de décision à partir d'énoncés composés de « si » et « ensuite » pour une règle spécifique.

Les développeurs enrichissaient alors la base avec des informations factuelles, et le moteur interrogeait ces dernières pour parvenir à un résultat. Mais ce type d'IA s'avérait limitée, notamment parce que son apprentissage reposait fortement sur la contribution humaine. Lorsqu'il s'agit d'apprendre et d'évoluer, ces systèmes manquent de flexibilité, et ils ne sont aujourd'hui plus guère considérés comme « intelligents ». En plus, dans de nombreux cas, il est impossible de s'appuyer sur un moteur de règles. C'est le cas, par exemple, de la reconnaissance faciale.

2.2. Apprentissage Machine

L'apprentissage est défini comme étant un changement durable dans les comportements qui résultent de l'expérience. La capacité d'apprendre donne à chaque organisme vivant la capacité de s'adapter à un environnement changeant. Relier cette définition à la machine donne lieu à l'apprentissage machine (ou apprentissage automatique, Machine Learning).

L'apprentissage machine est un sous-domaine de l'IA qui s'intéresse en particulier aux capacités d'apprentissage. Le principe est de reproduire un comportement non pas en le programmant « à la main » dans un ordinateur, mais en concevant un système plus général capable d'apprendre à partir d'exemples à résoudre un problème (Alpaydin, 2014).

L'apprentissage machine est capable de reproduire un comportement grâce à des algorithmes, eux-mêmes alimentés par un grand nombre de données. Confronté à de nombreuses situations, l'algorithme apprend quelle est la décision à adopter et crée un modèle. La machine peut automatiser les tâches en fonction des situations.

A la différence des premiers systèmes basés sur les règles, l'apprentissage machine adopte donc une approche radicalement différente : la machine apprend toute seule en ingérant d'énormes quantités de données et en détectant des schémas récurrents. Pour fonctionner, nombre d'algorithmes sont capables d'apprendre à partir de données historiques, ce qui les rend pertinents pour une plus vaste gamme d'applications (Fleury, 2021).

2.3. Apprentissage profond

L'apprentissage profond est une sous branche de l'apprentissage machine (Goodfellow et al., 2016). L'architecture se base sur un réseau de neurones à plusieurs couches cachées donnant alors une notion de profondeur dans le but d'imiter le fonctionnement du cerveau humain. Le concept d'apprentissage profond consiste à apprendre à travers plusieurs niveaux de représentation et d'abstraction qui aident à donner un sens aux données. En effet, les algorithmes d'apprentissage profond cherchent à exploiter la structure inconnue dans la distribution en entrée afin de découvrir de bonnes représentations, souvent à plusieurs niveaux, avec des fonctionnalités de niveau supérieur apprises définies en termes de fonctionnalités de niveau inférieur (Bengio, 2018).

Avantage et limites de l'apprentissage profond

La question qui se pose : pourquoi un grand nombre de filières technologiques adoptent régulièrement l'apprentissage profond comme solution ? Pour comprendre la raison, nous devons examiner les avantages qui peuvent être obtenus en utilisant une approche d'apprentissage profond. Voici quelques avantages clés de l'utilisation de cette technologie.

- **Utilisation maximale des données non structurées**

Pour la majorité des algorithmes d'apprentissage automatique, il est difficile de traiter des données non structurées, ce qui signifie qu'elles restent inutilisées et c'est exactement là que l'apprentissage profond devient utile. Nous pouvons utiliser différents types de données pour apprendre des algorithmes d'apprentissage profond tout en obtenant des informations pertinentes par rapport à l'objectif de l'apprentissage.

- **Capacité à traiter un grand nombre de caractéristiques**
La capacité à traiter un grand nombre de caractéristiques rend l'apprentissage profond très puissant lorsqu'il s'agit de données non structurées. Par contre, ils peuvent échouer pour les problèmes moins complexes car ils nécessitent l'accès à une grande masse de données pour être efficaces. Par exemple, ImageNet, la référence commune pour l'apprentissage contient plus de 14 millions d'images.
- **Élimination du besoin d'ingénierie des caractéristiques**
Dans l'apprentissage automatique, l'ingénierie des caractéristiques est un travail fondamental car elle améliore la précision et parfois le processus peut nécessiter une connaissance du domaine sur un certain problème. L'un des plus grands avantages de l'utilisation de l'approche d'apprentissage profond est sa capacité à extraire les caractéristiques par elle-même. Dans cette approche, un algorithme analyse les données pour identifier les caractéristiques qui sont en corrélation, puis les combine pour favoriser un apprentissage plus rapide sans qu'on lui dise explicitement de le faire.
- **Capacité à fournir des résultats de haute qualité**
Une fois l'apprentissage achevé correctement, un modèle d'apprentissage profond devient capable d'effectuer des milliers de tâches routinières et répétitives dans un laps de temps relativement court par rapport à ce qu'il faudrait à un être humain. De plus, la qualité du travail ne se dégrade jamais, à moins que les données d'apprentissage ne contiennent des données brutes qui ne représentent pas le problème à résoudre.

L'une des limites de l'apprentissage profond se situe au niveau des données fournies pour son apprentissage (des milliers d'images et de vidéos). Plus le système reçoit de données, plus sa précision augmente. Malheureusement la quantité ne suffit pas, la qualité et la diversité sont également des paramètres importants (Bengio, 2018).

L'apprentissage profond est extrêmement coûteux en apprentissage en raison de modèles de données complexes. De plus, l'apprentissage en profondeur nécessite des GPU performants. Cela augmente les coûts pour les utilisateurs.

Il n'y a pas de théorie standard pour la sélection des bons outils d'apprentissage profond car cela nécessite une connaissance de la topologie, de la méthode

d'apprentissage et d'autres paramètres. En conséquence, il est difficile d'être adopté par des personnes moins qualifiées.

2.4. Différence entre apprentissage machine traditionnel et apprentissage profond

L'apprentissage machine (ML) et l'apprentissage profond (DL) sont les deux principales techniques d'intelligence artificielle utilisées à l'heure actuelle. La distinction entre IA, ML et DL peut prêter à confusion. L'apprentissage machine est une catégorie d'IA, et l'apprentissage profond est une technique de l'apprentissage machine (Figure 2.1).

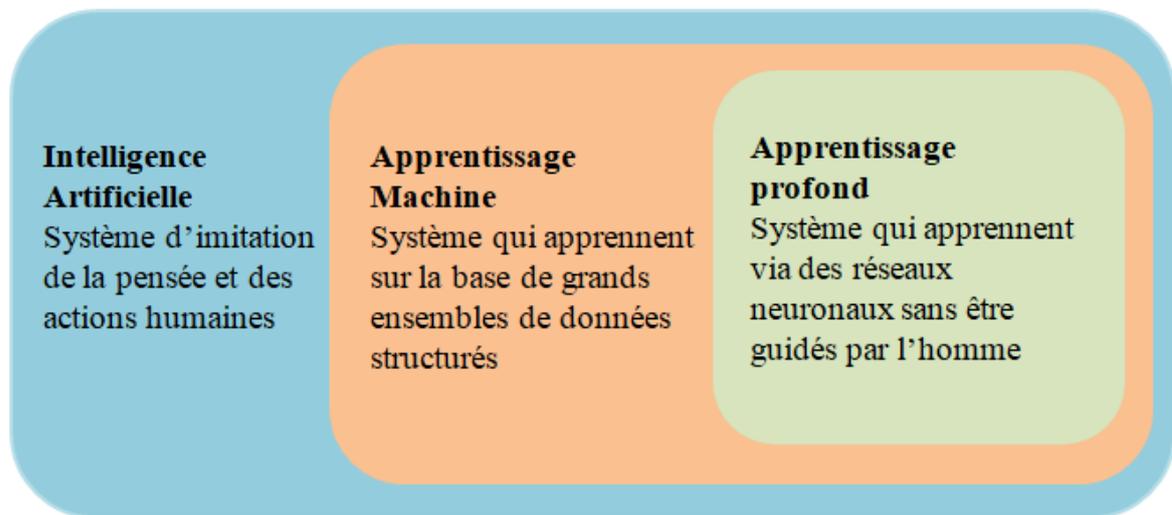


Figure 2.1 : Distinction entre IA, ML et DL.

La plupart des algorithmes d'apprentissage machine traditionnels fonctionnent bien sur des dataset contenant jusqu'à quelques centaines de caractéristiques (features). Cependant, un ensemble de données non structuré, comme celui d'une image par exemple, a un grand nombre de caractéristiques. Une seule image de 800×1000 pixels en couleur RVB comporte 2,4 millions de caractéristiques. Le processus devient lourd ou complètement irréalisable par les algorithmes d'apprentissage machine traditionnels (Fleury, 2021).

Les algorithmes d'apprentissage profond apprennent progressivement plus sur l'image au fur et à mesure qu'elle traverse chaque couche du réseau de neurones. Les premières couches apprennent à détecter les caractéristiques de bas niveau comme les bords, et les couches suivantes combinent les caractéristiques des couches précédentes dans une représentation plus globale. Par exemple, une couche intermédiaire peut identifier les bords pour détecter des parties d'un objet sur la

photo, comme une jambe ou une branche, tandis qu'une couche profonde détectera l'objet complet, comme un chien ou un arbre.

L'apprentissage machine traditionnel fait référence au processus d'extraction de caractéristiques à partir d'un grand ensemble de données chargé dans la machine (Figure 2.2). Les concepteurs formulent les règles et corrigent les erreurs commises par la machine. Dans l'apprentissage machine traditionnel, une machine reçoit des données d'apprentissage et des exemples pour l'aider à prendre les bonnes décisions (Kaware, 2020). En d'autres termes. Examinons les différences entre l'apprentissage automatique traditionnel et l'apprentissage profond.

- Les modèles d'apprentissage profond sont capables de créer de nouvelles caractéristiques par eux-mêmes, tandis que dans l'approche d'apprentissage automatique traditionnelle, les fonctionnalités doivent être identifiées avec précision par les utilisateurs.
- Dans l'apprentissage profond, les problèmes sont résolus de bout en bout, tandis que dans l'apprentissage automatique, les tâches sont divisées en petits morceaux, puis les résultats reçus sont combinés en une seule conclusion.

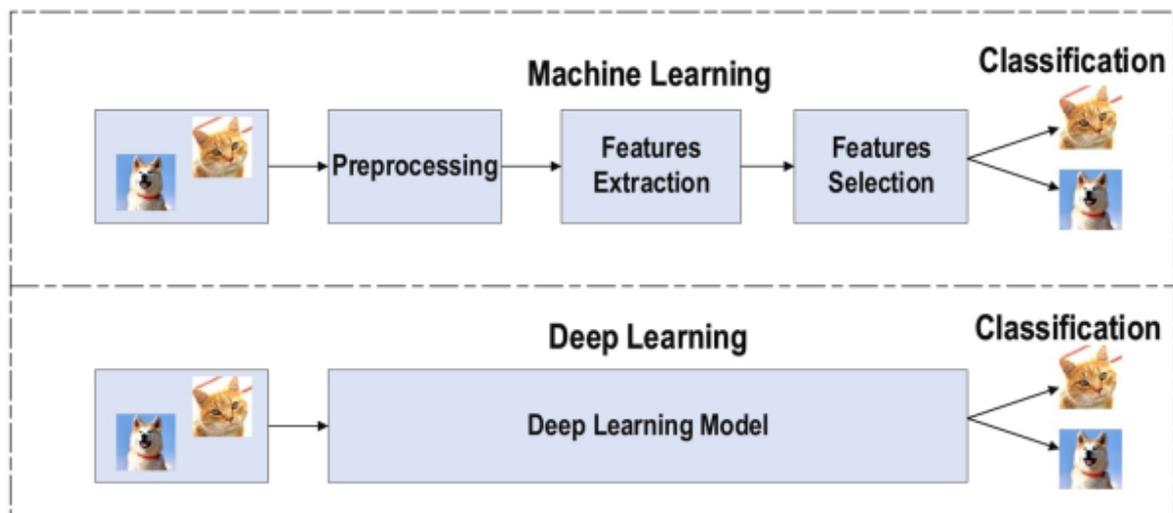


Figure 2.2 : Différence entre apprentissage machine et apprentissage profond.

3. Classification, régression et clustering

3.1. Régression

La régression vise à décrire ou à prévoir une valeur numérique particulière à l'aide de données précédemment collectées, et le programme d'apprentissage automatique doit estimer et comprendre les relations entre les variables (Fleury, 2021). L'analyse de régression est particulièrement utile pour la modélisation et la prévision car elle se concentre sur une variable dépendante et plusieurs autres variables évolutives.

3.2. Classification

Dans l'apprentissage automatique supervisé, les algorithmes de classification peuvent décrire ou prévoir une valeur de classe. Le problème de classification implique de prendre des vecteurs d'entrée et de déterminer à quelle classe ils appartiennent, en utilisant des exemplaires de chaque classe comme apprentissage variables (Fleury, 2021). La caractéristique la plus importante du problème de classification est qu'il est discret, ce qui signifie que chaque exemple appartient à une seule classe et que l'ensemble des classes englobe tout l'espace de sortie.

3.3. Clustering

Les approches d'apprentissage non supervisé sont des algorithmes de clustering. Kmeans, mean-shift et expectation-maximization sont trois algorithmes de clustering populaires. Le clustering est le processus de regroupement d'une collection d'éléments afin que les objets liés soient regroupés et que les objets dissemblables soient divisés en classes distinctes variables (Fleury, 2021). Il peut être utilisé pour diviser les données en plusieurs classes et effectuer une analyse de modèle sur chaque ensemble de données. Les stratégies de clustering sont particulièrement utiles dans les applications métier où de grandes quantités de données doivent être segmentées ou catégorisées.

4. Types d'apprentissage

Les types d'apprentissage automatique varient en fonction de l'utilisation prévue et des résultats cibles. Comme le montre la Figure 2.3, le ML comprend trois catégories principales : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

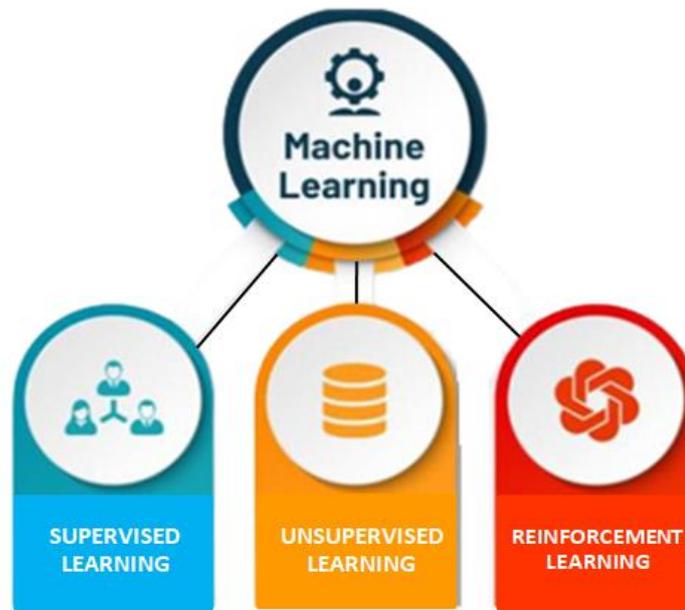


Figure 2.3 : Types d'apprentissage machine.

4.1. Apprentissage supervisé

Un apprentissage supervisé apprend un modèle avec un ensemble de données qui comprend des variables d'entrée (X) et de sortie étiquetées (Y). Selon le modèle entraîné en apprentissage supervisé, les données de sortie sont estimées en fonction des nouvelles données d'entrée (Figure 2.4). L'apprentissage supervisé est également utilisé en classification et en régression.

Les algorithmes d'apprentissage supervisé les plus utilisés sont : la régression linéaire, régression logistique, forêt aléatoire, Machines à vecteurs de support (SVM), réseaux de neurones, arbres de décision, Bayes naïf, le plus proche voisin.

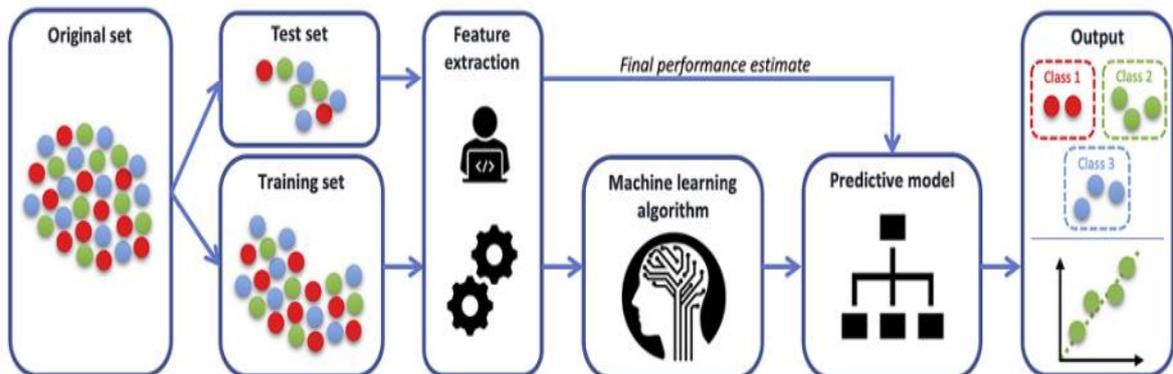


Figure 2.4 : Pipeline de l'apprentissage supervisé.

4.2. Apprentissage non supervisé

Les algorithmes d'apprentissage non supervisé les plus utilisés sont : le clustering K-means, PCA (Analyse en Composantes Principales), règle d'association.

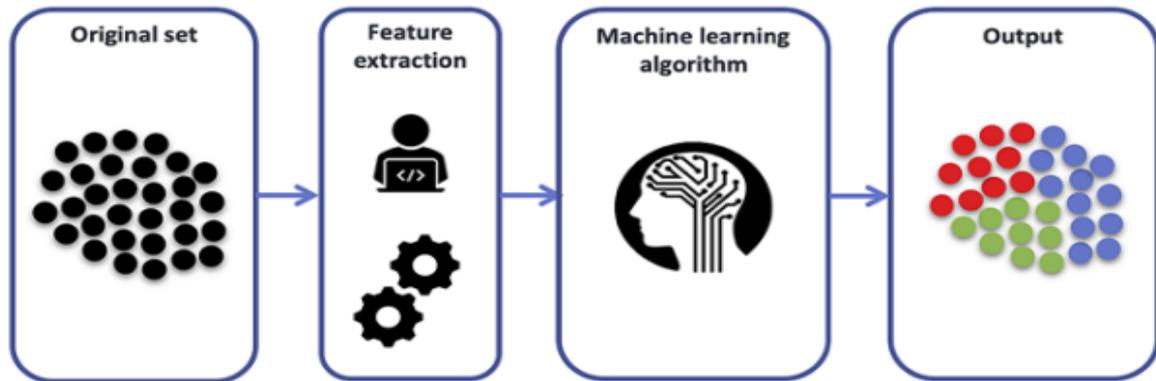


Figure 2.5 : Pipeline de l'apprentissage non supervisé.

4.3. Apprentissage par renforcement

L'apprentissage par renforcement est une approche d'apprentissage automatique inspirée du comportementalisme qui traite des actions que les sujets doivent entreprendre afin d'obtenir le plus grand nombre de récompenses dans un environnement. En apprentissage automatique, l'environnement est souvent modélisé comme un processus de décision de Markov (MDP), dans ce contexte, de nombreux algorithmes d'apprentissage par renforcement utilisent des techniques de programmation dynamique.

4.4. Apprentissage semi-supervisé

Les approches semi-supervisées sont des méthodes introduites en raison des limites des méthodes supervisées et non supervisées. Le processus coûteux d'apprentissage supervisé sur le gros volume de données et le spectre d'application limité des méthodes d'apprentissage automatique non supervisé. Les méthodes semi-supervisées nécessitent une plus petite quantité de données étiquetées, et les autres ne sont pas étiquetées. Donc, généralement, nous utilisons le clustering (méthode non supervisée) pour regrouper une plus grande partie des données, et le reste, nous l'étiquetons avec une méthode supervisée.

5. Classifieurs supervisés

Le processus de classification a pour but de regrouper les éléments possédant des caractéristiques communes en sous-groupes. Dans la littérature scientifique nous retrouvons plusieurs classifieurs. Dans cette thèse nous nous attarderons à étudier six d'entre eux avec leurs variantes qui seront utilisées dans la partie expérimentale, comme mentionné dans la figure suivante (Kaware, 2020 ;Fleury, 2021) :

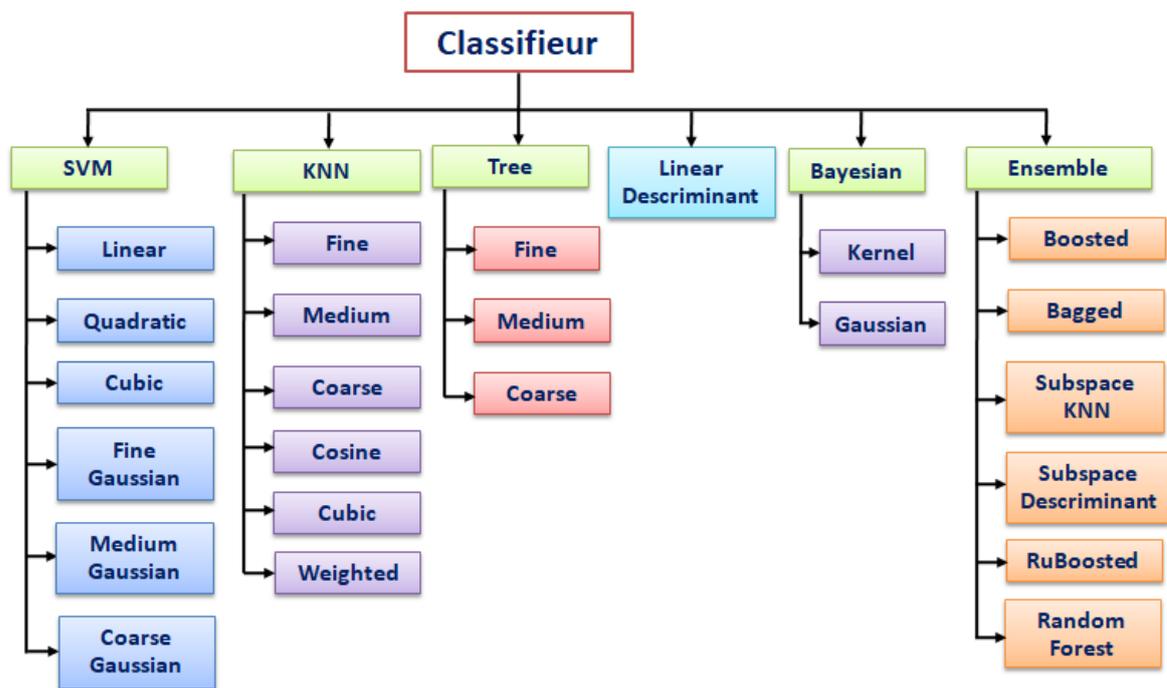


Figure 2.6 : Classification des algorithmes supervisés utilisés.

5.1. Support Vector Machine (SVM)

Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais Support Vector Machine, SVM) est un algorithme puissant introduit à l'origine par Vapnik (Vapnik et al., 1998). Il possède des performances remarquablement robustes vis-à-vis des données clairsemées et bruitées, ce qui le rend utile dans de nombreuses applications aussi bien dans les cas de classification que de régression. Les SVM ont rapidement été adoptés pour leur capacité à travailler avec des données de grandes dimensions, le faible nombre d'hyperparamètres, le fait qu'ils soient bien fondés théoriquement, et leurs bons résultats en pratique. Les SVM reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau.

Fondamentalement, le SVM est un classifieur linéaire à deux classes. Il classe les données en construisant un hyperplan qui détermine une ligne droite séparant l'espace en deux classes aussi homogènes que possible, de telle façon que la distance entre les différentes classes et la frontière qui les sépare soit maximale. Cette distance est aussi appelée « marge » et les SVMs sont ainsi qualifiés de « séparateurs à vaste marge ». L'ensemble des points de données les plus proches de cette frontière est appelé les vecteurs de support.

La Figure suivante illustre le principe du SVM. On peut y observer un hyperplan séparant deux classes. Chaque classe se trouve à une distance (D) de cette frontière. La marge que l'on cherche à maximiser étant alors la somme de ces deux distances.

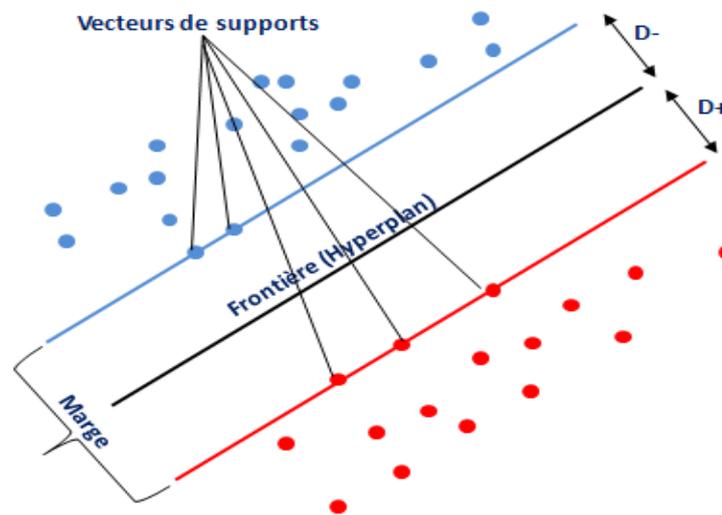


Figure 2.7 : Illustration du principe du SVM.

Ce type d'algorithme est particulièrement efficace lorsque les données sont linéairement séparables, un SVM linéaire est utilisé, ce qui en pratique est rarement le cas. Pour y pallier lorsque les données sont non linéairement séparables, la technique consiste donc à projeter les données dans un espace de dimension supérieure à l'aide d'un élément appelé noyau, permettant alors une création de frontière. L'objectif c'est de forcer les points de données à devenir linéairement séparables si possible.

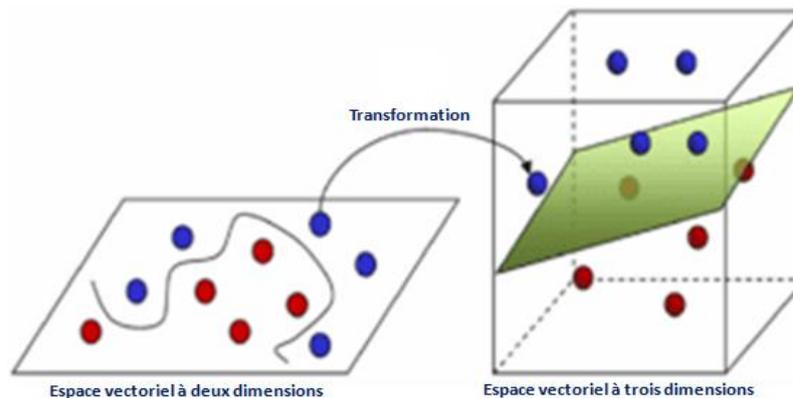


Figure 2.8 : Projection des données dans un nouvel espace.

Le noyau non linéaire se compose de deux fonctions polynomiales (quadratique et cubique) et de la fonction de base radiale (fine, medium, et coarsegaussian).

Le tableau suivant résume les différents noyaux des SVM:

Modèle du noyau	Fonction du noyau
Linear SVM	Linear
Quadratic SVM	Quadratic
Cubic SVM	Cubic
Fine Gaussian SVM	Gaussian
Medium Gaussian SVM	Gaussian
CoarseGaussian SVM	Gaussian

Tableau 2.1 : Différents noyaux des SVM.

5.2. K-NearestNeighbor

K-NN (k plus proches voisins ou knearest Neighbors) fut inventé par Weiss et Kulikowski dans les années 1990 (Kaware, 2020). Il s'agit d'un algorithme de classification supervisé simple et facile à mettre en œuvre qui peut être utilisé pour résoudre les problèmes de classification et de régression.

L'algorithme k-NN est basé sur la distance qui détermine la réponse prédite d'un point donné en vérifiant la classe majeure des k points les plus proches (K étant le

nombre de voisins à considérer). Il existe de nombreuses façons de mesurer la similarité entre les attributs dans l'ensemble de test et l'ensemble d'apprentissage. Les plus significatives sont la distance euclidienne et de Manhattan.

Le nombre de voisins K peut être modifié pour ajuster l'algorithme k -NN. Il est courant de choisir une valeur de K entre 3 et 10, pour surmonter les problèmes de sur-ajustement et de sous-ajustement (Fleury, 2021).

La figure suivante illustre ce principe. Pour une observation nous devons définir son groupe d'appartenance. Si nous choisissons pour paramètre $K=5$, cela signifie que le groupe choisi sera celui des cinq plus proches voisins ayant le même groupe d'appartenance.

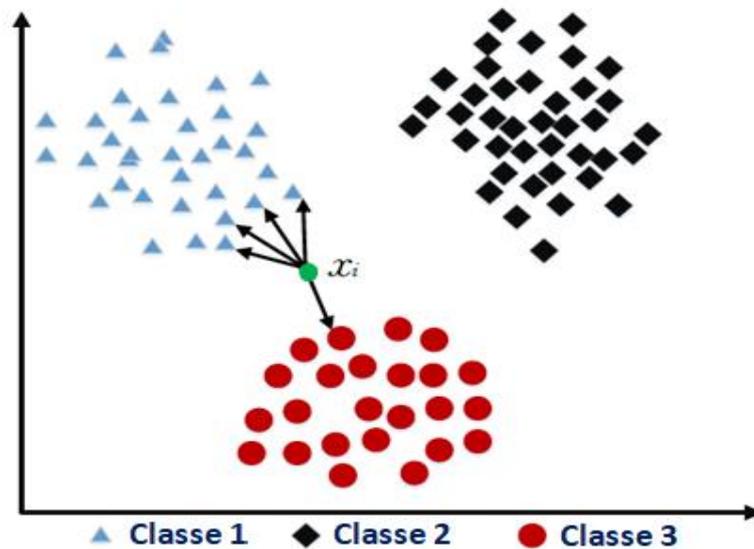


Figure 2.9 : K-Nearest Neighbor avec $K=5$.

Chaque système K-NearestNeighbor nécessite un ensemble de paramètres :

- Une fonction de distance qui mesure la similarité entre les problèmes ou les entrées de données. Ceci est nécessaire pour mesurer quels sont les voisins les plus proches du nouveau problème.
- Un nombre de voisins qui sont pris en compte lors de la résolution du nouveau problème
- Une fonction de pondération qui permet une quantification plus poussée des voisins trouvés pour augmenter la prédiction et la qualité de l'apprentissage
- Une méthode d'évaluation qui décrit une fonction sur la façon d'utiliser les voisins trouvés pour résoudre le problème donné.

Les points faibles de cet algorithme sont : d'une part, son coût en puissance de calcul (pour prédire l'image d'un nouveau point, on doit calculer sa distance à tous les autres), d'autre part le fait de devoir conserver toutes les données d'entraînement en mémoire (k-NN convient donc plutôt aux problèmes d'assez petite taille). Il est également important de noter que cet algorithme est vulnérable à la malédiction de la dimensionnalité : le nombre de données nécessaires pour avoir un bon estimateur croît de manière exponentielle avec la dimension, autrement dit avec la complexité de la représentation des données. Pour ces raisons, k-NN est assez peu utilisé dans sa forme première mais plutôt avec des versions améliorées qui limitent partiellement ces défauts (Wu et al., 2002). Il existe des variantes différentes de l'algorithme k-NN qui peuvent être utilisées en se basant sur diverses distances. Nous pouvons citer :

- Fine k-NN: fait des distinctions finement détaillées entre les classes avec le nombre de voisins défini à un ($K=1$).
- Medium k-NN: moins de distinctions qu'un k-NN fin avec le nombre de voisins fixé à 10 ($K=10$).
- Coarse k-NN: un nombre de voisins défini à 100 ($K=100$).
- Cosine k-NN: utilise la distance cosinus. La distance cosinus entre deux vecteurs u et v est définie comme suit :

$$1 - \frac{u \cdot v}{|u| \cdot |v|} \quad (2.1)$$

- Cubic k-NN: utilise la distance cubique. La distance cubique entre deux vecteurs u et v à n dimensions est définie comme suit :

$$\sqrt[3]{\sum_{i=1}^n |u_i - v_i|^3} \quad (2.2)$$

- Weighted k-NN: utilise la pondération de distance en attribuant aux k plus proches voisins un poids de $1/k$ et tous les autres un poids de 0. La distance euclidienne pondérée entre deux vecteurs u et v à n dimensions est définie comme suit :

$$\sqrt{\sum_{i=1}^n w_i (x_i - y_i)^2} \quad (2.3)$$

avec $0 < w_i < 1$ et $\sum_{i=1}^n w_i = 1$

5.3. Arbres de décision

Les arbres de décision sont des approches populaires, largement utilisées et efficaces en classification et en régression (Kaware, 2020). Elles sont également connues sous le nom de méthode d'apprentissage supervisé non paramétrique.

L'objectif des arbres de décision est de créer un modèle qui prédit la classe d'un attribut en apprenant des règles de décision simples déduites de l'ensemble des données d'entrée.

Il s'agit d'un classifieur arborescent qui commence par un nœud racine et se termine par une décision prise par les feuilles (Figure 2.10). La structure d'un arbre de décision couvre deux types de nœuds : le nœud feuille et le nœud interne. Une feuille contient une étiquette de classe et chaque nœud interne est une question sur les caractéristiques. Il se subdivise en fonction des réponses soit en :

- Coarse : moins de feuilles avec un nombre maximum de divisions fixé à 4
- Medium : le nombre maximum de divisions est fixé à 20
- Fine : le nombre maximum de divisions est fixé à 100

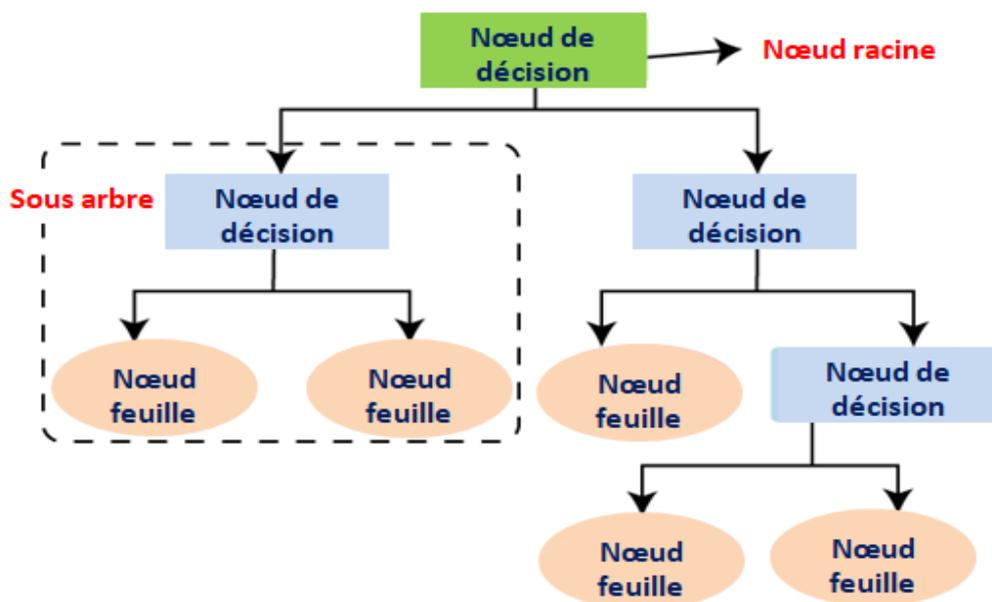


Figure 2.10 : Structure d'un arbre de décision.

Pour définir la décision finale. Chaque nœud comporte une condition, et les branchements sont en fonction de cette condition (Vrai ou Faux). Plus on descend dans l'arbre, plus on cumule les conditions. Chaque individu (ou observation), qui doit être attribué(e) à une classe, est décrit(e) par un ensemble de variables qui sont testées dans les nœuds de l'arbre. Les tests s'effectuent dans les nœuds internes et les décisions sont prises dans les nœuds feuilles.

Avantages

- Simple à comprendre car ils imitent généralement la capacité de réflexion humaine lors de la prise de décision.
- Utile pour résoudre des problèmes liés à la prise de décision.
- Toutes les issues possibles d'un problème sont explorées.
- Moins prétraitement des données par rapport à d'autres algorithmes.

Limites

- Contient de nombreuses couches, ce qui le rend complexe.
- Peut y avoir un problème de sur-ajustement, qui peut être résolu à l'aide de l'algorithme Random Forest.
- Complexité de calcul de l'arbre de décision qui peut augmenter pour plus d'étiquettes de classe.

5.4. Analyse Discriminante Linéaire

L'analyse discriminante linéaire (ADL) est un modèle linéaire de classification et de réduction de la dimensionnalité. En 1936, Fisher a formulé un discriminant linéaire pour deux classes (Fisher, 1936), et plus tard, en 1948, Rao l'a généralisé pour plusieurs classes (Rao, 1948).

L'analyse discriminante linéaire fait partie des techniques d'analyse discriminante prédictive. Il s'agit d'expliquer et de prédire l'appartenance d'un individu à une classe (groupe) prédéfinie à partir de ses caractéristiques mesurées à l'aide de variables prédictives. Elle consiste à regrouper les données où une donnée sera considérée proche d'une autre si elle provient de la même densité de probabilité plutôt qu'en effectuant un simple calcul de distance déterministe. Chaque classe de données pourra donc être résumée par les paramètres de sa densité de probabilité. Chaque classe est supposée être Gaussienne.

L'apprentissage du modèle ADL consiste en l'estimation des paramètres μ_k (les moyennes) et Σ (matrice de covariance) à partir des données d'apprentissage. Ces estimations sont données par :

$$\pi_k = \frac{\sum_{i|z_i=k} 1}{n} = \frac{n_k}{n} \quad (2.4)$$

Où n_k est le cardinal de la classe k , z_i est l'étiquette de classe de l'exemple x_i et $z_i = k$ désigne le fait que l'exemple x_i appartient à la classe k .

L'estimation des paramètres μ_k et Σ à partir des données d'apprentissage sont données par :

$$\mu_k = \frac{1}{n_k} \sum_{i|z_i=k} x_i \quad (2.5)$$

$$\Sigma = \frac{1}{n - K} \sum_{k=1}^K \sum_{i|z_i=k} (x_i - \mu_k)(x_i - \mu_k)^T \quad (2.6)$$

Une fois que les paramètres sont estimés, on peut prédire les classes des données de test par la règle du MAP (maximum a posteriori). Cette règle consiste à maximiser les probabilités a posteriori, i.e. affecter chaque donnée de test x_i à la classe \hat{z}_i ayant la plus grande probabilité a posteriori :

$$\tau_{ik} = \mathbb{P}(z_i = k | x_i; \mu_k, \Sigma) = \frac{\pi_k f_{ik}(x_i; \mu_k, \Sigma)}{\sum_{l=1}^K \pi_l f_{il}(x_i; \mu_l, \Sigma)} \quad (2.7)$$

Avantages

- La régression logistique est l'un des modèles de classification linéaire les plus populaires qui fonctionne bien pour la classification binaire mais qui échoue dans le cas de problèmes de classification multiples avec des classes bien séparées. Alors que ADL les gère assez efficacement.
- ADL peut également être utilisé dans le prétraitement des données pour réduire le nombre de caractéristiques, ce qui réduit considérablement le coût de calcul.

Limites

- Les limites de décision linéaires peuvent ne pas séparer efficacement les classes non linéairement séparables.
- Dans les cas où le nombre d'observations dépasse le nombre d'entités, ADL peut échouer. C'est ce qu'on appelle le problème de petite taille d'échantillon.

5.5. Naive Bayesian

Le classifieur Naive Bayesian est une approche de classification probabiliste simple mais robuste qui est basée sur le théorème de Bayes qui décrit la probabilité d'un événement à partir d'une connaissance préalable de la condition potentiellement liée à l'événement (Murty, 2011). Le théorème de Bayes s'exprime sous la formule suivante :

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad (2.8)$$

Un classifieur Naive Bayesian suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques. Il calcule toutes les probabilités de toutes les classes pour une caractéristique cible et sélectionne celle dont la probabilité est la plus élevée. Selon la nature de chaque modèle probabiliste, les classifieurs bayésiens naïfs peuvent être entraînés efficacement dans un contexte d'apprentissage supervisé.

L'avantage du classifieur Naïve Bayes est qu'il ne nécessite qu'une petite quantité de données d'apprentissage pour estimer les moyennes et les variances des variables nécessaires à la classification. Étant donné que des variables indépendantes sont supposées, seules les variances des variables pour chaque classe doivent être déterminées et non la matrice de covariance entière.

Gaussian Naive Bayes est une variante de Naive Bayes qui suppose que les valeurs associées avec chaque classe de chaque caractéristique suivent une distribution gaussienne. La vraisemblance de la caractéristique est supposée être gaussienne, par conséquent, la probabilité conditionnelle est donnée par :

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (2.9)$$

La Figure suivante indique le fonctionnement d'un classifieur GaussianNaive Bayes. À chaque point de données x , la distance z-score entre ce point et chaque moyenne de classe est calculée, à savoir la distance par rapport à la moyenne de classe divisée par l'écart type de cette classe. Ainsi, le GaussianNaive Bayes a une approche légèrement différente et peut être utilisé efficacement.

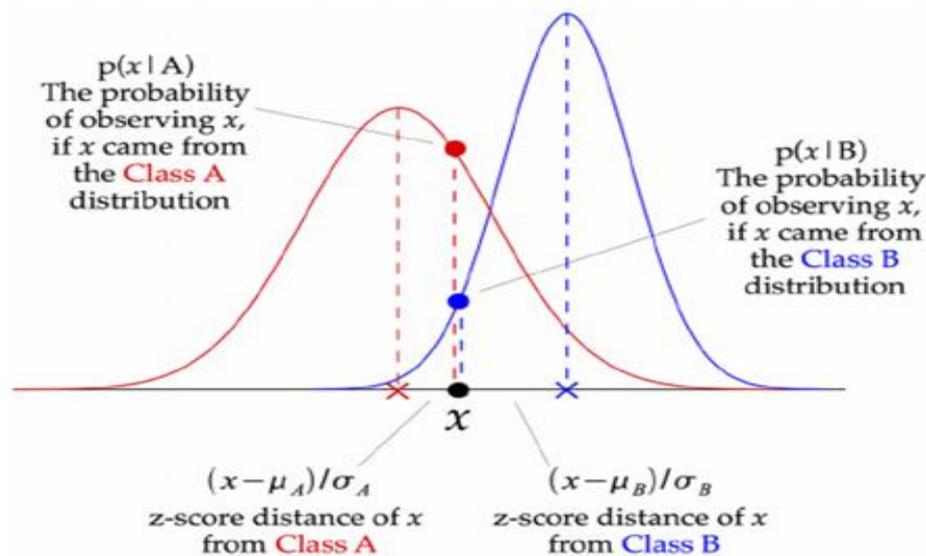


Figure 2.11 : Distribution gaussienne pour deux classes.

Contrairement à l'opérateur Naive Bayes, l'opérateur KernelNaive Bayes peut être appliqué sur des attributs numériques. Un noyau est une fonction de pondération utilisée dans les techniques d'estimation non paramétriques. Les noyaux sont utilisés dans l'estimation de la densité du noyau pour estimer les fonctions de densité des variables aléatoires, ou dans la régression du noyau pour estimer l'espérance conditionnelle d'une variable aléatoire. L'avantage de l'approche noyau est qu'aucune hypothèse n'a besoin d'être faite sur la distribution d'une variable, toute distribution peut être approchée et qu'elle est connue pour avoir des propriétés asymptotiques souhaitables.

KernelNaive Bayes nécessite plus de temps de calcul et plus de mémoire que la distribution normale. Pour chaque prédicteur que vous modélisez avec une distribution de noyau, le classifieur naïf de Bayes calcule une estimation de densité de noyau distincte pour chaque classe en fonction des données d'apprentissage de cette classe. Par défaut, le noyau est le noyau normal et le classifieur sélectionne automatiquement une largeur pour chaque classe et prédicteur.

5.6. Apprentissage d'ensemble

L'apprentissage d'ensemble est un paradigme d'apprentissage automatique dans lequel plusieurs apprenants sont formés et combinés pour résoudre le même problème (Figure 2.12). Lorsque plusieurs apprenants sont utilisés, la capacité de généralisation intégrée peut être beaucoup plus forte que celle d'un seul apprenant. Contrairement aux approches d'apprentissage automatique ordinaires qui tentent d'apprendre une hypothèse à partir de données d'apprentissage, les méthodes d'ensemble tentent de construire un ensemble d'hypothèses et de les combiner pour les utiliser.

Techniquement, l'apprentissage d'ensemble repose sur la mise en commun de multiples algorithmes d'apprentissage pour obtenir de meilleures performances que celles que l'on pourrait obtenir de n'importe lequel des algorithmes d'apprentissage constituants pris individuellement.

L'ensemble de méthodes bagging, boosting et les forêts aléatoires est un exemple d'apprentissage ensembliste.

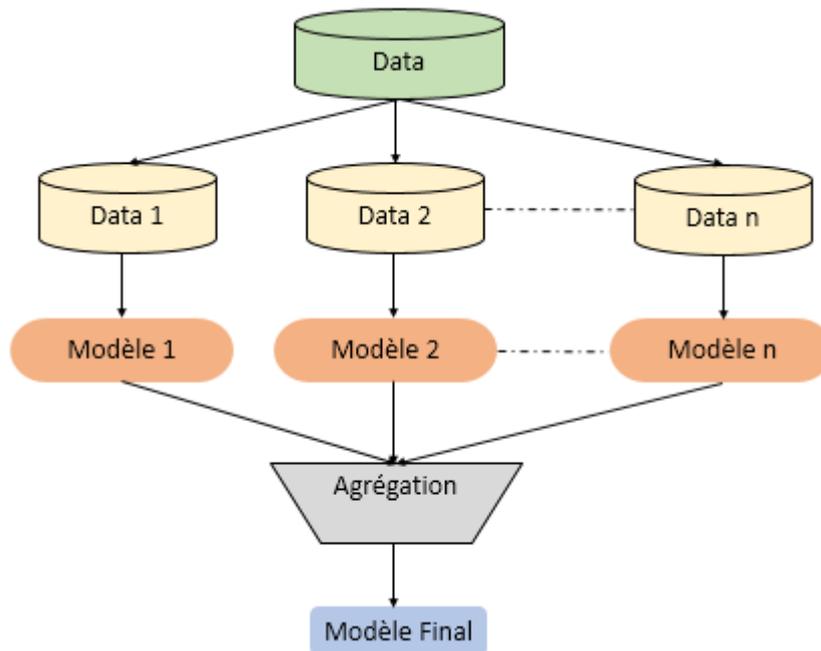


Figure 2.12. Schéma général d'apprentissage d'ensemble.

5.6.1. Pourquoi un apprentissage d'ensemble ?

Dietterich énumère trois principales raisons d'utiliser un système basé sur un ensemble (Dietterich, 2000) :

- Statistique : est liée au manque de données adéquates pour représenter correctement la distribution des données,
- Informatique : est le problème de sélection de modèle, où parmi de nombreux modèles qui peuvent résoudre un problème donné, lequel nous devrions choisir,
- Représentatif : est de s'adresser aux cas où le modèle choisi ne peut pas représenter correctement la limite de décision recherchée.

Sélection du modèle

C'est peut-être la principale raison pour laquelle les systèmes basés sur l'apprentissage d'ensembles sont utilisés : quel est le classifieur le plus approprié pour un problème de classification donné ? Cette question peut être interprétée de deux manières différentes :

- i) quel type de classifieur choisir parmi de nombreux modèles concurrents,
- ii) pour un algorithme de classification particulier, quelle paramétrage et initialisation de cet algorithme doit être choisie.

La procédure la plus couramment utilisée, à savoir choisir les classifieurs avec la plus petite erreur sur les données d'apprentissage, est malheureusement erronée. Les performances sur un ensemble de données d'apprentissage peuvent être trompeuses en termes de performances de classification sur des données test.

Ensuite, parmi tous les classifieurs qui peuvent tous avoir le même entraînement lequel devrait être choisi ? On peut être tenté de choisir au hasard, mais cette décision s'accompagne du risque de choisir un modèle particulièrement médiocre.

L'utilisation d'un ensemble de modèles avec la combinaison de leurs sorties au lieu d'en choisir un seul peut réduire le risque d'une mauvaise sélection d'un classifieur particulièrement peu performant.

Il est important de souligner qu'il n'y a aucune garantie que la combinaison de plusieurs classifieurs fonctionnera toujours mieux que le meilleur classifieur

individuel de l'ensemble. De même, une amélioration de la performance moyenne de l'ensemble ne peut être garantie (Fumera & Roli, 2005), mais cela réduit certainement le risque global de faire une sélection particulièrement mauvaise.

Trop ou peu de données

Les systèmes basés sur des ensembles peuvent être utiles lorsqu'il s'agit de gros volumes de données ou d'un manque de données adéquates. Lorsque la quantité de données d'apprentissage est trop importante, elle peut rendre difficile l'apprentissage d'un seul classifieur. Les données peuvent être stratégiquement partitionnées en sous-ensembles plus petits. Chaque partition peut ensuite être utilisée pour apprendre un classifieur séparé qui peut ensuite être combiné à l'aide d'une règle de combinaison appropriée. Si, d'autre part, il y a trop peu de données, alors le bootstrap peut être utilisé pour entraîner différents classifieurs en utilisant différents échantillons de données, où chaque échantillon bootstrap est un échantillon aléatoire de données avec augmentation et prétraitement (Efron, 1979).

Diviser et conquérir

Certains problèmes sont tout simplement trop difficiles à résoudre pour un classifieur donné. En fait, la frontière de décision qui sépare les données de différentes classes peut être trop complexe. Un classifieur linéaire, capable d'apprendre des frontières linéaires, ne peut pas déterminer une frontière non linéaire complexe. Cependant, une combinaison appropriée d'un ensemble de classifieurs linéaires peut déterminer n'importe quelle frontière non linéaire.

Dans un sens, le système de classification suit une approche (diviser pour mieux régner) en divisant l'espace de données en partitions plus petites et plus faciles à apprendre, où chaque classifieur n'apprend qu'une des partitions les plus simples. La limite de décision complexe sous-jacente peut alors être approximée par une combinaison appropriée de différents classifieurs.

Fusion des données

Dans de nombreuses applications qui nécessitent une prise de décision automatisée, il n'est pas rare de recevoir des données obtenues de différentes sources qui peuvent

fournir des informations complémentaires et peut conduire à une précision améliorée de la décision de classification par rapport à une décision basée sur l'une des sources de données individuelles seules.

Ces caractéristiques hétérogènes ne peuvent pas être utilisées tous les ensembles pour entraîner un seul classifieur. Dans de tels cas, un ensemble de classifieurs peut être utilisé (Parikh, 2007), où un classifieur distinct est entraîné sur chacun des ensembles de caractéristiques indépendamment. Les décisions prises par chaque classifieur peuvent ensuite être combinées par des règles de combinaison.

Estimation de la confiance

La structure d'un système à base d'ensemble permet naturellement d'attribuer une confiance à la décision prise par un tel système. Si une grande majorité des classifieurs sont d'accord avec leurs décisions, un tel résultat peut être interprété comme l'ensemble ayant une grande confiance dans sa décision. Si, cependant, la moitié des classifieurs prend une décision et l'autre moitié une décision différente, cela peut être interprété comme l'ensemble ayant une faible confiance dans sa décision. Il convient de noter qu'un ensemble ayant une confiance élevée dans sa décision ne signifie pas que la décision est correcte, et inversement, une décision prise avec une confiance faible n'est pas forcément incorrecte. Cependant, il a été démontré qu'une décision d'ensemble correctement entraînée est généralement correcte si sa confiance est élevée, et généralement incorrecte si sa confiance est faible. En utilisant une telle approche, les décisions d'ensemble peuvent être utilisées pour estimer les probabilités a posteriori des décisions de classification (Muhlbaier et al., 2005).

5.6.2. Bagging

Le Bagging, qui représente une agrégation de bootstrap, est un algorithme intuitif et simple avec une bonne performance (Breiman, 1996) conçu pour améliorer la stabilité et la précision des algorithmes d'apprentissage automatique.

La notion de bagging consiste à découper les données d'apprentissage en échantillons et d'utiliser pour chaque échantillon un algorithme différent (Figure 2.13). On obtient ainsi un ensemble de prédictions issues de différents algorithmes.

Cet algorithme utilise une méthode appelée "bootstrapping" pour générer des ensembles différents. Le bootstrapping (Efron & Tibshirani, 1993) est une méthode de rééchantillonnage avec remise. Il consiste à choisir des exemples à partir de l'échantillon d'apprentissage pour créer de nouveaux ensembles. Ceux-ci sont appelés "échantillons d'amorçage".

Chaque échantillon d'amorçage est utilisé pour entraîner un classifieur différent du même type. Les classifieurs individuels sont ensuite combinés en prenant un simple vote à la majorité de leurs décisions. Pour une instance donnée, la classe choisie par la plupart des classifieurs est la décision d'ensemble.

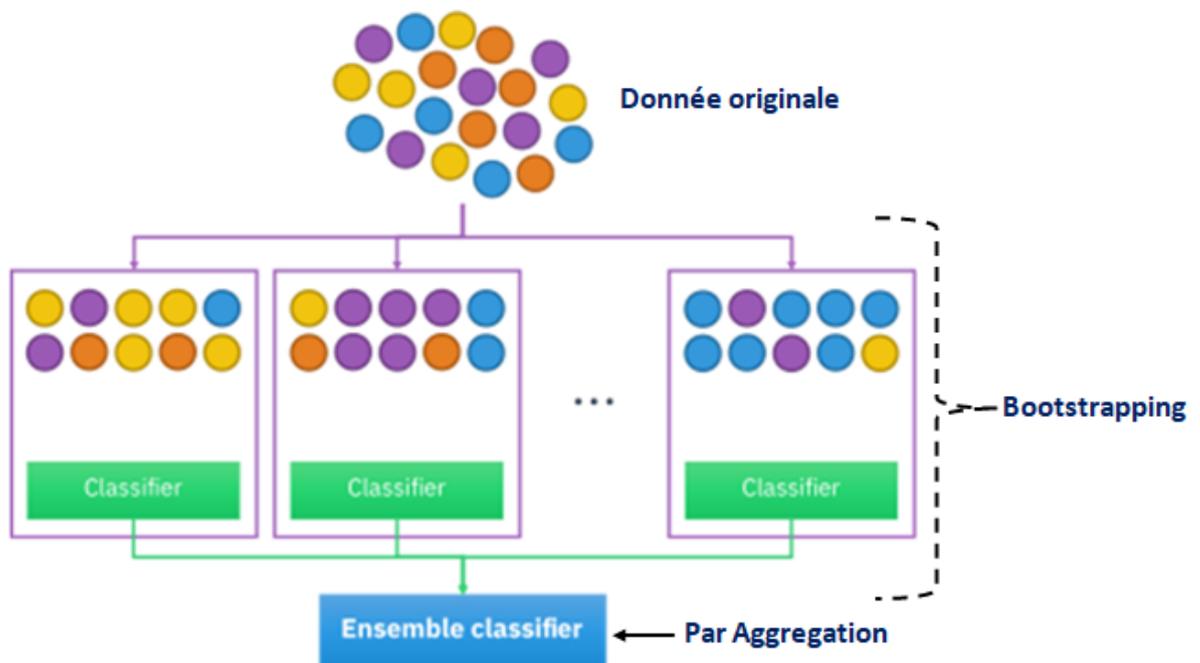


Figure 2.13 : Principe de l'approche Bagging.

5.6.3. Boosting

Le Boosting désigne un principe général d'apprentissage permettant d'améliorer la précision d'un algorithme d'apprentissage donné (Schapire, 1990). Le principe général est de combiner linéairement les résultats d'analyses dites "faibles" classifieurs afin de construire un classifieur d'apprentissage "fort" à partir de l'ensemble d'origine et une méthode de combinaison de classifieurs construits à partir de chaque nouvel ensemble (Figure 2.14). Il s'agit d'utiliser divers algorithmes pour réaliser une prédiction. Les différents algorithmes sont notés

selon leur prédiction. Plus l'algorithme prédit une bonne valeur, plus il obtient une bonne note.

Pour définir sa nouvelle technique de "Boosting", Shapire a proposé l'idée que tout classifieur faible est capable d'apprendre avec une certaine confiance et une erreur de classification inférieure à "0.5", peut être transformée en un classifieur plus sûr et avec une erreur de classification aussi petite que souhaité. À chaque itération, l'algorithme cherche à trouver un classifieur faible qui corrige au mieux les erreurs des classifieurs dans les précédentes itérations.

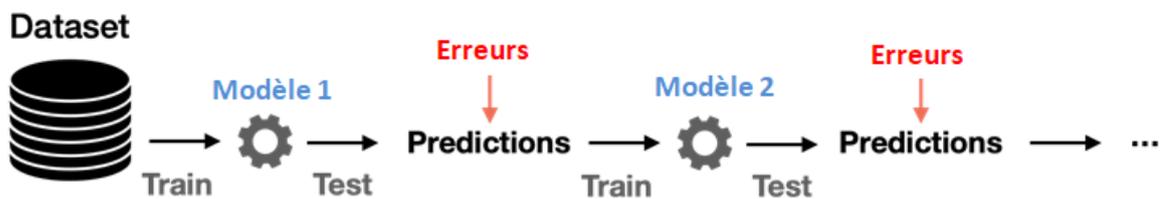


Figure 2.14 : Principe de l'approche Boosting.

En d'autres termes et à titre d'exemple, chaque itération de boosting crée trois classifieurs faibles : le premier classifieur C1 est entraîné avec un sous-ensemble aléatoire des données d'apprentissage disponibles. Le sous-ensemble de données d'apprentissage pour le deuxième classifieur C2 est choisi comme le sous-ensemble le plus informatif. C2 est entraîné sur des données d'entraînement dont seulement la moitié est correctement classée par C1, et l'autre moitié est mal classée. Le troisième classifieur C3 est entraîné avec des instances sur lesquelles C1 et C2 sont en désaccord. Les trois classifieurs sont combinés par un vote majoritaire à trois voix.

5.6.4. Forêt aléatoire

La forêt aléatoire se compose d'un grand nombre d'arbres de décision individuels qui fonctionnent comme un ensemble. C'est un algorithme de classification qui réduit la variance des prévisions d'un arbre de décision seul, améliorant ainsi leurs performances. Pour cela, il combine de nombreux arbres de décisions dans une approche de type bagging (Figure 2.15).

L'algorithme des « forêts aléatoires » a été proposé par Leo Breiman et Adèle Cutler en 2001 (Leo, 2001). Dans sa formule la plus classique, il effectue un

apprentissage en parallèle sur de multiples arbres de décision construits aléatoirement et entraînés sur des sous-ensembles de données différents. Le nombre idéal d'arbres, qui peut aller jusqu'à plusieurs centaines voire plus, est un paramètre important : il est très variable et dépend du problème. Concrètement, chaque arbre de la forêt aléatoire est entraîné sur un sous ensemble aléatoire de données selon le principe du bagging, avec un sous ensemble aléatoire de caractéristiques selon le principe des « projections aléatoires ». Les prédictions sont ensuite moyennées lorsque les données sont quantitatives ou utilisés pour un vote pour des données qualitatives, dans le cas des arbres de classification. L'algorithme des forêts aléatoires est connu pour être un des classifieurs les plus efficaces.

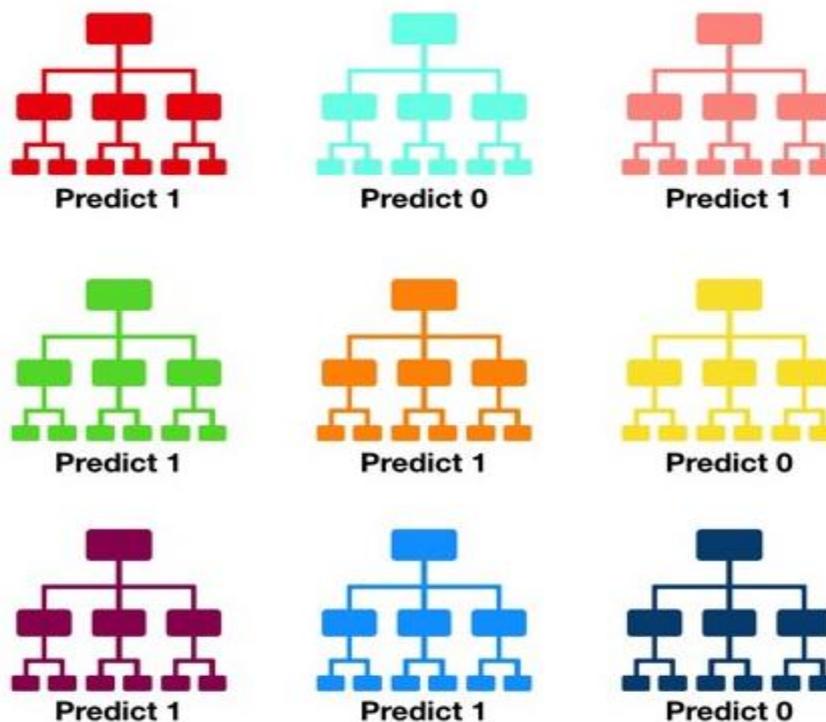


Figure 2.15 Exemple d'une forêt constituée de 6 arbres de décision.

6. Transfert Learning

Le transfert learning ou apprentissage par transfert est une méthode d'apprentissage automatique qui se concentre sur la conservation des connaissances acquises en résolvant un problème et en les appliquant à un problème différent mais connexe (Yang, 2020). Par exemple, les connaissances acquises en apprenant à reconnaître des bicyclettes pourraient s'appliquer en essayant de reconnaître des motocyclettes.

En effet, bien souvent, les modèles utilisés dans ce domaine nécessitent des temps de calcul élevés et des ressources importantes. Or, en utilisant des modèles pré-entraînés comme point de départ, le transfert learning permet de développer rapidement des modèles performants et résoudre efficacement des problèmes (Figure 2.16).

L'une des grandes motivations du transfert learning c'est qu'il faut une grande quantité de données pour avoir des modèles robustes (plus spécialement en deeplearning). Alors, si l'on peut transférer certaines connaissances acquises lors de la création d'un modèle X, on pourra utiliser moins de données pour la création d'un modèle Y. En plus, le transfert learning permet d'éviter le sur-apprentissage. En apprentissage automatique classique, les algorithmes d'apprentissage sont caractérisés par leur dépendance de la distribution des attributs en entrée, où les données source et cible doivent avoir la même représentation des données (Fleury, 2021). Le changement de la distribution des attributs exige de reprendre l'apprentissage à partir du début. Contrairement à ces algorithmes, en apprentissage profond, il est possible de transférer la connaissance à partir des modèles formés précédemment, où la distribution des paramètres (poids) des réseaux DL offre cette possibilité.

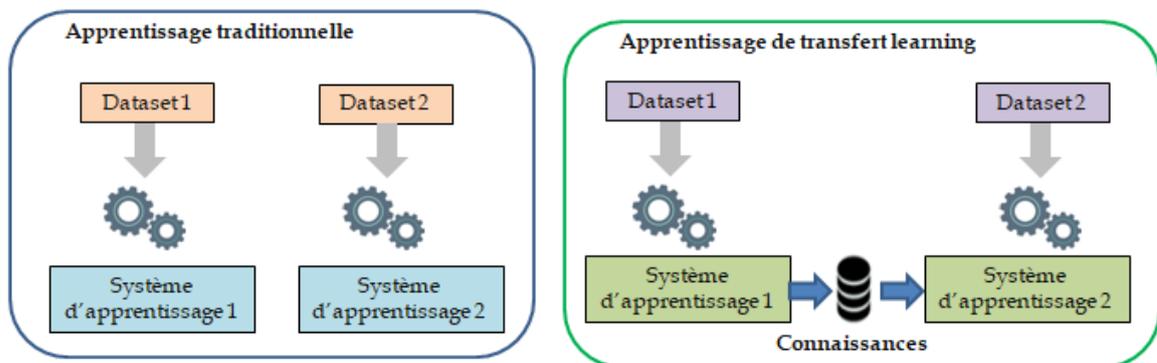


Figure 2.16 : Différence entre approche traditionnelle et approche de transfert learning.

Le transfert learning repose sur la ré-exploitation des connaissances acquises dans d'autres configurations (sources) pour la résolution d'un problème particulier (cible). Dans ce contexte, on peut distinguer plusieurs approches selon ce que l'on souhaite transférer, quand et comment réaliser le transfert. Globalement, nous pouvons distinguer trois types de transfert learning :

- **Apprentissage par transfert inductif**

Dans cette configuration, les domaines source et cible sont les mêmes (même données), mais les tâches source et cible sont différentes mais proches. L'idée consiste alors à utiliser les modèles existants pour réduire de manière avantageuse le champ d'application des modèles possibles (biais de modèle).

- **Apprentissage par transfert non supervisé**

Comme dans le cas de l'apprentissage par transfert inductif, les domaines source et cible sont similaires, mais les tâches sont différentes. Toutefois, les données des deux domaines ne sont pas labellisées.

Il est souvent plus facile d'obtenir de grandes quantités de données non labellisées, à partir de bases de données et de sources sur le web par exemple, que des données labellisées. C'est pourquoi l'idée d'utiliser l'apprentissage non supervisé en combinaison avec le transfert learning suscite un grand intérêt. A titre d'exemple, le Self-taughtclustering est une approche qui permet de réaliser le clustering de petites collections de données cibles non labellisées, avec l'aide d'une grande quantité de données sources non labellisées.

- **Apprentissage par transfert transductif**

Dans cette configuration, les tâches sources et cibles sont similaires, mais les domaines correspondants sont différents soit en termes de données ou de distributions de probabilités marginales. Par exemple, les modèles de traitement du langage naturel (NLP), tels que ceux utilisés pour l'étiquetage morpho-syntaxique de mots (Part-Of-Speech Tagger), sont généralement entraînés et testés sur des données d'actualité comme celles du Wall Street Journal. Ils peuvent être adaptés aux données issues des réseaux sociaux dont le contenu est différent mais proche de celles des journaux.

7. Conclusion

Les techniques de Machine Learning et de Deep Learning nous permettent de créer des systèmes informatiques et des applications qui effectuent des tâches généralement associées à l'intelligence humaine. Ces tâches incluent la reconnaissance d'images, le diagnostic médicale, le suivi d'objets, ...etc.

Dans ce chapitre, nous avons parcouru et décrit les différents algorithmes du Machine Learning supervisés les plus rencontrés et évoqués. Bien entendu, il en existe d'autres. Nous n'avons pas non plus évoqué les réseaux de neurones artificiels que nous aborderons dans le prochain chapitre.

Le transfert learning associé au deep learning a été également exposé dans ce chapitre. Le transfert learning se concentre sur la conservation des connaissances acquises en résolvant un problème et en les appliquant à un problème différent.

Le prochain chapitre sera consacré à l'étude des réseaux de neurones convolutionnels. Les réseaux de neurones convolutifs (CNN) est considéré comme la technologie d'apprentissage en profondeur la plus populaire en vision par ordinateur. Avec un apprentissage suffisant et appropriée, les CNN sont capables d'apprendre et d'extraire automatiquement des caractéristiques.

Chapitre 3 : Réseaux de Neurones Convolutionnels

1	Introduction.....	59
2	Réseaux de neurones artificiels	59
3	Présentation des réseaux de neurones convolutifs	60
4	Architecture générale d'un réseau de neurones convolutif.....	61
5	Régularisation des réseaux de neurones convolutifs.....	69
6	Apprentissage et optimisation des réseaux de neurones convolutifs.....	73
7	Conclusion.....	85

1. Introduction

Les techniques d'apprentissage profond sont classées en trois grandes catégories : non supervisées, partiellement supervisées (semi-supervisées) et supervisées. En outre, l'apprentissage par renforcement profond (DRL), également connu sous le nom de RL, est un autre type de technique d'apprentissage, qui est principalement considéré comme appartenant à la catégorie des techniques d'apprentissage partiellement supervisées (et parfois non supervisées).

Dans la première section, une brève présentation du réseau de neurones artificiels (RNA) est réalisée, puis dans la seconde section est détaillée leur extension aux réseaux de neurones convolutifs (Convolutional Neural Network, CNN).

Dans le même contexte la régularisation ainsi que l'apprentissage et l'optimisation des réseaux de neurones convolutifs sont discutées.

2. Réseaux de neurones artificiels

Un réseau de neurones artificiels (ANN) est un modèle informatique qui imite le fonctionnement des cellules nerveuses dans le cerveau humain. Il est conçu pour simuler la façon dont le cerveau humain analyse et traite les informations. Les réseaux de neurones artificiels sont des structures multicouches composées d'unités basiques appelées neurones. Dans ce schéma, chaque neurone ou unité est connecté à l'ensemble des unités de la couche $N - 1$ et $N + 1$ et forme un maillage.

Les neurones se partagent ainsi l'information de point à point de l'entrée vers la sortie, appliquant respectivement l'opération dont ils sont responsables. Cette opération au sein d'une unité est de la forme $y = Wx + b$ dans laquelle W représente une matrice de poids qui permet de pondérer les signaux des prédécesseurs et b un terme de correction appelé biais (Mermillod, 2016). Ainsi, les neurones en entrée d'un réseau reçoivent une information brute tandis que, les neurones situés en sortie du réseau reçoivent une information prétraitée par les prédécesseurs. Le schéma présent sur la [Figure 3.1](#) permet de mettre en évidence l'architecture d'un réseau de neurones artificiels.

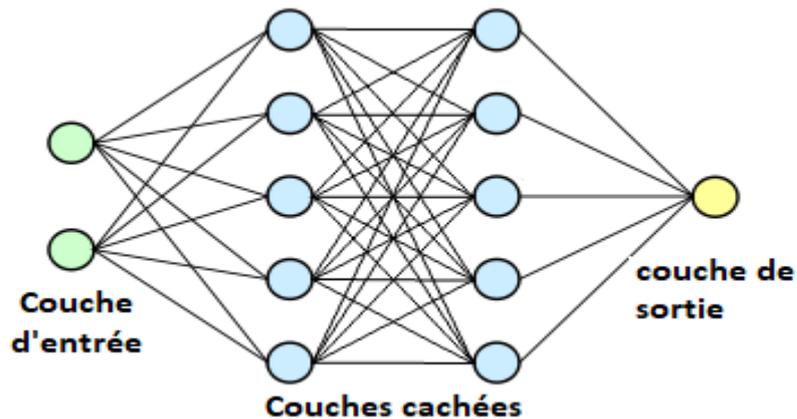


Figure 3.1. : Architecture générale d'un réseau de neurones artificiel.

On distingue différents types de réseaux de neurones. En règle générale ils sont catégorisés en fonction du nombre de nœuds cachés du modèle, ou encore du nombre d'entrées et de sorties de chaque nœud.

En fonction du type de réseau, la propagation des informations entre les différents tiers de neurones peut varier.

- Dans la variante la plus simple, celle du réseau de neurones dit « feed-forward », les informations passent directement de l'entrée aux nœuds de traitement puis aux sorties.
- Les réseaux de neurones récurrents, quant à eux, sauvegardent les résultats produits par les nœuds de traitement et nourrissent le modèle à l'aide de ces résultats.
- Enfin, les réseaux de neurones convolutifs sont de plus en plus utilisés dans différents domaines : reconnaissance faciale, numérisation de texte, traitement des images...etc.

3. Présentation des réseaux de neurones convolutifs

Les réseaux de neurones convolutifs s'inspirent des travaux de Hubel et Wiesel ([Hubel & Wiesel, 1968](#)) sur les cortex visuels des chats et des singes. En 1980, Fukushima introduit le Neocognitron ([Fukushima, 1980](#)) un réseau hiérarchique composé de nombreuses couches qui permettent d'extraire à partir des images des caractéristiques robustes aux légères déformations. En 1989, LeCun et al. ([LeCun et al., 1989](#)) proposent une architecture de perceptron multi-couches dont la première couche est convolutive, entraînée par rétro-propagation. Sur ce principe, ils élaborent ensuite l'architecture LeNet-5 ([LeCun et al., 1998](#)), premier réseau de neurones convolutif moderne dédié spécifiquement à la classification d'images de chiffres manuscrits la base de données MNIST. Malgré les succès, des ConvNets, ils ont été

en grande partie délaissés en vue de leurs architectures, en particulier concernant leur profondeur. Cette limitation s'est effondrée en 2010 avec le début de la compétition de reconnaissance d'objets ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). À cette période, on observe en effet une disponibilité accrue de puissance de calcul et une disponibilité assez nouvelle de grands volumes de données. Un million d'images sont annotées pour mille classes d'intérêt différentes. C'est ainsi qu'en 2012 la compétition est remportée par Krizhevsky, Sutskever et Hinton, qui ont proposés AlexNet (Krizhevsky et al., 2012) un réseau de neurones convolutif largement inspiré par LeNet-5 constitué de plusieurs couches convolutives dont la particularité d'utilisation a été de le faire tourner sur des processeurs graphiques(GPU) capables de puissants calculs matriciels.

Le succès des réseaux convolutifs profonds depuis 2012 est donc dû à la convergence de trois facteurs : des avancées théoriques (ReLU, réseaux convolutifs) permettant d'entraîner des réseaux plus profonds, la mise à disposition de grandes bases de données annotées pour l'apprentissage et des implémentations efficaces sur GPU rendant les temps de calcul acceptables.

4. Architecture générale d'un réseau de neurones convolutif

L'architecture d'un réseau de neurones convolutif (CNN) se compose de trois types de couches : couche de convolution, couche de pooling et couche entièrement connectée(Bezdan et al., 2019). Chaque type de couche joue un rôle différent. L'architecture d'un réseau de neurones convolutif est illustrée par la Figure 3.2.

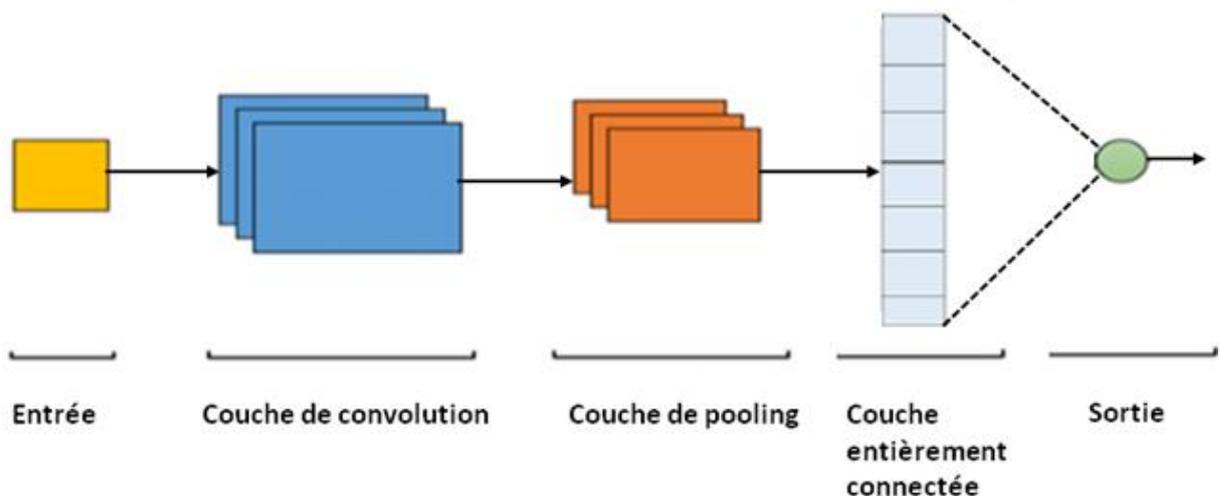


Figure 3.2. : Architecture générale d'un réseau de neurones convolutif.

4.1. Couche de convolution

Une convolution est une opération entre une matrice M et une matrice de convolution K , appelée noyau. L'opération de convolution est définie comme la somme des éléments de M pondérés par les éléments de K . Si K est une matrice carrée de dimension $2n + 1$, alors on peut définir la matrice G résultat de la convolution entre M et K par :

$$G(i, j) = \sum_{u=-n}^n \sum_{v=-n}^n K(u, v) * M(i - u, j - v) \tag{3.1}$$

On peut visualiser cette opération sur la [Figure 3.3](#), où une matrice de convolution K de taille $3 * 3$ est appliquée sur une matrice M de taille $4 * 4$. On obtient une matrice G dont la taille va dépendre de la manière dont on applique le noyau sur M .

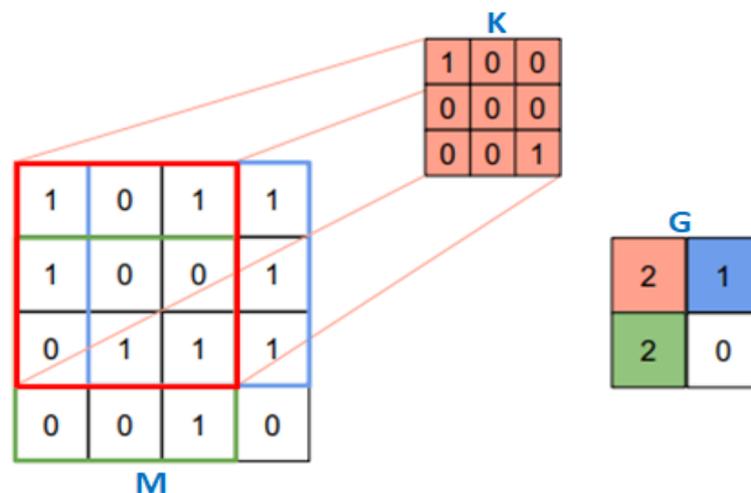


Figure 3.3 : Illustration d'une convolution sur une matrice $4 * 4$.

On peut ainsi calculer la taille de la sortie G de la convolution d'une matrice carrée de taille m par une matrice de convolution de taille $2n + 1 < m$ par $\frac{m-(2*n+1)}{s} + 1$ (S est un paramètre qui dénote le nombre de pixels par lesquels la fenêtre se déplace après chaque opération). Elle sera toujours inférieure à m , et donc la convolution va réduire la taille de la matrice d'entrée. Cela vient du fait que la matrice de convolution ne peut pas être appliquée avec pour centre la première ou la dernière ligne ou colonne. Pour corriger ce problème, on peut définir un remplissage (padding) autour de la matrice M . En ajoutant ligne et colonne de 0 autour de la matrice M on va pouvoir appliquer la convolution sur un domaine plus grand, sans

que cela ne perturbe le résultat, car en ajouter des 0 est sans effet dans l'Equation 3.1. Comme on ajoute le padding de chaque côté de la matrice, la taille de la sortie sera : $\frac{m-(2*n+1)+2p}{s} + 1$.

En prenant un padding de taille $p = n$, avec un pas de 1, on assure que la taille de la sortie sera égale à la taille de l'entrée.

Chaque couche d'un réseau convolutif profond va être définie par un ensemble de convolutions, dont la sortie peut être connectée à un neurone. Une image n'étant pas une matrice en deux dimensions, mais un ensemble de matrices, on peut voir celle-ci comme une matrice 3D de taille $L * H * C$ où L et H sont la largeur et la hauteur de l'image, et C le nombre de canaux. Si on applique un certain nombre K de convolutions sur celle-ci, on obtient une matrice 3D de taille $L * H * K$. Un ensemble de convolutions comme celui-ci va définir une couche de convolution, composant de base pour un réseau convolutif (Figure 3.4).

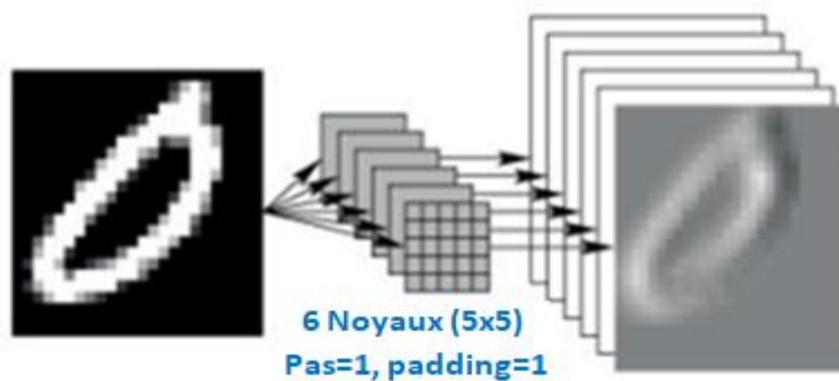


Figure 3.4 : Exemple d'une convolution.

4.2. Couche de pooling

Une couche de pooling fournit une opération de sous-échantillonnage typique qui réduit la dimensionnalité des cartes de caractéristiques. Par conséquent, cela réduit la quantité de paramètres et la complexité de calcul du réseau. Il existe généralement trois types d'opérations :

- max-pooling : sélectionne le plus grand élément dans la région de la carte de caractéristiques couverte par le noyau
- average pooling : implique le calcul de la moyenne pour chaque région de la carte de caractéristiques couverte par le noyau
- min-pooling : sélectionne l'élément minimum de la région de la carte de caractéristiques couverte par le noyau

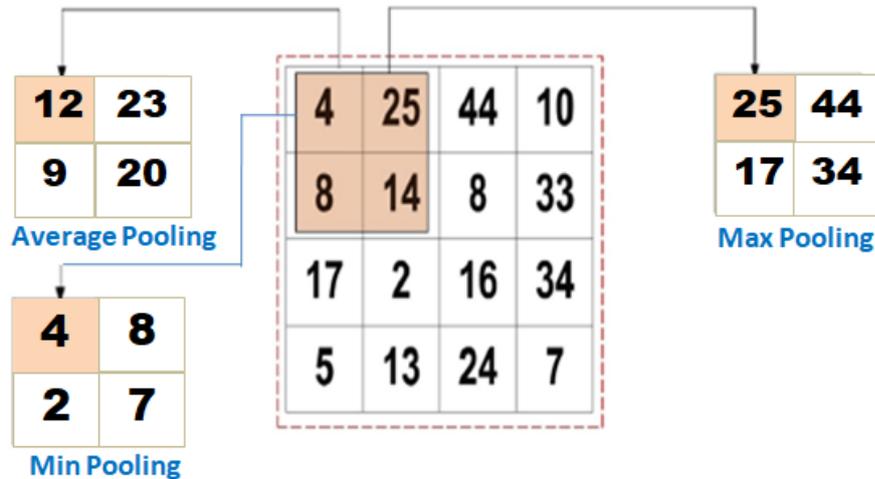


Figure 3.5 : Exemple d'application du pooling.

4.3. Fonction d'activation

La fonction d'activation sert à modifier de manière non-linéaire les données. Cette non-linéarité permet de modifier spatialement leur représentation. Elle consiste à prendre la carte des caractéristiques calculée et crée la carte d'activation en tant que sortie (feature map ou aussi activation map). La fonction d'activation prend la décision de déclencher ou non un neurone en référence à une entrée particulière en créant la sortie correspondante.

Les types de fonctions d'activation suivants sont les plus couramment utilisés dans les réseaux de neurones convolutifs.

Sigmoïde : L'entrée de cette fonction d'activation est constituée de nombres réels, tandis que la sortie est limitée entre zéro et un. La courbe de la fonction sigmoïde est en forme de S et peut être représentée mathématiquement par l'équation 3.2.

$$f(x)_{sigm} = \frac{1}{1 + e^{-x}} \quad (3.2)$$

Tanh : Elle est similaire à la fonction sigmoïde, car son entrée est constituée de nombres réels, mais la sortie est limitée entre -1 et 1. Sa représentation mathématique est donnée par l'équation 3.3.

$$f(x)_{tanh} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.3)$$

Softmax : La fonction Softmax (fonction exponentielle normalisée) permet de générer une activation strictement positive dont la somme des activations sur

l'ensemble des paramètres de la fonction sera égale à 1. Ainsi, la fonction softmax aura la forme d'une probabilité où l'activation de la fonction pour chacun de ses paramètres sera une probabilité parmi l'ensemble des probabilités. Softmax peut être représenté mathématiquement par l'équation 3.4.

$$f(x)_{softmax} = \frac{e^x}{\sum e^{x_i}} \quad (3.4)$$

ReLU (Rectified Linear Unit) : La fonction la plus couramment utilisée dans le contexte CNN. Les fonctions Sigmoid et Tanh comportent un problème lié à la saturation. Les grandes valeurs sont cantonnées à l'intervalle [0,1] et les plus petites à l'intervalle [-1,0] et une fois que la saturation est atteinte, l'apprentissage du réseau devient complexe. La fonction ReLU palie ce problème. Elle convertit les valeurs négatives de l'entrée en nombres positifs. Elles permettent un apprentissage plus rapide comparé aux fonctions Sigmoid et Tanh. Sa représentation mathématique est donnée par l'équation 3.5.

$$f(x)_{ReLU} = \max(0, x) \quad (3.5)$$

Parfois, quelques problèmes importants peuvent survenir lors de l'utilisation de la fonction ReLU. Lorsque cette fonction fait face à une valeur négative, la valeur 0 est automatiquement attribuée, ce qui entraîne de surcroît un non-apprentissage du réseau (Dying ReLU). Certaines alternatives ReLU existent pour résoudre ce problème. Ce qui suit en traite quelques-unes.

Leaky ReLU : Au lieu que ReLU réduise les entrées négatives, cette fonction d'activation garantit que ces entrées ne sont jamais ignorées. Elle est utilisée pour résoudre le problème Dying ReLU. Leaky ReLU peut être représenté mathématiquement par l'équation 3.6.

$$f(x)_{LeakyReLU} = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & x \leq 0 \end{cases} \quad (3.6)$$

α est le facteur de fuite généralement il prend une valeur petite : 0.001.

PReLU (Parametric ReLU) : La paramétrique Leaky Relu permet quant à elle de définir alpha (Equation 3.6) comme paramètre du modèle et non plus comme hyperparamètre. Il sera ainsi modifié durant la rétropropagation du gradient.

ELU (Exponential Linear Unit) : Autre dérivé de la ReLU. Celle-ci va approcher les valeurs moyennes proches de 0, ce qui va avoir comme impact d'améliorer les performances d'entraînements. Elle utilise l'exponentiel pour la partie négative et non plus une fonction linéaire. Elle paraît plus performante en expérimentation que les autres variantes de Relu. ELU peut être représenté mathématiquement par l'équation 3.7.

$$f(x)_{ELU} = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases} \quad (3.7)$$

α un scalaire positif, qui permet de contrôler la pente de ELU lorsque $x < 0$. Plus α est grand, plus la courbe est pentue.

SeLU (Scaled ELU) : est une optimisation de ELU mais avec en plus un paramètre « *scale* » prédéfini. Bon résultat, bonne vitesse, et évite les problèmes d'explosion et disparition de gradients en s'auto normalisant et gardant les mêmes variances pour les sorties de chaque couche, et ce tout au long de l'entraînement. SELU peut être représenté mathématiquement par l'équation 3.8.

$$f(x)_{SELU} = scale * f(x)_{ELU} \quad (3.8)$$

scale est un scalaire supérieur à 1. Cela permet à la pente de SELU sur $x > 0$ d'être supérieur à 1 et évite certains problèmes lors du calcul de gradient.

Swish Activation : L'équipe Google Brain a suggéré une activation plus récente qui attendance à mieux fonctionner pour les CNN que ReLU qui est l'activation Swish. Swish est une multiplication d'une activation linéaire et d'une activation sigmoïde, représenté mathématiquement par l'équation 3.9.

$$Swish(x) = x * sigmoid(x) \quad (3.9)$$

Afin de conclure les trois points abordés précédemment (convolution, fonctions d'activation et pooling), un exemple illustratif est montré par la [Figure 3.6](#).

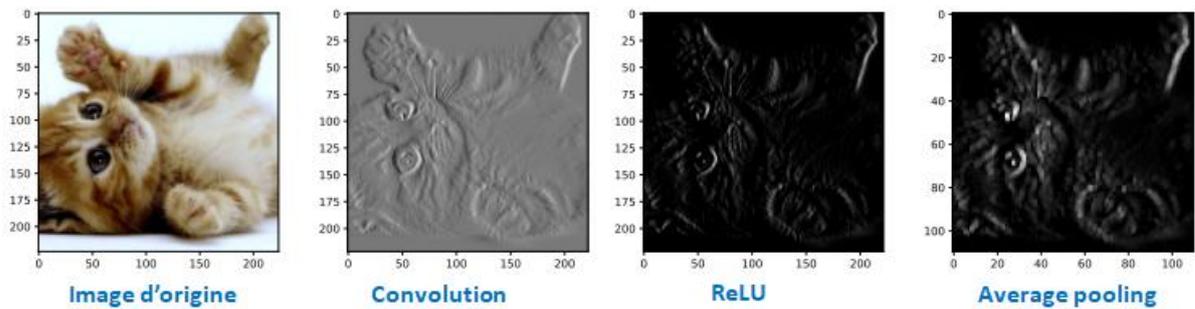


Figure 3.6 : Illustration des opérations de convolution, de ReLU et de pooling.

4.4. Couche entièrement connectée

Cette couche se trouve à la fin de chaque architecture CNN. À l'intérieur de cette couche, chaque neurone est connecté à tous les neurones de la couche précédente, l'approche est dite Fully Connected (FC). Elle suit le modèle base du réseau de neurones perceptron multicouches. L'entrée de la couche FC provient de la dernière couche pooling ou de convolution. Cette entrée est sous la forme d'un vecteur, qui est créé à partir des cartes de caractéristiques après aplatissement (flattening). La sortie de la couche FC représente la sortie CNN finale, comme le montre la [Figure 3.7](#).

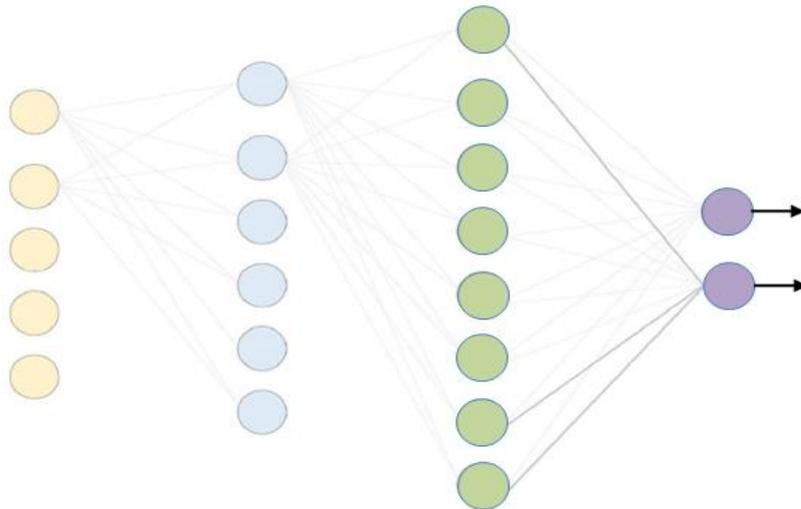


Figure 3.7 : Couche entièrement connectée.

4.5. Fonctions de perte

Une fonction de perte (Loss function), aussi appelée fonction d'erreur ou fonction de coût est une fonction qui évalue l'écart entre les prédictions réalisées par le réseau de neurones convolutif et les valeurs réelles des observations utilisées pendant l'apprentissage. Plus le résultat de cette fonction est minimisé, plus le réseau de neurones convolutif est performant. Sa minimisation, c'est-à-dire réduire au

minimum l'écart entre la valeur prédite et la valeur réelle pour une observation donnée, se fait en ajustant les différents poids du réseau de neurones convolutif (Vannieuwenhuyze, 2019).

Une fonction de perte $L : Y \times Y \rightarrow R$ est une fonction utilisée pour quantifier la qualité d'une prédiction : $L(y, f(x))$ est d'autant plus grande que l'étiquette $f(x)$ est éloignée de la vraie valeur y .

Étant donnée une fonction de perte L , nous cherchons donc f qui minimise cette perte sur l'ensemble des valeurs possibles de $x \in X$.

Il existe de nombreuses fonctions de perte. Le choix d'une fonction de perte dépend d'une part du problème en lui-même, autrement dit de ce que l'on trouve pertinent pour le cas pratique considéré, et d'autre part de considérations pratiques : peut-on ensuite résoudre le problème d'optimisation qui résulte de ce choix de façon suffisamment exacte et rapide ? Cette section présente les fonctions de perte les plus couramment utilisées.

4.5.1. Entropie croisée

Cette fonction est couramment utilisée pour mesurer la différence entre deux probabilités de distribution pour une variable. Elle est utilisée pour la classification binaire ou la classification multi-classes. La représentation mathématique de la fonction de perte d'entropie croisée est donnée par l'équation suivante :

$$L(y, f(x)) = - \sum_{i=1}^N y_i \log f(x_i) \quad (3.10)$$

N représente le nombre de neurones dans la couche de sortie.

$f(x)$ la valeur estimée et y la valeur réelle.

4.5.2. Erreur quadratique moyenne

L'erreur quadratique moyenne mesure la moyenne des carrés des erreurs, c'est-à-dire la différence quadratique moyenne entre les valeurs estimées et la valeur réelle.

L'expression mathématique est donnée par l'équation suivante :

$$L(y, f(x)) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2 \quad (3.11)$$

N représente le nombre de neurones dans la couche de sortie.

$f(x)$ la valeur estimée et y la valeur réelle.

4.5.3. Fonction de perte hinge

Cette fonction est couramment utilisée dans les problèmes liés à la classification binaire. Ce problème est lié à la classification basée sur la marge maximale; c'est surtout important pour les SVMs, qui utilisent cette fonction de perte, où dans l'optimiseur tente de maximiser la marge autour des classes à objectif dual. Sa formule mathématique est donnée par l'équation :

$$L_{hinge}(y, f(x)) = \max(0, 1 - yf(x)) \quad (3.12)$$

5. Régularisation des réseaux de neurones convolutifs

La régularisation est l'ensemble des méthodes qui permettent à l'algorithme de réduire son erreur de généralisation. Lorsque l'on réalise un algorithme d'apprentissage machine, on souhaite que celui-ci n'ait pas uniquement de bonnes performances sur les données d'apprentissages, mais sur toutes données qu'on lui présente.

Les facteurs déterminant les performances d'un algorithme d'apprentissage machine sont sa capacité à :

- Réduisez l'erreur d'entraînement.
- Réduisez l'écart entre l'erreur d'entraînement et l'erreur de test.

Ces deux facteurs correspondent aux deux enjeux centraux de l'apprentissage machine : sous-apprentissage et sur-apprentissage.

Le sous- apprentissage se produit lorsque le modèle n'est pas en mesure d'obtenir une valeur d'erreur suffisamment faible sur l'ensemble d'apprentissage. Le sur-apprentissage se produit lorsque l'écart entre l'erreur d'apprentissage et l'erreur de test est trop grand (Figure 3.8).

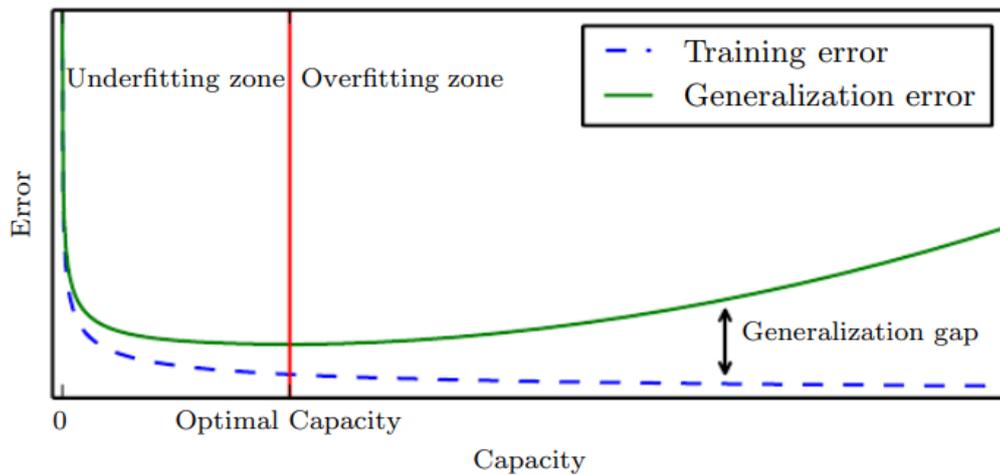


Figure 3. 8 : Relation typique entre la capacité et l'erreur.

La généralisation est la capacité d'un réseau de neurones convolutif, et de n'importe quel algorithme d'apprentissage automatique, à bien se comporter sur des données qu'il n'a pas vu pendant l'entraînement. Généralement mesuré grâce à un corpus de test séparé du corpus d'entraînement.

Un exemple extrême d'un algorithme qui généralise mal, est un réseau capable d'atteindre 100% de reconnaissance sur les données fournies pendant l'apprentissage et 0% sur le corpus de test. On parle de sur-apprentissage (Vannieuwenhuyze, 2019).

- Le sur-apprentissage (overfitting) est un phénomène se traduisant par le fait que la solution est trop adaptée aux données d'apprentissage et ne se généralise pas à de nouvelles données qui lui sont inconnues (Figure 3.9.a).
- Le sous-apprentissage (underfitting), sous-entend que le modèle prédictif généré lors de la phase d'apprentissage, s'adapte mal aux données d'apprentissage. Autrement dit, le modèle prédictif n'arrive même pas à capturer les corrélations de la base d'apprentissage (Figure 3.9.b).

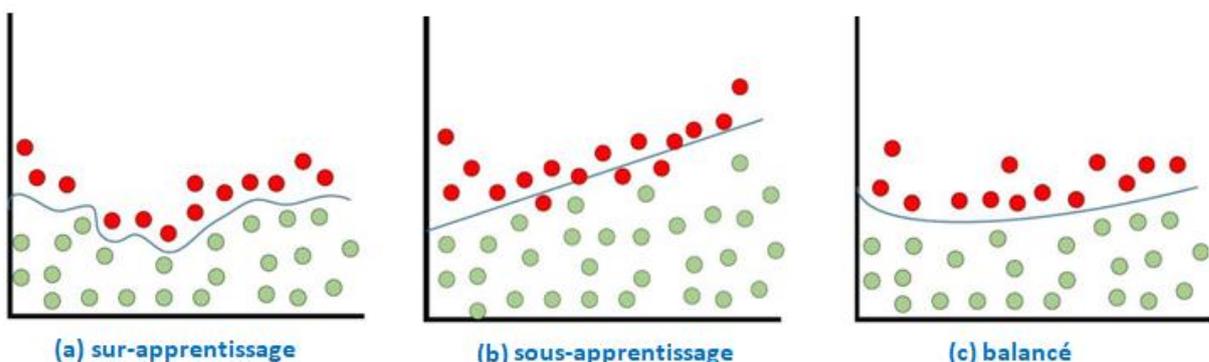


Figure 3.9 : Différentes formes d'apprentissage.

Divers concepts intuitifs sont utilisés pour aider à la régularisation afin d'éviter le sur-apprentissage. Il existe de nombreuses formes de régularisation, qui dépendent de l'objectif recherché et des hypothèses fixées sur le problème. Cette section présente les approches les plus couramment utilisées.

5.1. Dropout

Il s'agit de masquer une partie du réseau aléatoirement pendant l'apprentissage (Srivastava et al., 2014). Pour cela, chaque neurone du réseau a une probabilité p d'être abandonné (dropped) à chaque passage. La Figure 3.10 montre à gauche le réseau initial, et à droite un exemple avec 7 neurones abandonnés (X). Une fois l'apprentissage terminé avec ces coupes aléatoires, toutes les connexions sont utilisées lorsque l'on déploie le réseau. Ceci entraîne une activation totale supérieure, car aucune connexion n'est ignorée. Pour garder une activation du même ordre de grandeur que pendant l'entraînement, elles sont toutes réduites d'un facteur f .

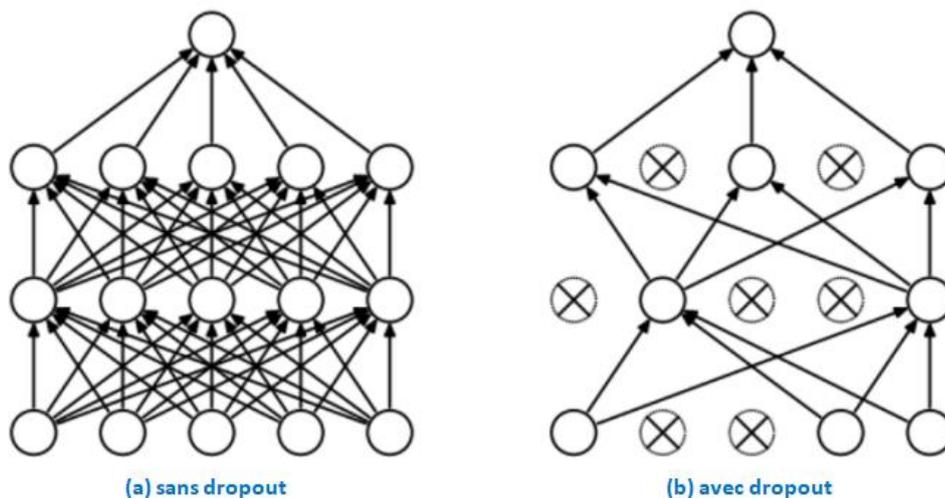


Figure 3.10 : Exemple illustratif du dropout.

5.2. Drop-Weight

Cette méthode est très similaire au dropout (Krogh & Hertz, 1991). Dans chaque étape d'apprentissage, les connexions entre les neurones (poids) sont abandonnées plutôt qu'abandonner les neurones ; ceci représente la seule différence entre les poids abandonnés et le l'activation des neurones.

5.3. Weight decay

Une forme de régularisation importante et utiliser dans tous les apprentissages aujourd'hui est le "weight decay", ou la normalisation L2 des poids. Les poids sur

les connexions de chaque neurones peuvent prendre des valeurs de forte amplitude pendant l'apprentissage, or cela est pénalisant pour la généralisation, en créant une variance de sortie plus élevée (Krogh & Hertz, 1991). On souhaite donc forcer les poids à rester petits en pénalisant les grandes valeurs. En utilisant la norme L2 avec un paramètre α :

$$W = \alpha \sum_{i=1}^N w_i \quad (3.13)$$

On force ainsi les poids à rester dans une sphère de rayon α . Ce paramètre est un hyperparamètre à optimiser pendant l'apprentissage, et va dépendre de chaque réseau, avec des valeurs généralement faibles entre 10^{-3} et 10^{-4} .

5.4. Augmentation des données

L'apprentissage du modèle sur une masse de données importante est le moyen le plus simple pour éviter le sur-apprentissage. Pour ce faire, des données supplémentaires sont utilisées. Plusieurs techniques sont utilisées pour élargir artificiellement la taille de l'ensemble de données d'apprentissage (Taylor & Nitschke, 2018). On peut citer par exemple le recadrage aléatoire, où l'on prend aléatoirement une zone de l'image plutôt que de mettre l'image à l'échelle pour l'entrée du réseau. On utilise également fréquemment des rotations des images, avec des renversements aléatoire de l'image, entre $[-180^\circ; 180^\circ]$. Il est également possible de modifier les contrastes, la teinte ou la saturation, toujours dans le but de déformer les images pour montrer le plus d'exemples différents possible au réseau.

5.5. Normalisation par lots

La normalisation par batch étend essentiellement la logique de normalisation de l'entrée du réseau neuronal à la normalisation de l'entrée de chaque couche cachée du réseau (Ioffe & Szegedy, 2015). L'idée de base est d'avoir une distribution fixe qui alimente chaque couche suivante d'un réseau neuronal puisque l'apprentissage se fait mieux lorsque nous avons une distribution fixe. Pour ce faire, nous calculons la moyenne et la variance de chaque batch avant chaque couche cachée et nous normalisons les valeurs entrantes par ces statistiques spécifiques aux batches, ce qui réduit la quantité par laquelle les valeurs vont finalement se déplacer pendant l'entraînement.

Les avantages de la normalisation par lots sont les suivants :

- empêche le problème de la disparition du gradient.
- peut contrôler efficacement la mauvaise initialisation du poids.
- réduit considérablement le temps nécessaire à la convergence des réseaux (pour les ensembles de données à grande échelle).
- permet de réduire la dépendance à l'égard de l'apprentissage à travers les hyperparamètres.
- Réduire le sur-ajustement, car il a une influence mineure sur la régularisation

6. Apprentissage et optimisation des réseaux de neurones convolutifs

L'apprentissage consiste à ajuster les poids d'un réseau de neurones convolutif afin d'obtenir les meilleurs résultats de classification. L'objectif de cette optimisation est de minimiser la fonction de coût (Section 3.5). Les paramètres réseau doivent toujours être mis à jour à toutes les étapes d'apprentissage, tandis que le réseau doit également rechercher la réponse optimisée localement dans toutes les étapes d'apprentissage afin de minimiser l'erreur.

Deux problèmes majeurs sont inclus dans le processus d'apprentissage : le premier est la sélection de l'algorithme d'apprentissage (optimiseur), tandis que le second est l'utilisation de nombreuses améliorations (telles que AdaDelta, Adagrad et momentum) avec l'algorithme d'apprentissage pour améliorer le résultat.

Ce qui suit décrit l'algorithme de descente de gradient et ces variantes qui améliorent encore puissamment le processus d'apprentissage des réseaux de neurones convolutifs.

6.1. Descente de gradient

Minimum local et minimum global

Lorsqu'un système est en phase d'apprentissage, il commet des erreurs. Le taux d'erreur diminue au fur et à mesure de l'apprentissage, mais il se peut qu'à un moment donné l'erreur augmente pour à nouveau rediminuer et atteindre un niveau d'erreur plus bas que le précédent qui est le niveau optimal d'apprentissage.

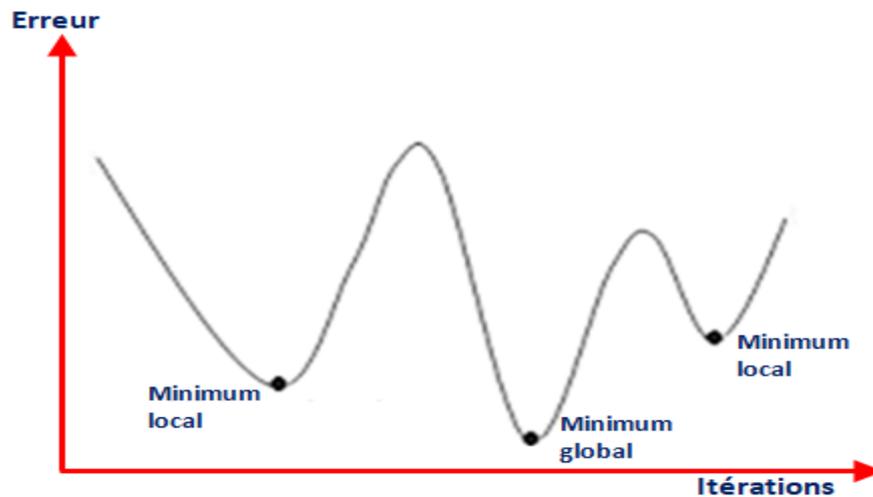


Figure 3.11 : Minimum local et minimum global.

Sur la figure précédente, on constate qu'en début d'apprentissage, l'erreur diminue progressivement pour ensuite remonter. Nous aurions donc tendance à dire que le niveau optimal d'apprentissage a été atteint puisque de nouvelles erreurs apparaissent. Cependant, on peut s'apercevoir qu'après de nouvelles itérations d'apprentissage, l'erreur continue de diminuer pour atteindre un niveau plus bas que le précédent appelé minimum global. Le niveau optimal d'apprentissage n'était donc pas atteint. L'algorithme du gradient consiste donc à trouver par itérations successives le minimum global de la fonction de coût (erreur).

Algorithme de descente de gradient

La descente de gradient est un algorithme d'optimisation qui permet de trouver le minimum de n'importe quelle fonction convexe en convergeant progressivement vers celui-ci (Fausett, 1994). L'algorithme permet d'approcher son minimum local en cherchant les points auxquels son gradient est nul (Figure 3.12).

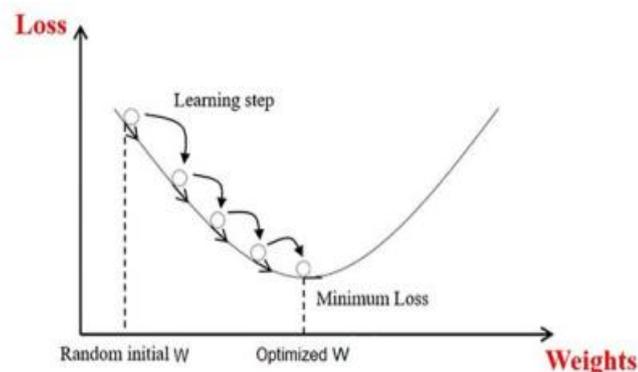


Figure 3.12 : Diagramme de la descente du gradient.

Pour minimiser l'erreur d'entraînement, cet algorithme met à jour les paramètres du réseau de manière répétitive à chaque période d'entraînement. Plus précisément, pour mettre à jour correctement les paramètres, il doit calculer le gradient de la fonction objectif (pente) en appliquant un dérivé de premier ordre par rapport aux paramètres du réseau. Ensuite, le paramètre est mis à jour dans le sens inverse du gradient pour réduire l'erreur.

Soit θ le vecteur contenant tous les paramètres du réseau CNN. Soit $J(\theta, f(x), y)$ la fonction de coût représentant l'erreur entre la vérité terrain $f(x)$ (liée à la donnée d'entraînement x) et la valeur prédite y estimée par le réseau en utilisant les paramètres θ .

Un pas de l'itération de la descente de gradient est formulé par :

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta, f(x), y) \quad (3.14)$$

Où η est le taux d'apprentissage, considéré comme l'un des hyperparamètres importants pour l'optimisation des réseaux de neurones profonds en agissant sur sa convergence. En effet, un taux d'apprentissage trop élevé conduit à des mises à jour des poids importantes et la convergence devient instable. Par contre, pour un taux d'apprentissage faible, la convergence est ralentie avec une possibilité de tomber dans des minimas locaux. L'approche populaire utilisée dans l'apprentissage profond pour avoir le taux d'apprentissage optimal est de commencer l'apprentissage avec une valeur élevée afin d'accélérer la descente du gradient et de la réduire par la suite pour améliorer la précision (Bengio, 2012). Pratiquement, il s'agit d'initialiser un η_0 à une valeur élevée au début et la faire diminuer ensuite par un facteur multiplicatif constant γ pendant la phase de l'apprentissage selon la formule (3.15), jusqu'à ce que l'erreur de validation atteigne une valeur stable ou lorsque l'erreur de l'apprentissage ne décroît plus (Bottou, 2012).

$$\eta_{i+1} = \eta_0 (1 + \gamma_i)^{-1} \quad (3.15)$$

En apprentissage profond, le nombre de paramètres à apprendre devient très vite important. Les arguments de la fonction de coût sont alors de très grande dimension, on observe alors une prolifération de points-selles, lesquels peuvent être très pénalisant quand on cherche un minimum car ils sont, en général, entourés de plateaux (Dauphin et al., 2014). Différentes alternatives de l'algorithme d'apprentissage basé sur le gradient sont disponibles et couramment utilisées en

apprentissage des réseaux de neurones convolutionnels; nous pouvons citer : Descente de gradient stochastique, Descente de gradient en lots et Descente de gradient en mini-lots.

6.2. Descente de gradient stochastique

En apprentissage profond, la fonction objective que l'on cherche à minimiser est souvent non convexe et non régulière. La convergence de la descente du gradient vers le minimum global n'est donc pas garantie et la convergence même vers un minimum local peut être extrêmement lente. Une solution à ce problème consiste en l'utilisation de l'algorithme de descente de gradient stochastique (SGD). L'idée de l'approche est de chercher à minimiser une fonction qui peut être écrite sous la forme de la somme de fonctions différentiables. Ce processus est alors réalisé de manière itérative sur des lots de données tirés aléatoirement. Chaque fonction objective minimisée de cette manière est une approximation de la fonction objective globale. Les paramètres du réseau sont mis à jour à chaque échantillon d'entraînement (Bottou et al., 2010). L'équation suivante décrit cette méthode :

$$\theta_{t+1} = \theta_t - \eta \frac{1}{B_s} \sum_{i=1}^{B_s} \nabla_{\theta} J_t^i(\theta_t, f(x_t^i), y_t^i) \quad (3.16)$$

où B_s est la taille du lot, $B_t = (x_t^i, y_t^i)_{i \in [1, B_s]}$ est le lot de données tiré à l'étape t , $f(x_t^i)$ et y_t^i sont respectivement la donnée estimée et la donnée de la vérité terrain dans ce lot, et $J_t = \frac{1}{B_s} \sum_{i=1}^{B_s} J_t^i$ est l'approximation stochastique de la fonction de coût globale à l'étape t sur le lot B_t , décomposée en somme de fonctions différentiables J_t^i liées à chaque paire (x_t^i, y_t^i) .

Pour un ensemble de données d'apprentissage de grande taille, cette technique est à la fois plus efficace en mémoire et beaucoup plus rapide que la descente de gradient par lots. Cependant, parce qu'il est fréquemment mis à jour, il prend des mesures extrêmement bruyantes dans la direction de la réponse, qui à son tour provoque le comportement de convergence de devenir très instable.

Avantages

- Les mises à jour fréquentes donnent immédiatement un aperçu des performances du modèle et du taux d'amélioration.
- Plus simple à comprendre et à mettre en œuvre.

- L'augmentation de la fréquence de mise à jour du modèle peut entraîner un apprentissage plus rapide.
- Le processus de mise à jour bruyant peut permettre au modèle d'éviter les minima locaux (par exemple, la convergence prématurée).

Inconvénients

- La mise à jour si fréquente du modèle est plus coûteuse en termes de calcul que d'autres configurations de descente de gradient, ce qui prend beaucoup plus de temps pour entraîner des modèles sur de grands ensembles de données.
- Les mises à jour fréquentes peuvent entraîner un signal de gradient bruyant, ce qui peut faire sauter les paramètres du modèle et, par conséquent, l'erreur du modèle (avoir une variance plus élevée sur les époques d'apprentissage).
- Le processus d'apprentissage bruyant le long du gradient d'erreur peut également rendre difficile de fixer sur un minimum d'erreur pour le modèle.

6.3. Descente de gradient stochastique avec Momentum

Cette technique permet d'améliorer à la fois la précision et la vitesse d'apprentissage en additionnant le gradient calculé à l'étape d'apprentissage précédente, qui est pondéré par un facteur λ appelé facteur momentum (Sutskever et al., 2013).

Le principal inconvénient des algorithmes d'apprentissage basés sur les gradients c'est se retrouver coincée dans un minimum local plutôt que global. Des problèmes de ce genre se produisent fréquemment si le problème n'a pas de surface convexe (ou d'espace de solution). Avec l'algorithme d'apprentissage, le momentum est utilisé pour résoudre ce problème.

L'objectif principal de la méthode de Momentum est d'accélérer le processus de descente de gradient, et ceci en rajoutant un vecteur de vélocité à l'expression initiale :

$$\begin{aligned}\theta_{t+1} &= v_{t+1} + \theta_t \\ v_{t+1} &= \lambda v_t - \eta \nabla J(\theta_t)\end{aligned}\tag{3.17}$$

Le vecteur v_{t+1} est calculé au début de chaque itération et représente la mise à jour de la vélocité. La vélocité s'accumule à chaque itération d'où l'introduction d'un hyper-paramètre λ permettant d'amortir la vélocité quand on atteint une surface plate. Une bonne stratégie peut être de modifier λ en fonction du niveau

d'apprentissage. La valeur du facteur de momentum est maintenue dans la plage $[0,1]$.

6.4. Descente de gradient accélérée de Nesterov

Nesterov proposa une modification de la méthode du Momentum et montra que son algorithme présente une meilleure convergence théorique pour l'optimisation des fonctions convexes (Nesterov, 1983). Cette approche devint très populaire grâce à ses performances en pratique comparée à la méthode classique.

La principale différence entre la méthode de Nesterov et la méthode du momentum réside dans le fait que cette dernière commence par calculer le gradient à l'emplacement courant θ_t avant de faire un pas dans la direction de la vitesse accumulée, alors que le momentum de Nesterov fait d'abord un pas de calcul pour obtenir une approximation du paramètre mis à jour, que l'on note $\tilde{\theta}_{t+1}$, et corrige ensuite ce pas en calculant le gradient à cet emplacement. Un pas du momentum de Nesterov est décrit par :

$$\begin{aligned}\tilde{\theta}_{t+1} &= \mu v_t + \theta_t \\ v_{t+1} &= \mu v_t - \eta \nabla J(\tilde{\theta}_{t+1}) \\ \theta_{t+1} &= v_{t+1} + \theta_t\end{aligned}\tag{3.18}$$

6.5. Descente de gradient par lots

La descente de gradient par lots (BGD) est une variante de l'algorithme de descente de gradient qui calcule l'erreur pour chaque exemple dans l'ensemble de données d'apprentissage, mais ne met à jour les paramètres du réseau qu'après l'évaluation de tous les exemples d'apprentissage.

Plus en profondeur, il calcule le gradient de l'ensemble d'apprentissage et utilise ensuite ce gradient pour mettre à jour les paramètres. Pour un ensemble de données de petite taille, le réseau de neurones convolutionnel converge plus rapidement et crée un gradient extra-stable. Étant donné que les paramètres ne sont modifiés qu'une seule fois pour chaque étape d'apprentissage, il faut beaucoup de ressources. En revanche, pour un grand ensemble de données d'entraînement, un temps supplémentaire est nécessaire pour converger, et il pourrait converger vers un optimum local (pour les cas non convexes).

Avantages

- Plus efficace en termes de calcul.
- Le poids est un paramètre apprenable : chaque fois que nous essayons de calculer un nouveau poids, nous essayons de considérer toutes les données qui sont à notre disposition sur la base de la sommation de la perte. Donc, nous essayons de trouver ou de dériver la nouvelle valeur du poids.

Inconvénients

- La consommation de mémoire est trop élevée : nous essayons d'envoyer toutes les données à l'intérieur du réseau une par une, nous avons donc besoin d'une sorte de mémoire pour stocker une perte que nous avons reçue à chaque itération. Une fois que nous avons fini de transmettre des ensembles de données à travers le réseau, nous calculons la perte. Donc, dans ce cas, la consommation de mémoire sera trop élevée, et cela se produit à chaque étape.
- Si la consommation de mémoire est trop élevée, on peut dire que la concurrence sera forte et le calcul sera très lent et donc l'optimisation sera plus lente par rapport à tout autre optimiseur.

6.6. Descente de gradient en mini-lots

La descente de gradient en mini-lot (MGD) est une variante de l'algorithme de descente de gradient qui divise l'ensemble de données d'apprentissage en petits lots qui sont utilisés pour calculer l'erreur du modèle et mettre à jour les coefficients du modèle.

Dans cette approche, les échantillons d'apprentissage sont divisés en plusieurs mini-lots, dans lesquels chaque mini-lot peut être considéré comme un ensemble d'échantillons trop petit sans chevauchement entre eux. Ensuite, la mise à jour des paramètres est effectuée après le calcul du gradient sur chaque mini-lot.

La descente de gradient en mini-lot cherche à trouver un équilibre entre la robustesse de la descente de gradient stochastique et l'efficacité de la descente de gradient en lot. Il s'agit de l'implémentation la plus courante de la descente de gradient utilisée dans le domaine de l'apprentissage profond. Ainsi, il a une convergence constante, une plus grande efficacité de calcul et une efficacité de mémoire supplémentaire.

Avantages

- La fréquence de mise à jour du modèle est supérieure au BGD. Nous n'attendons pas des données entières, nous passons juste un nombre d'échantillons limité, puis nous passons à l'optimisation.
- Le batching permet à la fois l'efficacité de ne pas avoir toutes les données d'apprentissage en mémoire.
- Les mises à jour par mini-lots fournissent un processus de calcul plus efficace que SGD.

Inconvénients

- Aucune garantie de convergence de l'erreur.
- Étant donné qu'un petit ensemble d'échantillons que nous prenons, ne représente pas les propriétés (ou la variance) d'ensembles de données entières. C'est la raison pour laquelle nous ne serons pas en mesure d'obtenir une convergence vers le minima global ou local.
- Puisque nous prenons des échantillons par mini-lots, il peut donc arriver que dans certains mini-lots, nous obtenions une erreur et que dans d'autres mini-lots, nous obtenions une autre erreur. Ainsi, nous devons contrôler le taux d'apprentissage par nous-mêmes. Si le taux d'apprentissage est très faible, le taux de convergence chutera également. Si le taux d'apprentissage est trop élevé, nous n'obtiendrons pas de minima global ou local absolu.

6.7. Algorithmes avec taux d'apprentissage adaptatifs

Le taux d'apprentissage est un hyperparamètre plus difficile à définir car il a un impact significatif sur les performances du modèle. Plus récemment, un certain nombre de méthodes incrémentales (ou basées sur des mini-lots) ont été introduites qui adaptent les taux d'apprentissage des paramètres du modèle. Cette section passera brièvement en revue quelques-uns de ces algorithmes.

6.7.1. AdaGrad

Le principe de cette méthode, proposée en 2011 par Duchi ([Duchi et al., 2011](#)), est de faire que le taux d'apprentissage s'adapte aux paramètres, faisant de sorte qu'il s'ajuste automatiquement, en fonction de "l'éparsité" des paramètres. Adagrad abaisse progressivement le taux d'apprentissage mais pas de la même manière pour tous les paramètres : les dimensions à pente plus prononcée voient leur taux abaissé plus rapidement que celles à pente douce. Les paramètres avec la plus grande dérivée partielle de la perte ont une diminution proportionnellement rapide de leur

taux d'apprentissage, tandis que les paramètres avec de petites dérivées partielles ont une diminution relativement faible de leur taux d'apprentissage.

Plus formellement, le pas est décrit par :

$$(\theta_{t+1})_i = (\theta_t)_i - \alpha \frac{(\nabla J(\theta_t))_i}{\sqrt{\sum_{u=1}^t (\nabla J(\theta_u))_i^2}} \quad (3.19)$$

AdaGrad est conçu pour converger rapidement lorsqu'il est appliqué à une fonction convexe. Lorsqu'il est appliqué à une fonction non convexe pour entraîner un réseau de neurones, la trajectoire d'apprentissage peut traverser de nombreuses structures différentes et finalement arriver à une région localement convexe.

Empiriquement, il a été constaté que, pour l'apprentissage du réseau de neurones profonds, l'accumulation de gradients au carré depuis le début de l'apprentissage peut entraîner une diminution prématurée et excessive du taux d'apprentissage effectif. AdaGrad réduit le taux d'apprentissage en fonction de l'historique complet du gradient au carré et peut avoir rendu le taux d'apprentissage trop faible avant d'arriver à une structure aussi convexe.

AdaGrad fonctionne bien pour certains modèles d'apprentissage profonds mais pas pour tous.

6.7.2. RMSprop

L'algorithme RMSProp (Root Mean Square Propagation) modifie AdaGrad pour qu'il fonctionne mieux dans le cadre non convexe en transformant l'accumulation de gradient en une moyenne mobile pondérée de manière exponentielle (Hinton, 2012).

RMSprop ajuste automatiquement le taux d'apprentissage à chaque paramètre, comme Adagrad. Cependant, il ne cumule que les gradients issus des itérations récentes. Pour cela, il utilise une moyenne glissante :

$$(\theta_{t+1})_i = (\theta_t)_i - \alpha \frac{(\nabla J(\theta_t))_i}{\sqrt{(\nabla_{t+1})_i}} \quad (3.20)$$

$$(\nabla_{t+1})_i = \delta (\nabla_t)_i + (1 - \delta) (\nabla J(\theta_t))_i^2$$

$\delta (\nabla_t)_i$ est la moyenne quadratique glissante du gradient. La division du gradient de la fonction objective par la racine de la moyenne quadratique glissante (amplitude) améliore la convergence. En le comparant à AdaGrad, l'utilisation de la

moyenne mobile introduit un nouvel hyperparamètre δ , qui contrôle l'échelle de longueur de la moyenne mobile.

Empiriquement, RMSProp s'est avéré être un algorithme d'optimisation efficace et pratique pour les réseaux de neurones profonds.

6.7.3. Estimation adaptative de moment

L'apprentissage par estimation adaptative de moment (Adam) représente les dernières tendances en matière d'optimisation de l'apprentissage profond (Kingma, 2015). Adam est une stratégie d'apprentissage qui a été conçue spécifiquement pour entraîner des réseaux neuronaux profonds. Plus de mémoire efficace et moins de puissance de calcul sont deux avantages d'Adam.

Adam peut être considéré comme une combinaison de RMSprop et descente de gradient stochastique avec Momentum. Il adapte automatiquement le taux d'apprentissage pour chaque paramètre. Sa particularité est de calculer (m_t, v_t) des "estimations adaptatives des moments".

Il peut donc être vu comme une généralisation de l'algorithme Adagrad :

$$\begin{aligned}
 (m_{t+1})_i &= \beta_1(m_t)_i + (1 - \beta_1)(\nabla J(\theta_t))_i \\
 (v_{t+1})_i &= \beta_2(v_t)_i + (1 - \beta_2)(\nabla J(\theta_t))_i^2 \\
 (\theta_{t+1})_i &= (\theta_t)_i - \alpha \frac{\sqrt{1 - \beta_2}}{1 - \beta_1} \frac{(m_t)_i}{\sqrt{(v_t)_i} + \epsilon}
 \end{aligned} \tag{3.21}$$

$$\alpha, \epsilon > 0 \text{ et } \beta_1, \beta_2 \in]0, 1[$$

m_t est le premier moment du gradient (la moyenne) et v_t est son second moment (variance non-centrée). ϵ est un paramètre de précision (peut prendre la valeur 10^{-8}). Les paramètres β_1 et β_2 sont utilisés pour réaliser des moyennes d'exécution sur les moments m_t et v_t respectivement.

6.8. Choisir le bon algorithme d'optimisation

Dans cette Section, nous avons discuté d'une série d'algorithmes connexes qui cherchent chacun à relever le défi de l'optimisation des modèles profonds en adaptant le taux d'apprentissage pour chaque paramètre du modèle. À ce stade, une question naturelle est : quel algorithme choisir ?

Malheureusement, il n'y a pas actuellement de consensus sur ce point. Schaul et al. ont présenté une comparaison intéressante d'un grand nombre d'algorithmes

d'optimisation à travers un large éventail de tâches d'apprentissage (Schaul et al., 2014). Alors que les résultats suggèrent que la famille d'algorithmes avec des taux d'apprentissage adaptatifs (représentée par RMSProp et Adagrad) a fonctionné de manière assez robuste, aucun meilleur algorithme n'a émergé.

Actuellement, les algorithmes d'optimisation les plus populaires activement utilisés incluent SGD, SGD avec momentum, RMSProp, RMSProp avec momentum et Adam. Le choix de l'algorithme à utiliser, à ce stade, semble dépendre en grande partie de la familiarité de l'utilisateur avec l'algorithme (pour faciliter le réglage des hyperparamètres).

Pour conclure cette section, la Figure suivante montre une comparaison d'Adam avec d'autres algorithmes d'optimisation entraînant un perceptron multicouche sur la base MNIST.

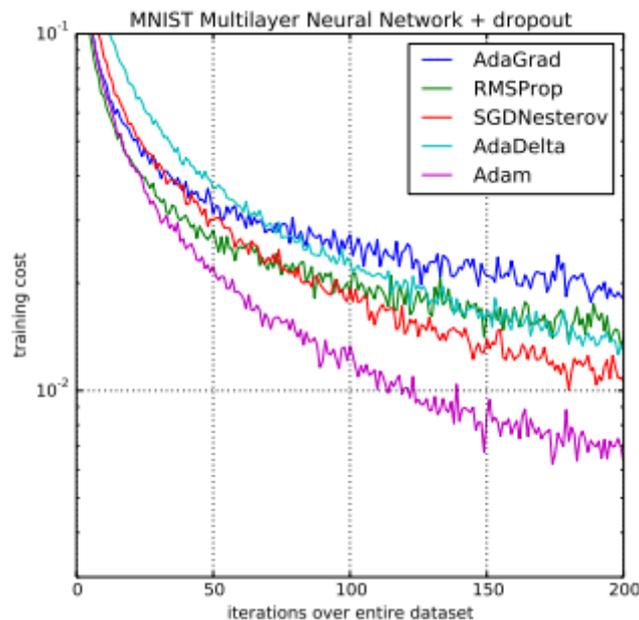


Figure 3.13 : Comparaison Adam avec d'autres optimiseurs (Kingma, 2015).

6.9. Apprentissage par transfert learning

Nous pouvons exploiter le réseau de neurones convolutionnel pré-entraîné par le processus de transfert learning (Chapitre 2, Section 6) de plusieurs façons, en fonction de la taille du jeu de données en entrée et de sa similarité avec celui utilisé lors du pré-entraînement. Plusieurs stratégies peuvent être définies :

Stratégie 1 : fine-tuning total

On remplace la dernière couche fully-connected du réseau pré-entraîné par un classifieur adapté au nouveau problème (SVM, régression logistique...) et initialisé de manière aléatoire. Toutes les couches sont ensuite entraînées sur les nouvelles données.

Cette stratégie doit être utilisée lorsque la nouvelle collection de données est grande : dans ce cas, on peut se permettre d'entraîner tout le réseau sans courir le risque de sur-apprentissage. De plus, comme les paramètres de toutes les couches (sauf de la dernière) sont initialement ceux du réseau pré-entraîné, la phase d'apprentissage sera faite plus rapidement que si l'initialisation avait été aléatoire.

Stratégie 2 : extraction des caractéristiques

Cette stratégie consiste à se servir des caractéristiques du réseau pré-entraîné pour représenter les données du nouveau problème. Pour cela, on retire la dernière couche fully-connected et on fixe tous les autres paramètres. Ce réseau tronqué va ainsi calculer la représentation de chaque donnée en entrée à partir des caractéristiques déjà apprises lors du pré-entraînement. On entraîne alors un classifieur, initialisé aléatoirement, sur ces représentations pour résoudre le nouveau problème.

Cette stratégie doit être utilisée lorsque la nouvelle collection de données est petite et similaire aux données de pré-entraînement. En effet, entraîner le réseau sur aussi peu de données est dangereux puisque le risque de sur-apprentissage est important. De plus, si les nouvelles données ressemblent aux anciennes, elles peuvent alors être représentées par les mêmes caractéristiques.

Stratégie 3 : fine-tuning partiel

Il s'agit d'un mélange des stratégies 1 et 2 : on remplace à nouveau la dernière couche fully-connected par le nouveau classifieur initialisé aléatoirement, et on fixe les paramètres de certaines couches du réseau pré-entraîné. Ainsi, en plus du classifieur, on entraîne sur les nouvelles données les couches non-fixées, qui correspondent en général aux plus hautes du réseau.

On utilise cette stratégie lorsque la nouvelle collection de données est petite mais très différente des données du pré-entraînement. D'une part, comme il y a peu de données d'entraînement, la stratégie 1 qui consiste à entraîner tout le réseau n'est pas envisageable à cause du risque de sur-apprentissage.

D'autre part, on élimine également la stratégie 2 puisque les nouvelles données ont très peu de points communs avec les anciennes : utiliser les caractéristiques

du réseau pré-entraîné pour les représenter n'est pas une bonne idée. Ainsi, la stratégie de fixer les couches basses et d'entraîner le classifieur et les couches hautes constitue un bon compromis.

7. Conclusion

Ce chapitre a fait preuve d'une brève présentation des réseaux de neurones artificiels (RNA), puis dans une seconde section les réseaux de neurones convolutifs (CNN) sont exposées d'une manière assez convaincante pour le lecteur afin de se familiariser avec les concepts de l'apprentissage profond. Dans le même contexte la régularisation ainsi que l'apprentissage et l'optimisation des réseaux de neurones convolutifs sont discutées.

Chapitre 4 : Méthodologie et modèles des réseaux de Neurones Convolutionnels

1	Introduction.....	87
2	Prétraitement.....	87
3	Augmentation de données.....	89
4	Extraction des caractéristiques des lésions cutanées.....	90
5	Modèles des réseaux de neurones convolutionnels	93
6	Conclusion.....	109

1. Introduction

Dans ce chapitre nous présentons la méthodologie globale des approches adoptées (Figure 4.1) et qui seront mises en œuvre dans la partie expérimentale.

Nous discutons les réseaux de neurones convolutifs les plus populaires que nous avons utilisés parmi toutes les variantes. L'étude des fonctionnalités de ces architectures est la clé pour choisir l'architecture appropriée à chaque tâche. L'extraction de caractéristiques est une étape cruciale pour obtenir une représentation discriminante des lésions cutanées. Une description globale des différentes techniques a été exposée dans ce chapitre.

Les images dermoscopiques n'ont pas la qualité attendue pour effectuer l'analyse diagnostique. Ainsi, l'étape de prétraitement des images est très importante et nécessaire, afin de réduire le nombre d'anomalies et de bruit des images.

En outre, ce chapitre dresse la problématique d'augmentation des données qui permet de corriger de manière artificielle le nombre d'exemples d'une base de données par application de transformations aux données d'origine

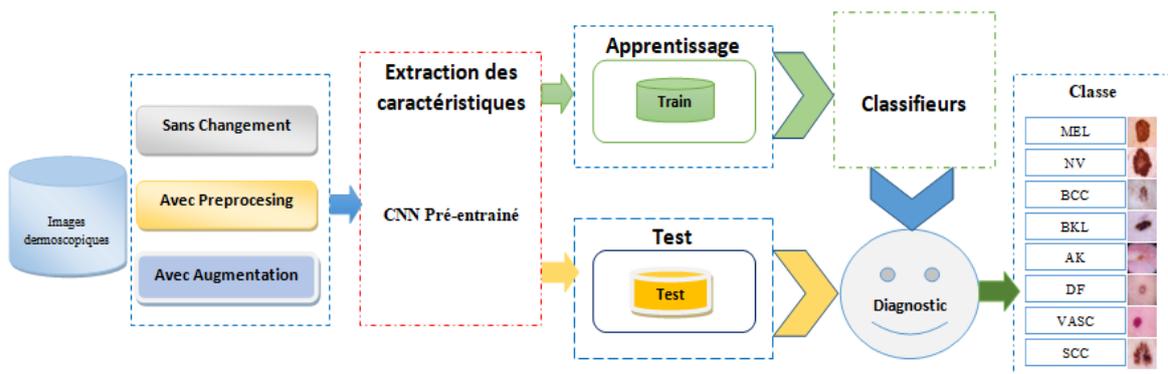


Figure 4.1 : Méthodologie globale des approches adoptées.

2. Prétraitement

Le but du prétraitement d'image est d'améliorer les données d'image en supprimant les distorsions indésirables de l'image ou en améliorant les caractéristiques de l'image. L'étape de prétraitement fait face aux imperfections de l'image introduites lors de l'étape initiale d'acquisition, en éliminant plusieurs artefacts, tels que les cheveux ou les marques de règle. Idéalement, l'image collectée devrait être exempte de ces artefacts.

Typiquement, les images dermoscopiques n'ont pas la qualité attendue pour effectuer l'analyse diagnostique. Ainsi, l'étape de prétraitement des images est très importante et nécessaire, afin de réduire le nombre d'anomalies et de bruit des images. Les cheveux foncés, les lignes de peau, l'huile, les bulles d'air et la réflexion éclaircissante causée par la mise en place de gel ou d'huile avant la capture de l'image dermoscopiquesont présents dans presque toutes les images (Figure 4.2); conduisant ainsi à un mauvais diagnostic de la maladie. Le processus d'amélioration de l'image implique deux opérations clés : les poils, la détection et l'élimination des artefacts de réflexion.

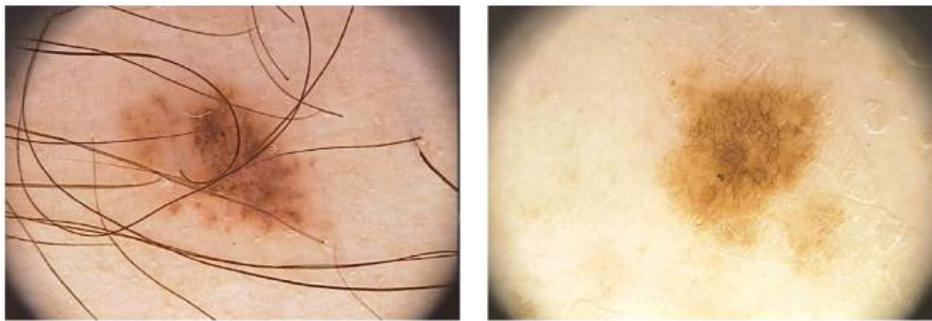


Figure 4.2 : Exemples de deux images dermoscopiques montrant différents types de bruit. A gauche : présence de poils. A droite : présence d'artefacts de gel et de réflexion.

Il existe des algorithmes de suppression d'artefacts courants tels que DullRazor (Kiani & Sharafat, 2011), qui est utilisé pour supprimer les artefacts capillaires (Figure 4.3). L'algorithme utilise des filtres morphologiques pour détecter la position des cheveux et remplacer les cheveux détectés par ses pixels voisins. Le filtre médian, le seuil adaptatif et les opérations morphologiques ont été largement utilisés à cette fin (Capdehourat et al., 2011 ; Fernández et al., 2010 ; Hoshyar et Al-Jumaily, 2014). De plus, pour améliorer encore la qualité de l'image, certaines méthodes d'amélioration de l'image sont également utilisées. Les plus importants d'entre eux sont la correction des couleurs ou l'étalonnage (Wighton et al., 2011). Nous pouvons trouver d'autres techniques telles que la correction de l'éclairage, l'amélioration du contraste et l'amélioration des contours (Malogiannis et al., 2006). Les lecteurs trouveront un résumé des méthodes de prétraitement liées aux images dermoscopiques dans (Celebi et al., 2015).

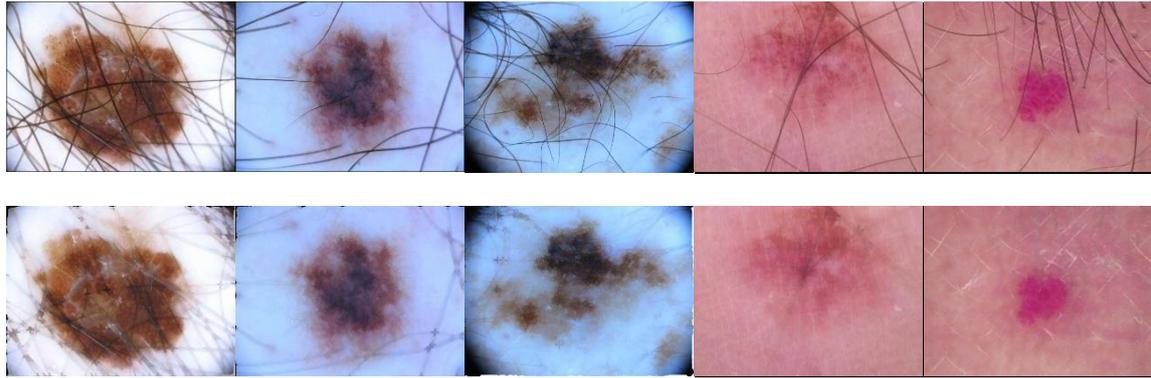


Figure 4.3 : Exemple d'élimination des poils par Dull Razor.

3. Augmentation de données

L'augmentation de données est une technique qui permet de corriger de manière artificielle le nombre d'exemples d'une base de données par application de transformations aux données d'origine (Taylor & Nitschke, 2018).

Pour résoudre le problème des données limitées, différentes méthodes d'augmentation des données sont disponibles. Ces méthodes d'augmentation incluent différentes techniques pour améliorer la taille et la qualité des données afin de faciliter l'apprentissage des CNN (Shorten, C., & Khoshgoftaar, 2019). Les méthodes d'augmentation des données ont amélioré les performances des CNN jusqu'à 13,82 % (Taylor, L., & Nitschke, 2018). De nombreux auteurs pensent que l'augmentation des données est fondamentale pour améliorer les performances des réseaux de neurones convolutifs (Perez, L., & Wang, 2017 ; Howard et al., 2017 ; Krizhevsky et al., 2012).

L'augmentation des données consiste à appliquer certaines transformations à l'ensemble d'apprentissage initial pour créer de nouveaux exemples artificiels. Les transformations appliquées ne changent pas la nature de la classe détectée et créent ainsi de nouveaux exemples. L'augmentation des données permet de créer artificiellement des invariances du réseau final et permet d'augmenter les performances en généralisation. Pour les images de lésions cutanées, parmi les opérations classiques utilisées pour l'augmentation des données (Perez et al., 2018), on peut trouver la rotation, le zoom, le recadrage, l'ajout de bruit, la mise à l'échelle et la translation. Des transformations plus avancées telles que la modification du contraste ou de la luminosité de l'image peuvent également être ajoutées en fonction des besoins de l'utilisateur (Figure 4.4).

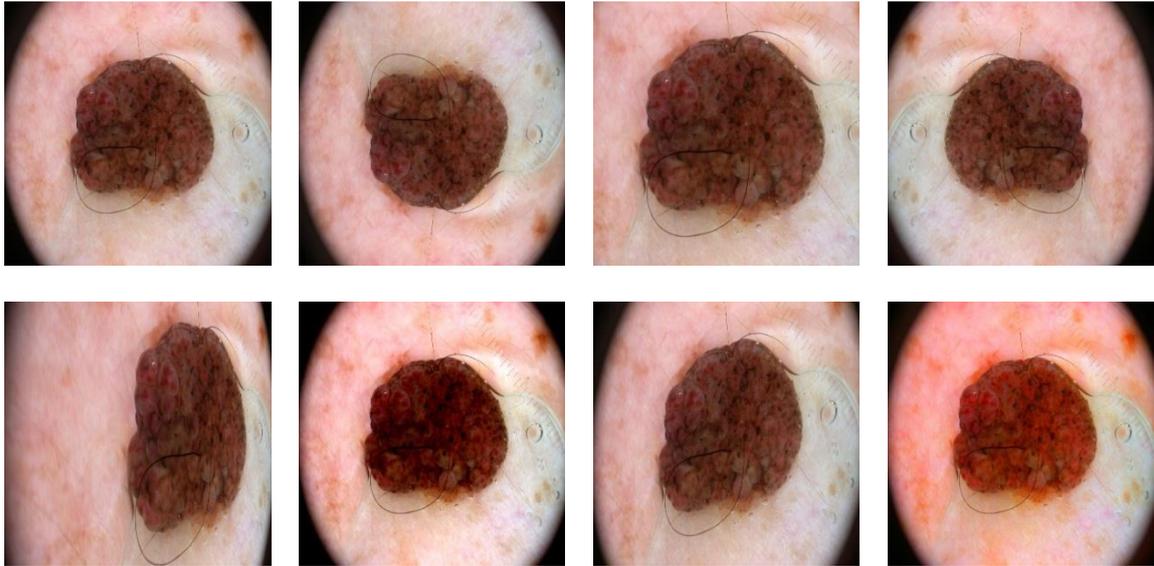


Figure 4.4 : Image originale (première à gauche en haut) et quelques exemples d'augmentation de données.

4. Extraction des caractéristiques des lésions cutanées

L'extraction de caractéristiques est une étape cruciale pour obtenir une représentation discriminante des lésions cutanées. La méthode d'extraction de caractéristiques est utilisée pour obtenir autant d'informations que possible sur l'image dermoscopique (Pathan et al., 2018).

Trouver des caractéristiques appropriées est un processus difficile et de nombreuses recherches ont été effectuées dans ce domaine, permettant d'identifier un large éventail de descripteurs qui caractérisent les lésions cutanées. La sélection et l'efficacité des caractéristiques choisies et l'extraction sont un défi majeur (Barata et al., 2019). De nombreuses méthodes sont utilisées pour extraire les caractéristiques, qui peuvent dépendre des caractéristiques géométriques, des caractéristiques statistiques, des caractéristiques de texture et des caractéristiques de couleur (Mutlag et al., 2020).

Diverses caractéristiques statistiques telles que les caractéristiques : couleur, texture, forme, contour, géométrique, etc., ainsi que les caractéristiques dermoscopiques comme ABCD, le motif (local/global) et la structure dermoscopique sont utilisées pour la détection des lésions cutanées. Mais le principal défi est de savoir comment sélectionner le nombre minimum de caractéristiques et choisir les meilleurs ensembles de caractéristiques qui donnent de bons résultats en termes de performances, de complexité, de précision et de temps de calcul.

Globalement, l'extraction des caractéristiques des lésions sont définis selon le schéma suivant :

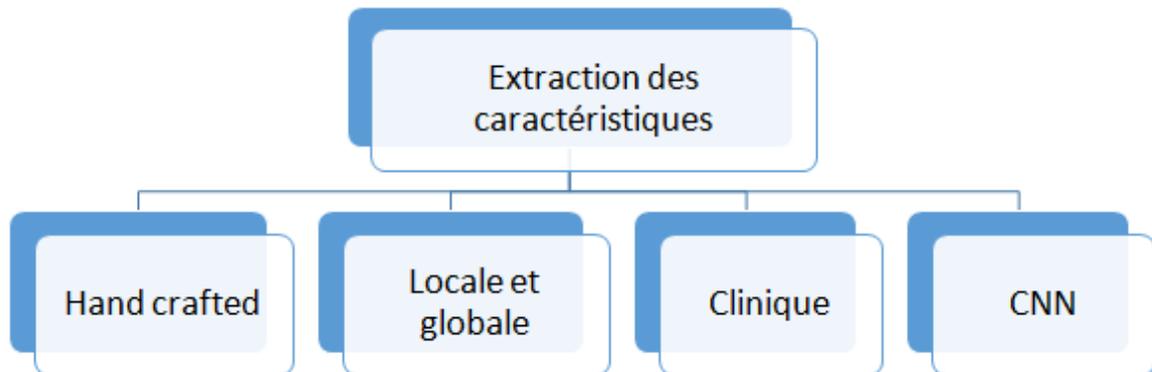


Figure 4.5 : Différentes approches d'extraction des caractéristiques.

4.1. Hand-crafted

Les caractéristiques sont extraites manuellement à partir des images séparées selon un certain algorithme prédéfini basé sur les connaissances de l'expert. La sélection du type de caractéristiques et de descripteurs le plus approprié est spécifique au problème est généralement basée sur les connaissances des praticiens.

Les caractéristiques peuvent être divisées en quatre types : caractéristiques de couleur, caractéristiques de texture, caractéristiques statistiques et caractéristiques géométriques.

- Caractéristiques de couleur : La couleur est utilisée pour identifier les lésions cutanées. Fondamentalement, la variation de la couleur de la peau est le signe précoce permettant de différencier les mélanomes bénins et malins à partir d'images dermoscopiques (Oliveira et al. 2018 ; Li et al., 2018).
- Caractéristiques de texture : La texture est une propriété native de la peau ayant les informations de structure ou d'arrangement de motif de la peau. Dans la littérature, quatre types de méthodes existent pour l'analyse de texture (Filali et al., 2019) telles que :
 - a) les techniques statistiques,
 - b) les techniques basées sur des modèles,
 - c) les techniques structurelles et
 - d) les techniques basées sur la transformation.
- Caractéristiques statistiques : Onze types de caractéristiques sont identifiés : Contraste, Entropie, RMS, Énergie, Aplatissement, Corrélacion, Variance,

Cinquième et sixième moment central, Lissage, Moyenne et écart type (Moldovanu et al., 2021).

- Caractéristiques géométriques : La forme est une caractéristique visuelle très importante pour identifier et reconnaître les lésions cutanées (Ruela et al., 2017). Pour déterminer la forme de la lésion cutanée, principalement six caractéristiques ont été extraites des règles B (bordure) et D (diamètre) à partir desquelles deux caractéristiques, le périmètre et la surface, sont utilisées pour mesurer la taille de la lésion cutanée (Javed et al., 2019). Les autres caractéristiques restantes sont :
 - périmètre à rapport de surface,
 - rapport d'aspect : défini comme la largeur divisée par la hauteur du rectangle minimal qui délimite la région de la lésion,
 - rapport d'étendue : défini comme la surface de la région de la lésion divisée par la surface du rectangle de délimitation minimal et
 - rapport de solidité : défini comme la surface de la lésion cutanée divisée par la surface de la coque convexe de la limite de la lésion.

4.2. Caractéristique locale et globale

L'idée de base des caractéristiques locales est qu'une lésion cutanée peut être représentée comme une collection d'éléments d'un dictionnaire de mots visuels. Ce dictionnaire est généralement appris de manière non supervisée. La méthode la plus simple et la plus populaire pour l'extraction de caractéristiques locales est appelée sac de caractéristiques (BoF).

Grâce à des caractéristiques locales et globales, l'analyse des modèles identifie les lésions mélanocytaires (Oliveira et al. 2018). Les caractéristiques locales contiennent la formation dermoscopique (réseau pigmentaire, points, gouttes, etc.) et les caractéristiques globales fournissent l'ensemble des caractéristiques d'évaluation constituées de motifs multi-composants : motifs réticulaires, globulaires, homogènes, parallèles et motifs non spécifiques (Mughal et al., 2017 ; Husham et al., 2016).

4.3. Caractéristiques cliniques

Différentes caractéristiques cliniques sont utilisées pour la détection du mélanome à partir des images dermoscopiques ([Chapitre 1, Section 3.6.3](#)).

- Règle ABCD ([Nachbar et al., 1994](#) ; [Li et al., 2018](#))
- Liste des 7 points à vérifier ([Cueva et al., 2017](#))
- Notation de Menzies ([Javed et al., 2019](#))

4.4. Approche CNN

Il est indéniable que le deep learning, et les réseaux de neurones convolutionnels (CNN) en particulier, sont devenus la technique de choix dans de nombreux problèmes de vision par ordinateur ([Guo et al., 2016](#)). La communauté de l'imagerie médicale a également adopté cette technique, avec un nombre toujours croissant d'applications qui utilisent cette méthodologie pour diagnostiquer ou segmenter des organes et des structures dans des images médicales ([Litjens et al., 2017](#)).

Le premier article sur l'apprentissage profond appliqué aux images dermoscopiques a été publié par Codella et al. ([Codella et al., 2015](#)) en 2015, et depuis lors, plusieurs auteurs ont exploré l'apprentissage en profondeur pour apprendre des représentations d'images appropriées et obtenir des performances de diagnostic d'experts (quasi) humains.

5. Modèles des réseaux de neurones convolutionnels

Au fil des ans, les réseaux de neurones convolutifs (CNN) ont subi une quantité considérable de restructurations et d'améliorations ([Figure 4.6](#)). Ces modifications comprennent la reformulation structurelle, la régularisation, l'optimisation des paramètres, etc ([Khan et al., 2020](#)). Il convient de noter que la mise à niveau clé des performances de CNN est due en grande partie à la réorganisation de l'unité de traitement, ainsi qu'au développement de nouveaux blocs. En particulier, les développements les plus novateurs des architectures CNN ont été effectués sur l'utilisation de la profondeur du réseau. Les architectures CNN, commençant par le modèle LeNet en 1998 et se terminant par efficient Net en 2019.

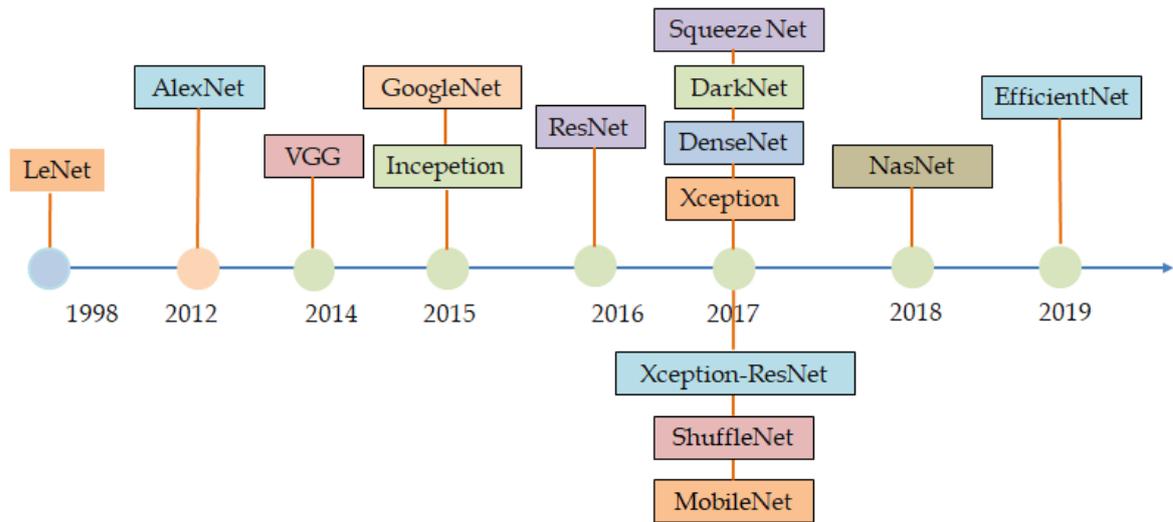


Figure 4.6 : Evolution des réseaux de neurones convolutionnels.

5.1. LeNet-5

LeNet-5 est une architecture proposée par Yann LeCun en 1998 (LeCun et al., 1998). Il est célèbre en raison de son importance historique car il s'agissait du premier modèle standard pour tous les CNN modernes, il a démontré des performances avancées sur les tâches de reconnaissance des chiffres manuscrits. La Figure 4.7 illustre la structure de l'architecture LeNet.

LeNet se compose de cinq couches alternées de trois couches de convolution et deux couches d'average pooling, suivies d'une couche convolutive d'aplatissement, puis de deux couches entièrement connectées et enfin d'un classificateur softmax. LeNet-5 a environ de 60 mille paramètres. Dans cette architecture, la taille des filtres dans les couches de convolution a été fixée à (5×5) tandis que (2×2) dans les couches average pooling.

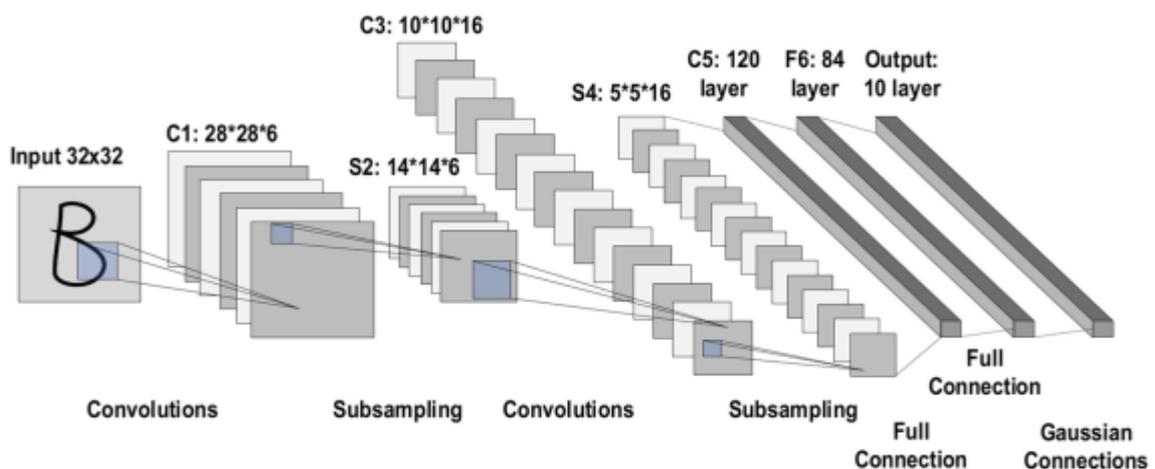


Figure 4.7 : Structure du réseau LeNet.

5.2. AlexNet

AlexNet est une architecture développée en 2012 par Alex Krizhevsky (Krizhevsky et al., 2012), est considéré comme la première architecture CNN profonde, qui a amélioré la capacité d'apprentissage du CNN en augmentant sa profondeur avec plusieurs stratégies d'optimisation des paramètres par rapport à LeNet qui était limité aux tâches de reconnaissance des chiffres manuscrits. La Figure 4.8 illustre la structure de l'architecture d'AlexNet.

AlexNet a environ 60 millions de paramètres, composée de huit couches : cinq couches conventionnelles et trois max pooling suivis de trois couches entièrement connectées. D'autres ajustements ont été effectués par rapport aux réseaux précédemment proposés telle que l'utilisation de filtres de grande taille (11×11 et 5×5) et l'utilisation des unités linéaires rectifiées (Relu) comme fonction d'activation non saturante et des entrées de taille 227 × 227.

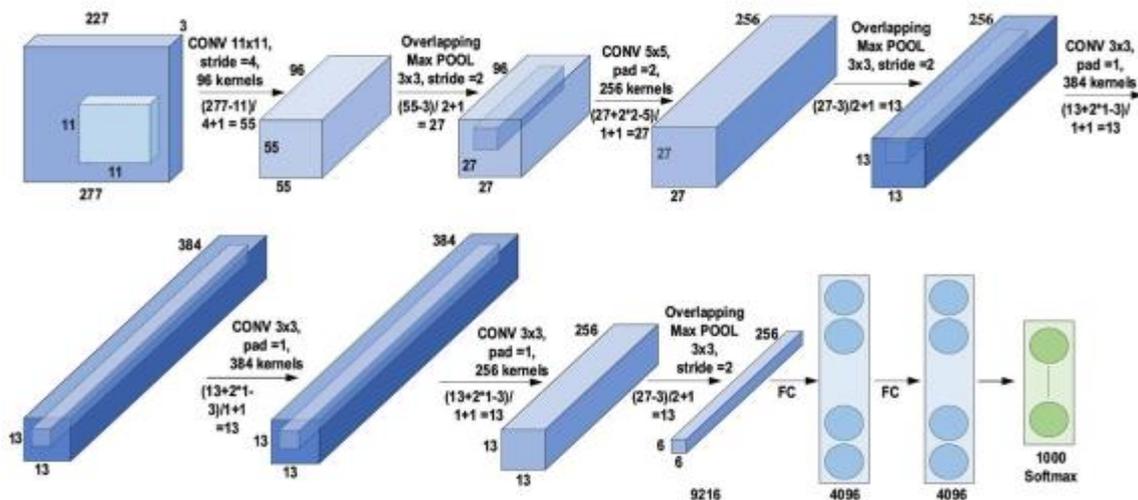


Figure 4.8 : Structure du réseau AlexNet.

VGGNet

VGGNet est une architecture multicouche proposée par Simonyan et Zisserman en 2014 (Simonyan & Zisserman, 2015). Estimant que la meilleure façon d'améliorer l'efficacité d'un CNN était d'y empiler plus de couches. La Figure 4.9 illustre la structure de l'architecture VGGNet. Les concepteurs ont développé VGG-16 avec 13 couches convolutionnelles et 3 couches entièrement connectées et une autre version plus profonde VGG-19 avec 16 couches convolutionnelles et 3 couches entièrement connectées. Les concepteurs ont suggéré que des filtres de petite taille peuvent améliorer les performances des CNN, sur la base de ces résultats, VGG a remplacé les filtres (11×11) et (5×5) par une pile de couches de filtres (3×3). Cela a démontré

expérimentalement que l'affectation parallèle de filtres de petite taille (3×3) pouvait produire la même influence que les filtres de grande taille (5×5 et 7×7). En d'autres termes, l'utilisation de filtres de petite taille offre un avantage supplémentaire de réduction de calcul en diminuant le nombre de paramètres. VGG était à la 2e place du concours ILSVRC 2014⁵ mais s'est fait connaître en raison de sa simplicité, de sa topologie homogène et de sa profondeur accrue.

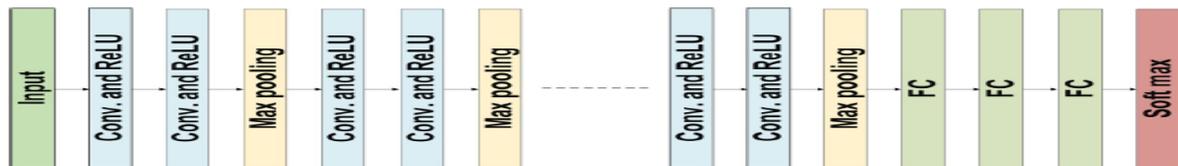


Figure 4.9 : Structure du réseau VGGNet. (Bezdan et al ,2019).

5.4. Google Net

GoogleNet (également appelé Inception-V1) est le gagnant du concours ILSVRC 2014⁶ avec un taux d'erreur de 6.67% (Ioffe & Szegedy, 2015). GoogleNet propose des blocs de démarrage (modules Inception) comme un nouveau concept dans le contexte CNN. Les modules présentent des variantes optimisées des couches de convolution classiques avec différentes échelles pour capturer des informations spatiales à des niveaux différents. Les modules de démarrage introduits des connexions partielles à l'intérieur d'une couche convolutive pour réduire sa dimensionnalité. La Figure 4.10 illustre l'architecture d'un bloc de démarrage. Ces modules utilisent des tailles variables (1×1, 3×3 et 5× 5), qui sont appliqués sur la même couche convolutive, puis les résultats des cartes de caractéristiques sont empilées pour en obtenir une plus représentative et former la couche convolutive suivante.

⁵<https://www.image-net.org/challenges/LSVRC/2014/>

⁶<https://www.image-net.org/challenges/LSVRC/2014/>

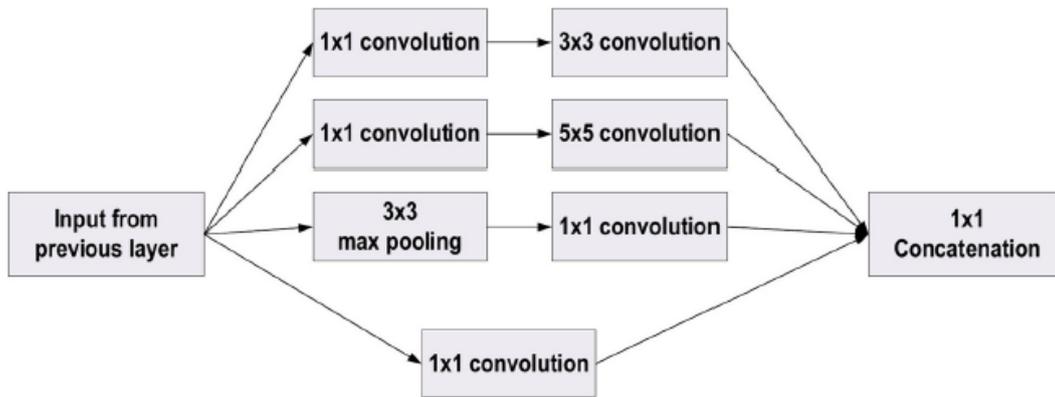


Figure 4.10 : Structure d'un bloc de démarrage

GoogLeNet présente une large architecture avec 22 couches au total et sept millions de paramètres avec neuf modules de démarrage, quatre couches convolutives, quatre couches max-pooling, trois couches average pooling, cinq couches entièrement connectées et trois couches softmax. La [Figure 4.11](#) illustre l'architecture globale de GoogLeNet.

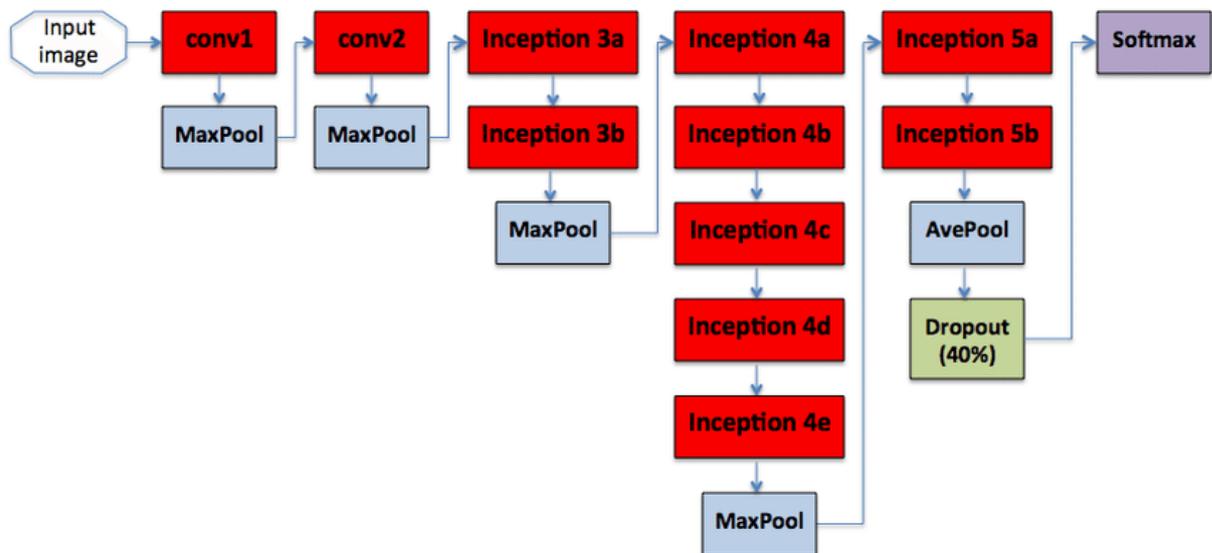


Figure 4.11 : Structure globale de GoogLeNet.

5.5. InceptionV2

Inception V2 est la deuxième génération d'architectures de réseaux de neurones convolutifs Inception qui utilise notamment la normalisation par lots ([Szegedy et al., 2016](#)). InceptionV2 utilise deux techniques de factorisation pour optimiser le cout de calcul. La première technique est la factorisation des noyaux de convolution

par le remplacement des couches convolutives de taille (5×5) par deux couches convolutives de taille (3×3). La deuxième technique est la factorisation spatiale en convolution asymétrique. L'architecture propose de remplacer les couches convolutives (n×n) par des couches convolutionnelles asymétriques (n×1 et 1×n). La **Figure 4.12** illustre l'architecture d'un bloc InceptionV2.

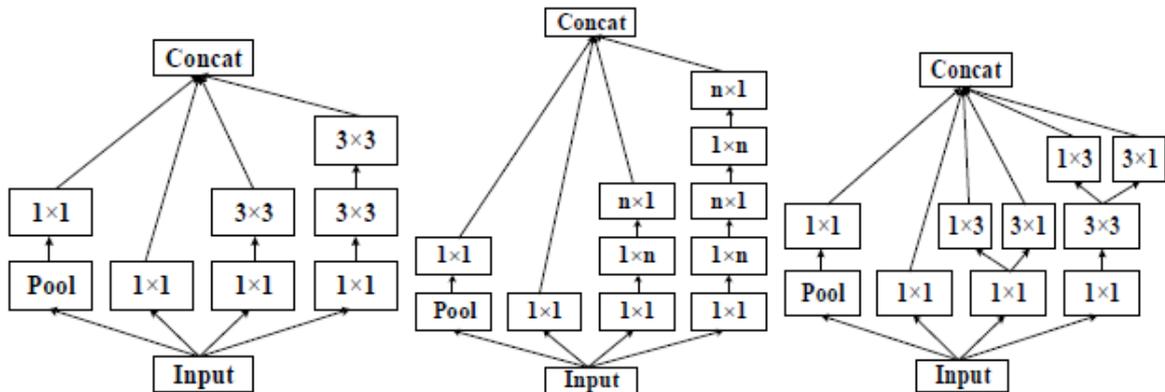


Figure 4.12 : Structure de base du module InceptionV2.

5.6. InceptionV3

InceptionV3 est une version optimisée du réseau Inception-V2 (Szegedy et al., 2016) avec 24 millions de paramètres. L'idée d'inception-V3 était de réduire le coût de calcul des réseaux profonds sans affecter la généralisation, avec des ajustements dans des blocs de construction symétriques et asymétriques. La **Figure 4.13** illustre la structure de l'architecture d'InceptionV3.

InceptionV3 a incorporé toutes les mises à niveau et les méthodes de factorisation d'InceptionV2, elle a également ajouté la factorisation pour les filtres (7×7), optimiseur RMSProp, et la taille d'entrée du réseau est passée de 224×224 à 299×299.

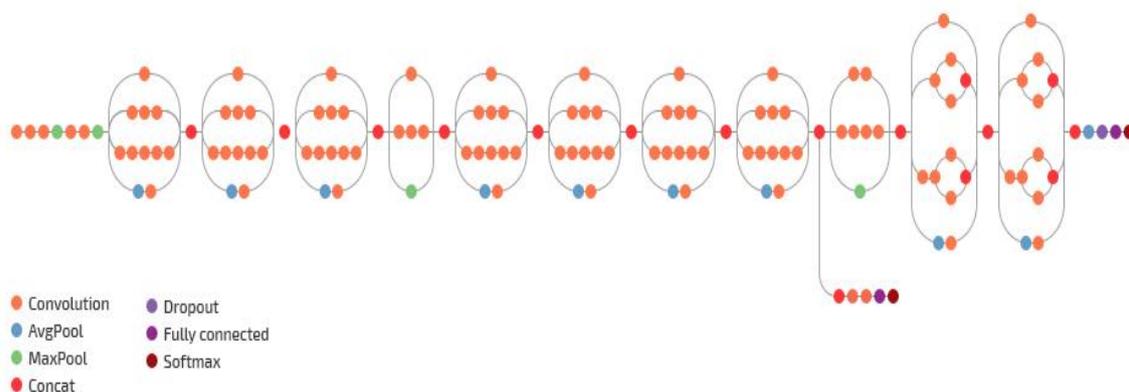


Figure 4.13 : Structure de l'architecture Inception V3.

5.7. ResNet

ResNet proposé par He en 2015 (He et al., 2016) a introduit le concept des blocs résiduels dans les CNN. Leur objectif était de concevoir un réseau profond en limitant la perte de gradient dans les couches les plus profondes de celui-ci par l'ajout d'une liaison résiduelle entre chaque couche de convolution (Figure 4.14).

Les avantages de ces réseaux sont leur capacité à réduire la fuite de gradient dans les couches inférieures du réseau, et la possibilité de leur mise à l'échelle en profondeur. Cette dernière permet l'ajustement de l'architecture à la tâche de classification à accomplir. Bien que coûteux en ressources de calcul, ils obtiennent de très bonnes performances en classification sur tous les niveaux de détails de l'image.

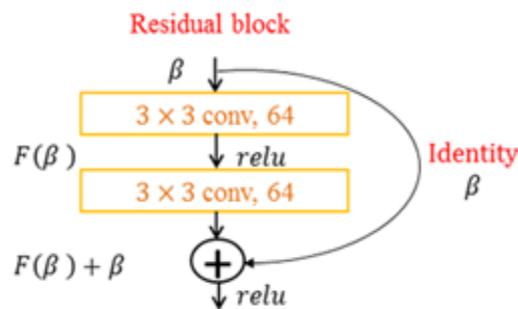


Figure 4.14 : Concept des blocs résiduels

Plusieurs types de ResNet ont été développés en fonction du nombre de couches (ResNet18, ResNet34, ResNet50, ResNet101, ResNet152). Le type le plus courant était ResNet50, qui comprenait 49 couches convolutionnelles plus une seule couche FC.

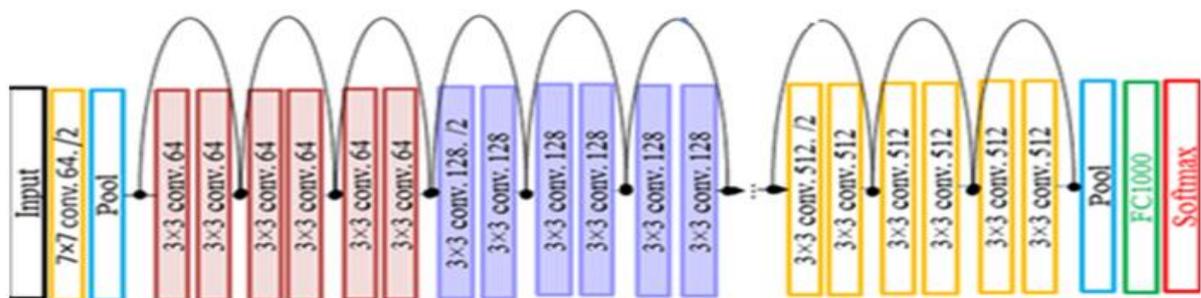


Figure 4.15 : Structure globale de l'architecture ResNet.

5.8. SqueezeNet

Squeeze and Expand Network (SE-Network) est une architecture proposée par Iandola en 2017 (Iandola et al, 2017). Elle propose une technique de réduction de dimensionnalité en ajoutant un nouveau bloc SE avant la couche de convolution pour sélectionner les cartes de caractéristiques. Le bloc SE (Fire Module) est basé sur deux opérations : squeeze et expand (Figure 4.16).

- Squeeze : à partir de la carte de caractéristiques de sortie de la couche de convolution, une transformation de mise en commun par moyenne globale (Global average Pool) des caractéristiques est faite. De façon imagée, les largeurs et longueurs initiales de la carte de caractéristiques sont écrasées pour obtenir des largeurs et longueurs égales à 1. Un vecteur condensé des caractéristiques de taille (1×1×nombre de canaux de la carte de caractéristiques) est obtenu.
- Expand : à partir de ce vecteur, une transformation de type couche “entièrement connectée” est calculée. Pour chaque canal un poids est associé, puis activé par une fonction non-linéaire, afin d’obtenir la valeur de la probabilité de contribution du canal à la tâche du réseau.

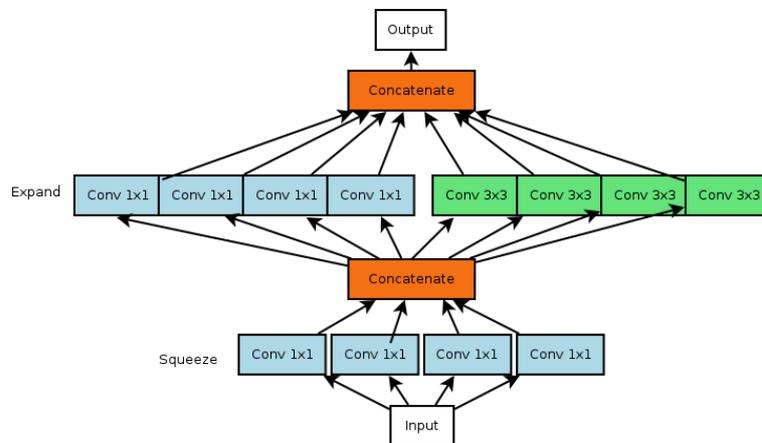


Figure 4.16 : Structure du bloc SE (Fire Module).

SqueezeNet est composé de deux convolutions standards, huit modules Fire, trois couches max-pooling, et une couche average pooling, suivie à la fin d'une couche softmax. La Figure 4.17 illustre la structure de l'architecture SqueezeNet.

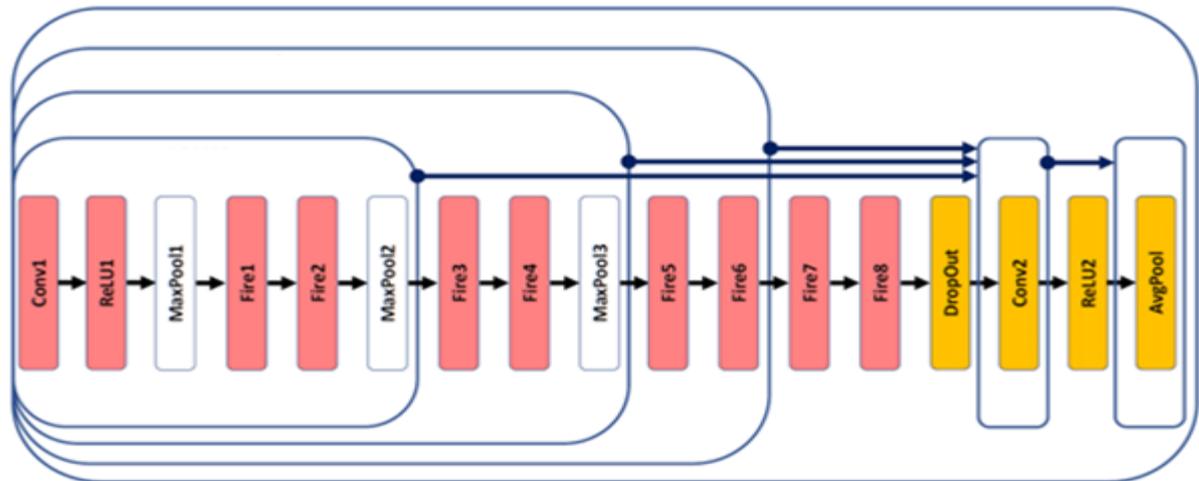


Figure 4.17 : Structure de l'architecture SqueezeNet.

5.9. DarkNet

DarkNet est un réseau de neurone convolutionnel rapide et précis avec différentes versions : Darknet-19 (Redmon & Farhadi, 2017) DarkNet-53 (Redmon & Farhadi, 2018). Le DarkNet-19 se compose de 19 couches convolutionnelles, 5 couches de max-pooling, average pooling et softmax. Il utilise le mappage d'identité et des filtres de convolution 3×3 au lieu de couches entièrement connectées, un pooling moyen global pour faire des prédictions en amont de la couche softmax. La normalisation par lots est utilisée pour régulariser le modèle, accélérer la convergence et stabiliser l'apprentissage sans utiliser de blocs résiduels, d'échantillonnage ou de sauts de connexion. Le DarkNet-53 se compose de la version améliorée de DarkNet-19 avec plus de couches et l'utilisation de connexions résiduelles. Il est composé de 53 couches avec des filtres de convolution 3×3 et 1×1 , cinq blocs résiduels, des couches average pooling, et une couche entièrement connectée. DarkNet-53 n'utilise aucune couche de pooling maximale ; à la place, des couches de convolution sont utilisées. DarkNet-53 permet la détection à trois échelles en réduisant les dimensions l'entrée par 32, 16 et 8.

Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

Type	Filters	Size	Output
Convolutional	32	3 × 3	256 × 256
Convolutional	64	3 × 3 / 2	128 × 128
Convolutional	32	1 × 1	
Convolutional	64	3 × 3	
Residual			128 × 128
Convolutional	128	3 × 3 / 2	64 × 64
Convolutional	64	1 × 1	
Convolutional	128	3 × 3	
Residual			64 × 64
Convolutional	256	3 × 3 / 2	32 × 32
Convolutional	128	1 × 1	
Convolutional	256	3 × 3	
Residual			32 × 32
Convolutional	512	3 × 3 / 2	16 × 16
Convolutional	256	1 × 1	
Convolutional	512	3 × 3	
Residual			16 × 16
Convolutional	1024	3 × 3 / 2	8 × 8
Convolutional	512	1 × 1	
Convolutional	1024	3 × 3	
Residual			8 × 8
Avgpool		Global	
Connected		1000	
Softmax			

Figure 4.18 : Structure de l'architecture DarkNet 19 et DarkNet 53.

5.10. DenseNet

DenseNet est une architecture proposée par Huang et al en 2017 (Huang et al., 2017). Elle est basé sur des connexions denses entre les couches de convolution et vise à résoudre le problème de la disparition du gradient et réduit le nombre d'entrées et de paramètres ainsi que renforcer l'extraction des caractéristiques.

L'architecture DenseNet exploite les effets des raccourcies de connexions, elle est composé d'un ensemble de blocs denses, qui sont reliés par des couches de transition (Figure 4.19). Chaque bloc contient un ensemble de couches de convolution, et chaque couche est connectée à toutes les autres couches précédentes appartenant au même bloc au lieu de les ajouter. Toutes les cartes de caractéristiques des couches précédentes sont transmises aux couches successives, ce qui rend le réseau plus mince et plus compact. DenseNet comprend des couches de convolution et des couches pooling, des couches de transition, une couche de classification et plusieurs blocs denses en série. DenseNet propose plusieurs

variantes avec un nombre différent de couches : DenseNet-121, DenseNet-169, DenseNet-20 et DenseNet-246.

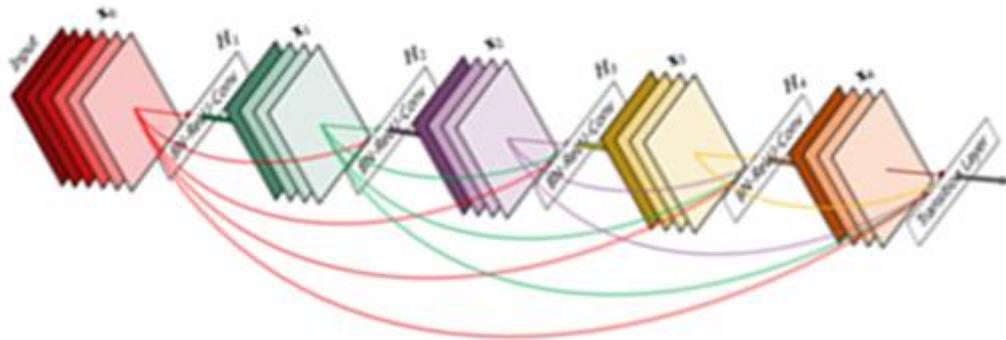


Figure 4.19 : Structure globale de l'architecture Densenet.

5.11. Xception

Xception est une architecture proposée par Google en 2017 (Chollet, 2017) qui implique des convolutions séparables en profondeur. Google a présenté une interprétation des modules Inception dans les réseaux de neurones convolutifs comme étant une étape intermédiaire entre la convolution régulière et l'opération de convolution séparable en profondeur. Xception repose sur deux points principaux :

- Convolution séparable en profondeur : est une convolution en profondeur suivie d'une convolution ponctuelle (Figure 4.20).
- Raccourcis entre les blocs de convolution

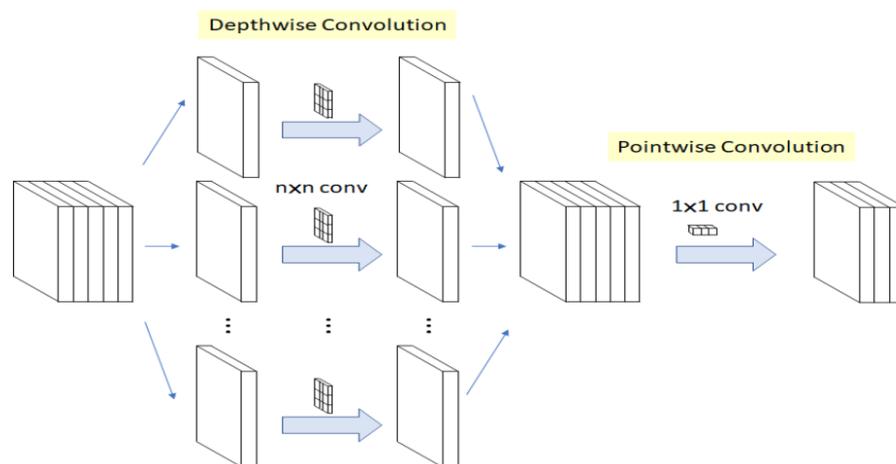


Figure 4.20 : Structure d'une convolution séparable en profondeur.

L'architecture Xception se compose de 36 couches de convolution structurées en 14 modules basés sur l'adaptation des blocs de démarrage en les remplaçant par des couches de convolutions séparables en profondeur.

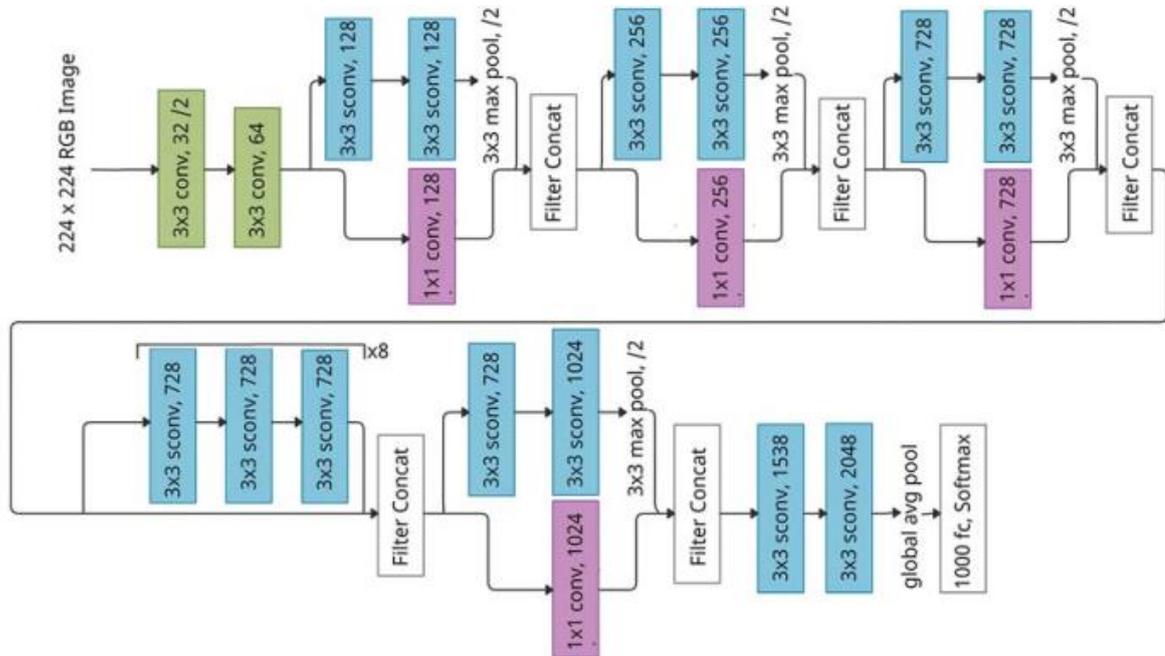


Figure 4.21 : Structure de l'architecture Xception.

5.12. Inception-ResNet

Inception-ResNet est proposé par Szegedy en 2017 (Szegedy et al., 2017) avec 56 millions de paramètres. Inspiré des performances du module Inception et ResNet, il propose une hybridation entre les modules introduits dans les architectures Inception V3/Inception V4 et les connexions résiduelles. Il consiste à remplacer les modules Inception avec les blocs Residual-Inception (Figure 4.22). En ajoutant la sortie de la convolution dans les couches de début à l'entrée, pour obtenir la même dimensionnalité de l'entrée et de la sortie, ils utilisent des convolutions 1x1 après les convolutions d'origine. Cette combinaison vise à accélérer le temps d'apprentissage du réseau Inception et à éviter le problème de la dégradation du gradient.

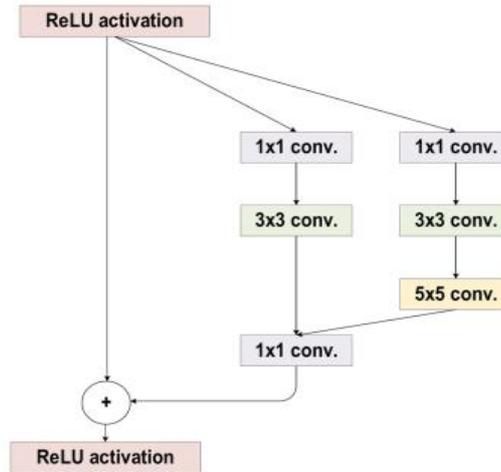


Figure 4.22 : Structure d'un bloc Inception-ResNet.

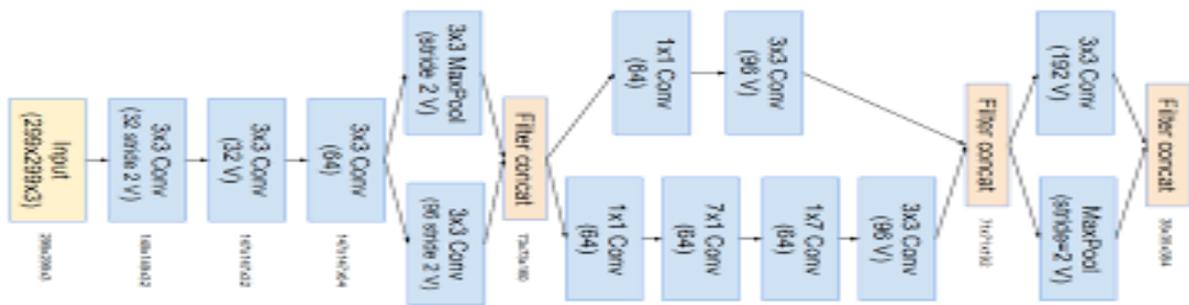


Figure 4.23 : Structure de l'architecture Inception-Resnet.

5.13. ShuffleNet

ShuffleNet est un réseau neuronal convolutif introduit par Zhang en 2017 (Zhang et al., 2018). ShuffleNet a fait avancer la technologie de pointe pour les architectures mobiles CNN en présentant les convolutions de groupe par point (pointwise group convolutions) et le mélange de canaux (channel shuffle). Les convolutions de groupe par point sont utilisées pour accélérer les convolutions 1x1 communément utilisées dans les architectures mobiles CNN. Les convolutions de groupe remplacent la première et seconde convolution 1x1, puis un mélange de canaux est appliqué après la première convolution 1x1.

ShuffleNetV2 propose de nombreux guides pratiques pour des architectures CNN efficaces. Il optimise davantage l'architecture avec des changements dans le goulot d'étranglement et en présentant la division de canaux (channel split). Dans la Figure 4.24, on peut voir les différences entre ShuffleNetV1 (a., b.) et ShuffleNetV2 (c. d.).

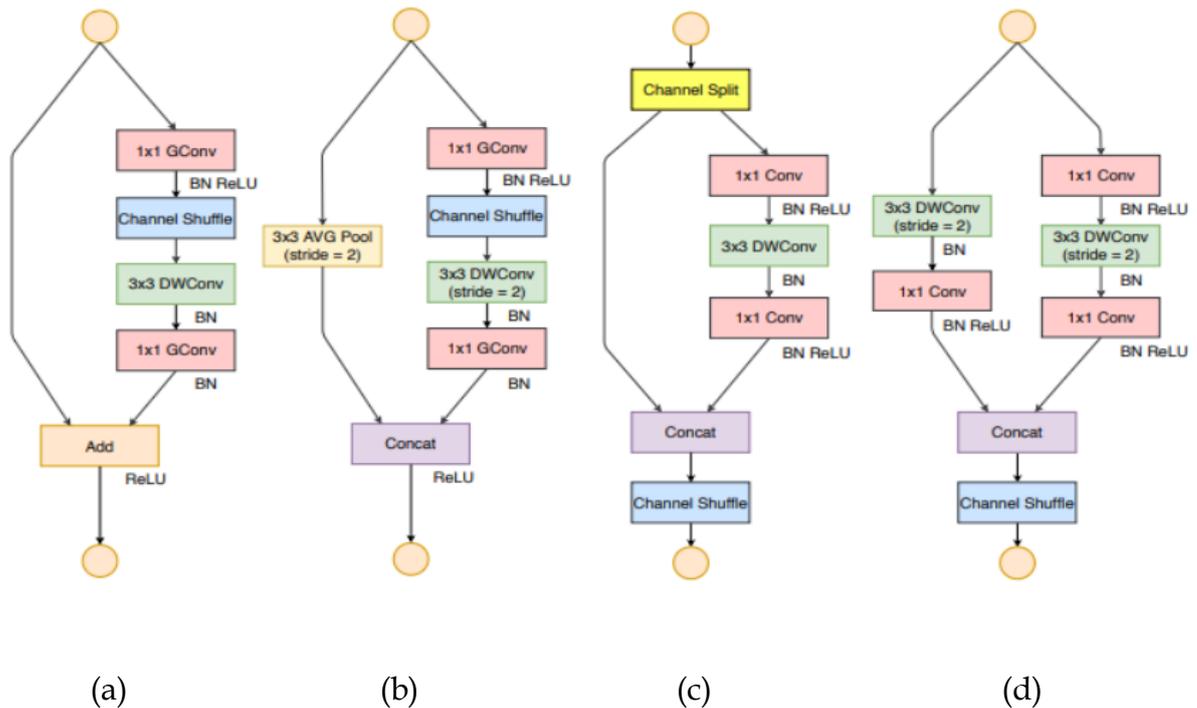


Figure 4.24 : Différence entre ShuffleNetV1 (a., b.) et ShuffleNetV2 (c. d.).

Layer	Output size	KSize	Stride	Repeat	Output channels (g groups)				
					$g = 1$	$g = 2$	$g = 3$	$g = 4$	$g = 8$
Image	224×224				3	3	3	3	3
Conv1	112×112	3×3	2	1	24	24	24	24	24
MaxPool	56×56	3×3	2						
Stage2	28×28		2	1	144	200	240	272	384
	28×28		1	3	144	200	240	272	384
Stage3	14×14		2	1	288	400	480	544	768
	14×14		1	7	288	400	480	544	768
Stage4	7×7		2	1	576	800	960	1088	1536
	7×7		1	3	576	800	960	1088	1536
GlobalPool	1×1	7×7							
FC					1000	1000	1000	1000	1000

Figure 4.25 : Structure de l'architecture ShuffleNet.

5.14. MobileNet

MobileNet a été proposé par Howard en 2017 (Howard et al., 2017). L'architecture MobileNet utilise les convolutions séparables en profondeur. Ces convolutions sont une forme de convolutions factorisées, où une convolution standard est factorisée en une convolution en profondeur et une convolution 1×1 appelée convolution ponctuelle. Cette factorisation réduit considérablement le temps de calcul et la taille du modèle. A ces convolutions, deux hyper-paramètres globaux simples qui font

un compromis efficace entre latence et précision ont été ajouté : le multiplicateur de largeur pour un modèle plus fin, le multiplicateur de résolution pour une représentation réduite.

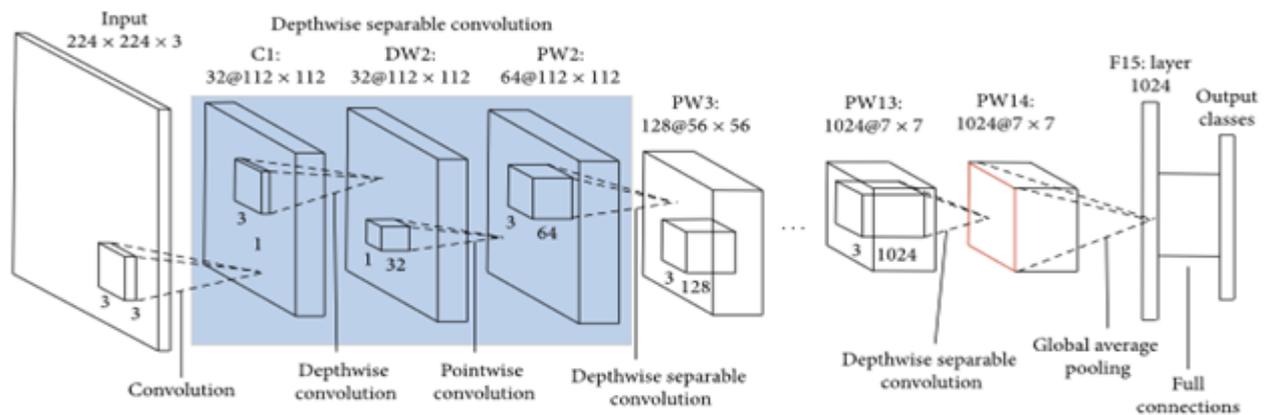


Figure 4.26 : Structure de l'architecture MobileNet.

5.15. Nasnet

Nasnet est une architecture CNN proposée par le groupe Google ML (Zoph et al., 2018). Il introduit le concept de réseau optimisé en utilisant l'apprentissage par renforcement. Fondamentalement, l'idée était de rechercher la meilleure combinaison de paramètres de l'espace de recherche à travers les tailles de filtres, des canaux de sortie, des foulées, du nombre de couches, etc.

Nasnet se compose de couches convolutives appelées cellules répétées plusieurs fois avec une architecture identique et des poids différents. Chaque cellule de l'architecture présente la concaténation de différentes petites unités appelées blocs. Ces blocs consistent en des convolutions régulières, des convolutions séparables, une pooling maximale et une pooling moyenne. Chaque bloc crée une connexion résiduelle en mappant deux entrées à une seule carte de caractéristiques de sortie.

Nasnet a deux types de cellules convolutionnelles :

- Les cellules normales renvoient une carte d'entités de la même dimension et contiennent trois couches de convolution 3x3 et deux couches 5x5 circonvolutions séparables en profondeur,
- Cellules de réduction qui renvoient une carte d'entités réduite en hauteur et en largeur par un facteur de deux. La réduction des cellules contiennent une convolution séparable en profondeur 3x3, deux 5x5 et deux 7x7.

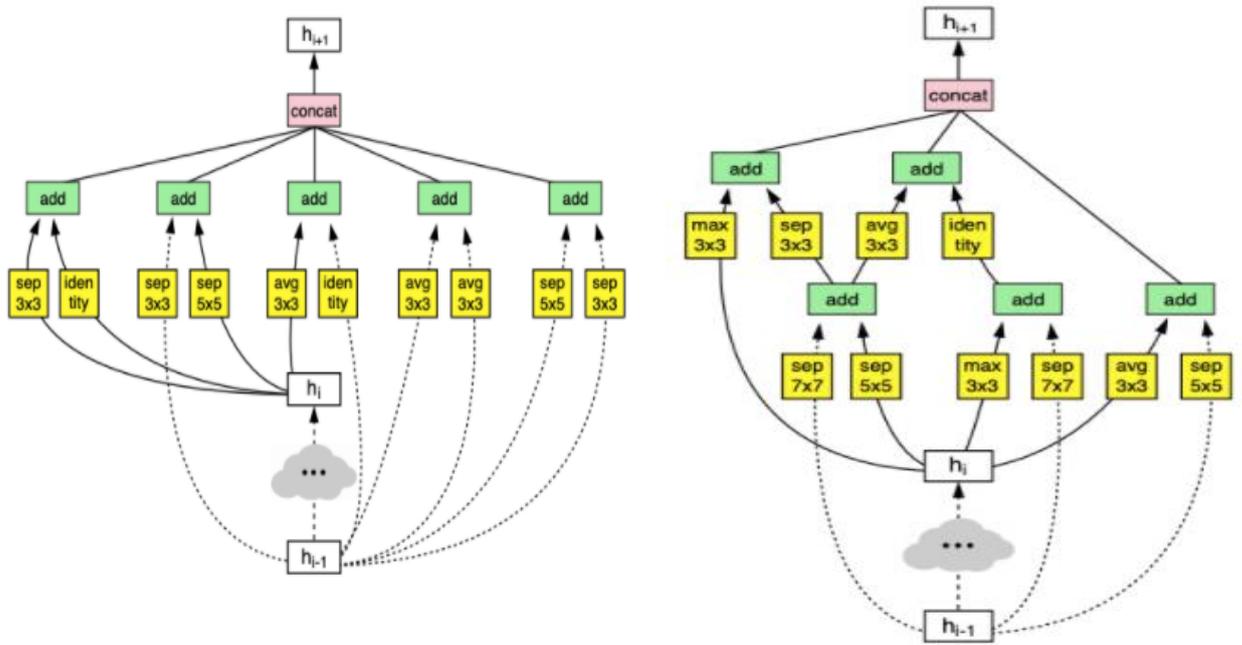


Figure 4.27 : Architecture d'une cellule convolutionnelle. A gauche cellule normale et à droite cellule réduite.

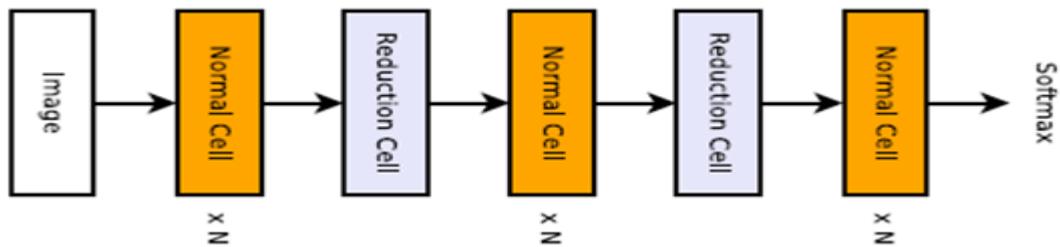


Figure 4.28 : Structure de l'architecture Nasnet.

5.16. EfficientNet

EfficientNet a été introduit par Tan en 2019 (Tan & Le, 2019). EfficientNet s'intéresse à la manière de mettre à l'échelle des architectures CNN de manière efficace en utilisant le calcul de paramètres de mise à l'échelle composite. La plus petite version d'EfficientNet est EfficientNet B0 (Figure 4.29). Cette architecture utilise principalement les goulots d'étranglement (Bottleneck). De plus, ils ont également ajouté une optimisation squeeze-and-excite et la puissante fonction d'activation Swish. C'est une activation qui a été optimisée en utilisant l'apprentissage de renforcement (NAS). La famille EfficientNets contient huit modèles différents, la version de base B0 avec une taille d'entrée de 224×224 . Les autres versions se dérivent d'EfficientNet-B0 en augmentant la résolution de l'image

en entrée, un certain nombre de cartes d'entités pour chaque couche et le nombre de couches en profondeur.

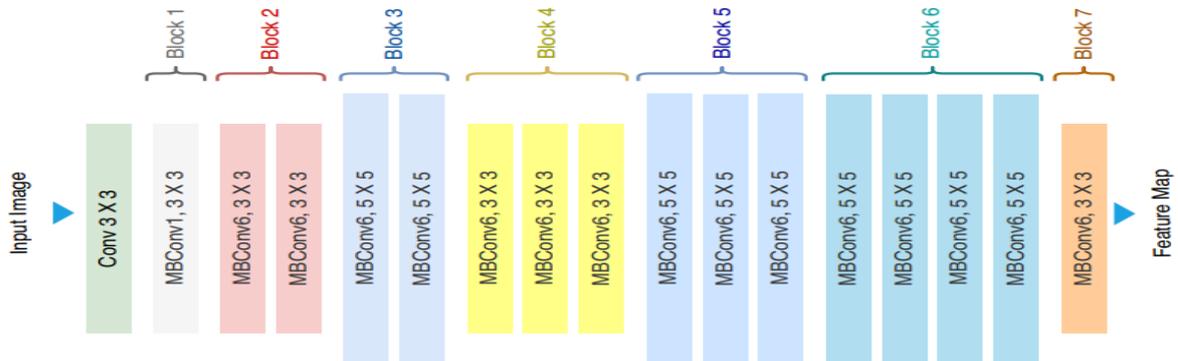


Figure 4.30 : Structure de l'architecture EfficientNet B0.

6. Conclusion

Pour conclure ce chapitre, le **Tableau 4.1** donne une synthèse comparative des différentes architectures CNN présentées ultérieurement.

Architecture	Année	Contribution	Paramètres	Profondeur
LeNet	1998	- Première architecture CNN populaire	0.060M	7
AlexNet	2012	- Plus profond et plus large que le LeNet - Utilise Relu, Dropout et chevauchement Pooling	60M	8
VGG	2014	- Topologie homogène - Petite taille de noyau	138M	16 , 19
GoogLeNet	2015	- Diviser Transformer Fusionner - Introduit le concept de bloc	4M	22
Inception-V3	2015	- Gère le problème d'un goulot d'étranglement représentationnel	23.6M	48

Chapitre 4 Méthodologie et modèles des réseaux de Neurones convolutionnels

		<ul style="list-style-type: none"> - Remplacez les filtres de grande taille par de petits filtres - Remplace le plus grand filtre par des filtres plus petits 		
Inception-V4	2016	<ul style="list-style-type: none"> - Diviser, transformer, fusionner - Utilise un filtre asymétrique 	43 M	27
ResNet	2016	<ul style="list-style-type: none"> - Connexion de saut basée sur l'apprentissage résiduel et la cartographie d'identité 	6.8M	18, 34, 50, 101, 152
SqueezeNet	2017	<ul style="list-style-type: none"> - Modélise les interdépendances entre les cartes de caractéristiques 	1.3M	18
DarkNet	2017	<ul style="list-style-type: none"> -Utilisation du mappage d'identité et des filtres de convolution 3×3 au lieu de couches entièrement connectées 	-	19, 53
DenseNet	2017	<ul style="list-style-type: none"> - Flux d'informations inter-couches 	25.6M	190
Xception	2017	<ul style="list-style-type: none"> - Convolution en profondeur suivie d'une convolution ponctuelle 	22.8M	71
Inception ResNet	2017	<ul style="list-style-type: none"> - Diviser, transformer, fusionner 	55.9M	164
ShuffleNet	2017	<ul style="list-style-type: none"> - Utilisation des convolutions de groupe par point et le mélange de canaux 	5.4 M	50

MobileNet	2017	- Utilise les convolutions séparables en profondeur	4.2 M	28
Nasnet	2018	- Utilisation des cellules convolutionnelles -Réseau optimisé en utilisant l'apprentissage par renforcement	5.3M	-
EfficientNet	2019	-Mise à l'échelle composite -Optimisation squeeze-and-excite -Utilisation de la fonction d'activation Swish	5.3M	237

Tableau 4.1 : Comparaison des architectures CNN.

Chapitre 5 : Expérimentation et Discussion

1	Introduction.....	113
2	Environnement.....	113
3	Métriques	115
4	Bases d'images dermoscopiques	117
5	Expérimentation et discussion	121
6	Conclusion.....	154

1. Introduction

L'extraction de caractéristiques est une étape cruciale dans les processus d'apprentissage automatique. En général, nous pouvons distinguer les caractéristiques artisanales et d'apprentissage en profondeur. Dans ce chapitre, nous étudions l'efficacité de l'utilisation des architectures de réseaux de neurones convolutifs (CNN) couramment pré-entraînés comme extracteurs de caractéristiques et de nombreux classificateurs d'apprentissage automatique pour évaluer la classification des lésions cutanées à partir de deux bases de données ISIC 2019 et PH2. Ainsi que nous avons effectués des expériences sur l'approche hiérarchique et séquentiel à partir de la base de données ISIC 2019.

2. Environnement

Un ordinateur équipé d'un processeur Core i7 et de 32 Go de DDRAM avec une carte graphique NVIDIA GeForce MX1080 a été utilisé pour réaliser les expériences. Pour mener à bien l'expérience, différentes bibliothèques Python sont utilisées, telles que CV2 pour la lecture et l'écriture d'images, PIL pour l'importation d'images, Matplotlib pour le traçage de graphiques, Pandas pour la lecture de l'ensemble de données, Sklearn pour l'encodage des étiquettes de l'ensemble de données et le prétraitement, Numpy, Keras, etc. Un MATLAB 2019 x64 bits a été utilisé pour exécuter d'autres programmes utilitaires.

2.1. Keras

Keras est une bibliothèque open de réseaux de neurones de haut niveau écrite en Python (sous licence MIT) basée principalement sur les travaux du développeur de Google François Chollet, keras interfaçable avec TensorFlow, CNTK et Theano, elle est considérée comme l'une des API les plus utilisées pour le développement et le test des réseaux de neurones. Elle permet d'aller de l'idée au résultat avec le plus faible délai possible, ainsi que de créer très facilement des couches pour les réseaux de neurones ou de mettre en place des architectures complexes. L'un des avantages de cette modularité est qu'il est très facile d'ajouter de nouvelles fonctionnalités sous forme de modules séparés. Keras est donc très flexible, et adapté à la recherche et à l'innovation.

2.2. Numpy

Numpy⁷ est la bibliothèque de base pour le calcul scientifique en Python. Il fournit un puissant objet tableau multidimensionnel hautes performances et des outils pour travailler avec ces tableaux. Les tableaux Numpy se déclinent essentiellement en deux versions : les vecteurs et les matrices. Les vecteurs sont strictement un tableau 1-d alors que les matrices sont 2-d mais les matrices ne peuvent avoir qu'une seule ligne/colonne.

2.3. Pandas

Pandas est une bibliothèque open source construite sur numpy fournissant des structures de données et des outils d'analyse de données hautes performances et faciles à utiliser pour le langage de programmation Python. Il permet une analyse rapide ainsi que le nettoyage et la préparation des données. Il excelle en termes de performances et de productivité. Il peut fonctionner avec des données provenant d'une grande variété de sources. Pandas convient à de nombreux types de données : données tabulaires, données de séries chronologiques, données matricielles arbitraires avec des étiquettes de ligne et de colonne, et toute autre forme d'ensembles de données d'observation/statistiques.

2.4. SciKit Learn

Sci-Kit Learn dispose divers modèles d'apprentissage automatique supervisés et non supervisés pour les applications de production et des outils statistiques pour analyser ces modèles. Il fournit également des fonctionnalités pour la réduction de la dimensionnalité, la sélection des caractéristiques, l'extraction des caractéristiques, les techniques d'ensemble et les ensembles de données intégrés. Sci-Kit Learn se concentre sur la qualité du code, la documentation, la facilité d'utilisation et les performances.

2.5. TensorFlow

TensorFlow est une plate-forme open source gratuite de bout en bout pour l'apprentissage automatique qui comprend un large éventail d'outils, de bibliothèques et de ressources. L'équipe Google Brain l'a publié pour la première fois le 9 novembre 2015. TensorFlow simplifie la conception et la formation de

⁷<https://www.numpy.org/>

modèles d'apprentissage automatique à l'aide d'API de haut niveau comme Keras. Il propose également différents niveaux d'abstraction. TensorFlow permet également de déployer des modèles de Machine Learning dans plusieurs environnements, y compris le cloud, le navigateur et les appareils.

2.6. OpenCV

Sous licence BSD, OpenCV est une bibliothèque gratuite d'apprentissage automatique spécialisée dans la vision par ordinateur. OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes en partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.

3. Métriques d'évaluation

Les métriques d'évaluation sont les mesures et les méthodes d'estimation des performances lors de la phase de l'évaluation de la précision et l'exactitude.

Une matrice de confusion est une matrice de statistiques de $N \times N$ dimension qui englobe tous les indicateurs afin de montrer plus de lisibilité sur l'évaluation

Cette matrice nous montre à quel point le modèle fonctionne, ce qui doit être amélioré et quelle erreur il commet, Il s'agit d'un tableau avec en colonne les différents cas prédits et en ligne les différents cas réels la matrice de confusion à 4 valeurs représentant les différentes combinaisons de valeurs réelles et valeurs prédites comme dans la [Figure 5.1](#) ci-dessous. Cette matrice est indispensable pour définir les différentes métriques de classification telles que l'exactitude, la sensibilité, la spécificité, la précision, l'exactitude équilibrée et le score F.

TP - Vrai Positif : il comprend toutes les valeurs de test de la catégorie positive qui ont été correctement prédite lors de l'évaluation.

TN - vrai négatif : il comprend toutes les valeurs de test de la catégorie négative qui ont été correctement prédit lors de l'évaluation.

FP - Faux positifs : il comprend toutes les valeurs de test de la catégorie positif qui ont été incorrectement prédit lors de l'évaluation.

FN - faux négatif : il comprend toutes les valeurs de test de la catégorie négatif qui ont été incorrectement prédit lors de l'évaluation.

- **Spécificité :**

La spécificité est le rapport entre les vrais négatifs et le total des négatifs dans les données, dans le cas multi-classes, nous nous concentrons sur une seule classe et traitons les autres comme la deuxième classe.

$$\text{Spécificité} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (5.4)$$

- **Score F1**

Le score F1 est la moyenne pondérée de la précision et du rappel.

$$\text{Score F1} = 2 * \frac{\text{Rappel} * \text{Précision}}{\text{Rappel} + \text{Précision}} \quad (5.5)$$

- **Exactitude équilibrée**

C'est la moyenne arithmétique de la sensibilité et de la spécificité, utiliser lorsqu'il s'agit de données déséquilibrées, c'est-à-dire lorsque l'une des classes cibles apparaît beaucoup plus que l'autre.

$$\text{Exactitude équilibrée} = \frac{\text{Spécificité} + \text{sensibilité}}{2} \quad (5.6)$$

4. Bases d'images dermoscopiques

4.1. Bases de données ISIC

ISIC (International Skin Imaging Collaboration) est devenue un référentiel pour les chercheurs en apprentissage automatique pour l'analyse d'images médicales, en particulier dans le domaine de la détection du cancer de la peau et de l'évaluation des malignités.

Elle fournit des ensembles de données d'images de lésions cutanées numériques haute résolution éprouvées par biopsie, avec des annotations et des métadonnées d'experts du monde entier. L'objectif est de promouvoir la recherche dans le domaine, qui conduira au développement de solutions automatisées de diagnostic assisté par ordinateur (DAO) pour le diagnostic du mélanome et d'autres cancers.

Le **Tableau 5.1** présente un résumé du nombre d'images dans les bases de données ISIC (2016-2020). Nous constatons que le nombre d'images a considérablement augmenté chaque année depuis son introduction.

Dataset	Apprentissage	Test	Total
ISIC 2016	900	379	1279
ISIC 2017	2000	600	2600
ISIC 2018	10015	1512	11527
ISIC 2019	25331	8238	33569
ISIC 2020	33126	10982	44108

Tablea 5.1 Résumés des bases ISIC.

ISIC 2016 ([Gutman et al., 2016](#)) contient 900 images d'entraînement et 379 images de test, soit un total de 1279 images. Des données de vérité terrain sont fournies pour les ensembles d'entraînement et de test, indiquant si chaque lésion est maligne ou bénigne.

ISIC 2017 ([Codella et al., 2017](#)) contient 2000 images d'entraînement et 600 images de test, soit un total de 2 600 images. La vérité terrain et les métadonnées du patient sont fournies pour les ensembles d'apprentissage et de test, indiquant si la lésion appartient à l'un des quatre groupes de classe : (1) mélanome ; (2) naevus ou kératose séborrhéique; (3) kératose séborrhéique; ou (4) mélanome ou naevus. L'âge et le sexe approximatifs du patient sont également fournis en tant que métadonnées supplémentaires.

En 2018, ISIC a partagé un ensemble de données plus substantiel ([Codella et al., 2018a](#) ; [Tschandl, 2018](#)) qui contient 10015 images d'entraînement et 1512 images de test, soit un total de 11527 images. Les données de vérité terrain sont fournies uniquement pour l'ensemble d'apprentissage, qui comprend des étiquettes de type de lésion plus détaillées, y compris le mélanome, le naevus mélanocytaire, le carcinome basocellulaire, la kératose actinique, la kératose bénigne, le dermatofibrome et les lésions vasculaires.

L'année suivante, ISIC 2019 ([Tschandl, 2018](#) ; [Codella et al., 2017](#) ; [Combalia et al., 2019](#)) a été publié. Cet ensemble de données contient 25331 images d'entraînement et 8238 images de test, soit un total de 33569 images. Les données de vérité terrain

sont fournies pour l'ensemble d'entraînement uniquement, indiquant les classes suivantes : mélanome, naevus mélanocytaire, carcinome basocellulaire, kératose actinique, kératose bénigne, dermatofibrome, lésions vasculaires et carcinome épidermoïde. L'ensemble de test se compose de 9 classes, 8 classes comme dans l'ensemble d'apprentissage plus une classe inconnue supplémentaire. Les métadonnées des patients sont fournies pour les ensembles d'apprentissage et de test. Les métadonnées d'apprentissage indiquent l'âge approximatif du patient, le site anatomique, l'ID de la lésion et le sexe. L'ID de lésion est spécifié pour 23 247 images et non spécifié pour 2 084 images, avec 11 848 ID uniques sur un total de 25 331 images. Les métadonnées de test indiquent l'âge approximatif du patient, son site anatomique et son sexe. L'ensemble de données ISIC 2019 se distingue également par l'inclusion de multiplats de lésions uniques qui présentent la même lésion à différents niveaux de zoom, ce qui peut fournir des caractéristiques uniques importantes à différents niveaux de grossissement.

Le **Tableau 5.2** présente la distribution des différentes classes de la base ISIC 2019.

Classe	Nb d'images	Pourcentage
Melanoma MEL	4522	17.85%
MelanocyticNevus NV	12 875	50.83%
Basal cellCarcinoma BCC	3323	13.12%
ActinicKeratosi s AK	867	3.42%
BenignKeratosi s BKL	2624	10.36%
Dermatofibroma DF	239	0.94%
VascularLesion VASC	253	1%
Squamoscellcarcinoma SCC	628	2.48%
None of the others UNK	0	0
Total	25 331	100%

Tableau 5.2. Distribution des différentes classes de la base ISIC 2019.

La **Figure 5.2** présente quelques images de cette base.

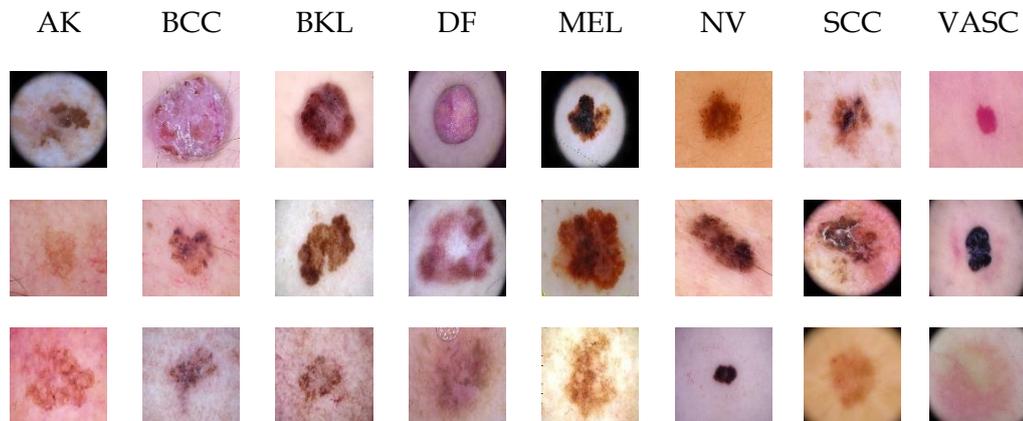


Figure 5.2. Quelques images de la base ISIC2019.

En 2020, le plus grand ensemble de données ISIC a été publié (**Rotemberg et al., 2020**) qui contient 33 126 images d'entraînement et 10 982 images de test, soit un total de 44 108 images. Comme l'année précédente, les données de vérité terrain sont fournies pour l'ensemble d'apprentissage uniquement, indiquant l'ID du patient, l'ID de la lésion, le sexe, l'âge approximatif, le site anatomique, le diagnostic et le statut bénin ou malin. Sur les 33 126 images de l'ensemble d'apprentissage, il existe 2 056 ID de patient uniques et 32 701 ID de lésion uniques. Cela suggérerait qu'un grand nombre d'images de lésions proviennent d'un groupe relativement restreint de patients à différents intervalles. L'ensemble de test comprend également des métadonnées patient indiquant l'ID du patient, l'âge approximatif du patient, le site anatomique et le sexe.

4.2. Base PH2

Le service dermatologique portugais de l'hôpital Pedro Hispano et l'Université de Porto ont collaboré pour établir la base de données PH2 ⁸(**Mendonça et al., 2013**). La base est composée de 200 images dermoscopiques classées en 160 bénignes et 40 mélanomes. Ce sont des images couleur RVB 8 bits avec une résolution de 768x560 pixels. Les images peuvent être obtenues en ligne auprès de l'hôpital Pedro Hispano. La vérité terrain est également fournie, qui est segmentée manuellement par des médecins ; classé comme normal, naevus atypique (bénin) ou mélanome. La **Figure 5.3** illustre quelques images de lésions pigmentées de la base PH2.

⁸<https://www.fc.up.pt/addi/ph2%20database.html>



Figure 5.3. Quelques images de la base PH2.

5. Expérimentation et discussion

Afin d'effectuer nous experimentation nous avons divisé la base ISIC 2019 composée de 25331 images en deux parties. La première partie correspond à 80% de l'ensemble de données (20265 images) pour l'apprentissage et la deuxième partie correspond à 20% (5066 images) utilisées pour les tests.

Le **Tableau 5.3** présente la répartition des classes en apprentissage et test de la base ISIC 2019.

Diagnostic	Nombre d'images	Pourcentage	Apprentissage	Test
Melanoma MEL	4522	17.85%	3641	881
MelanocyticNevus NV	12 875	50.83%	10317	2558
Basal cellCarcinoma BCC	3323	13.12%	2621	702
ActinicKeratosi s AK	867	3.42%	698	169
BenignKeratosi s BKL	2624	10.36%	2078	546
Dermatofibroma DF	239	0.94%	199	40
VascularLesion VASC	253	1%	217	36
Squamoscellcarcinoma SCC	628	2.48%	494	134
None of the others UNK	0	0	0	0
Total	25 331	100%	20265	5066

Tableau 5.3. Répartition des classes en apprentissage et test.

5.1. Approche pipeline en deux étapes avec la base ISIC 2019⁹

La Figure 5.4 illustre l'organigramme de l'approche proposée. Le processus de classification est composé de quatre parties successives. Le premier composant en entrée correspond aux images dermoscopiques, le second est la partie d'extraction de caractéristiques avec 17 architectures de réseaux de neurones convolutifs (CNN) pré-entraînés. A partir des caractéristiques obtenues, deux bases sont créées : une base d'apprentissage et une base de test. Le troisième composant du système est l'étape d'apprentissage par un ensemble de classificateurs, chacun étant effectué individuellement. Enfin, la dernière est l'étape de validation à partir du jeu de données de test.

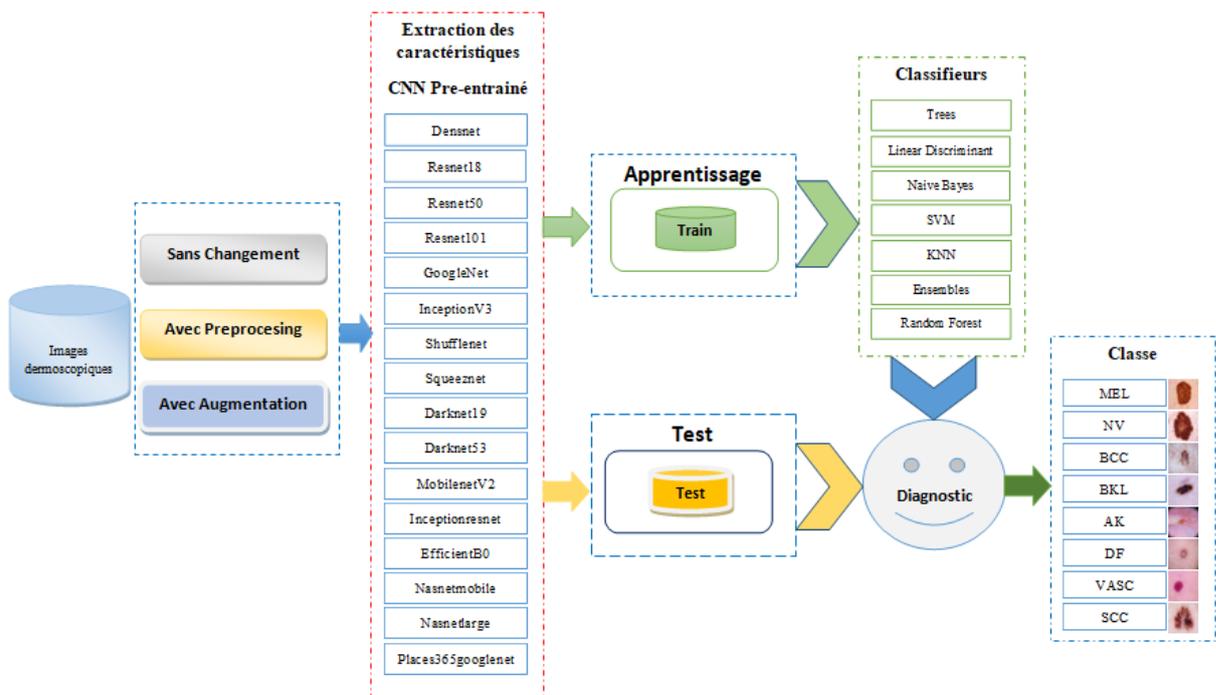


Figure 5.4. Méthodologie de l'approche pipeline en deux étapes avec la base ISIC2019.

17 modèles CNN pré-entraînés ont été utilisé pour extraire les caractéristiques des images dermoscopiques en entrée (Tableau 5.4), résultant ainsi 17 bases de caractéristiques. Pour chaque base de caractéristiques extraite, 24 classifieurs ont été appliqué.

⁹ Samia Benyahia, Boudjelal Meftah, Olivier Lézoray, Multi-Features Extraction Based on Deep Learning for Skin Lesion Classification, Tissue and Cell, Vol 74, 101701, 2022

Tous les classifieurs, quelle que soit l'architecture CNN utilisée, ont été entraînés à l'aide du même ensemble d'entraînement.

Pour établir l'efficacité du modèle, chaque classifieur a été évalué sur le même ensemble de test contenant des images originales.

Dans cette partie, nous définissons quatre expériences réalisées avec la base de données ISIC 2019 :

- a) Évaluation de la méthode proposée en utilisant la base de données originale.
- b) Évaluation de la méthode proposée en utilisant l'ensemble de données d'origine avec prétraitement d'images.
- c) Évaluation de la méthode proposée avec le jeu de données augmenté.
- d) Évaluation de la méthode avec réduction de données (PCA).

Le **Tableau 5.4** suivant illustre les différents paramètres des architectures convolutionnelles utilisées :

sArchitecture	Nombre de paramètres	Nombre de caractéristiques	Profondeur	Taille image en entrée	Nombre de couches
Googlenet	4M	1024	22	224x224	144
InceptionV3	23.8M	2048	48	299x299	316
ResNet18	11.17M	512	18	224x224	72
ResNet50	25.6M	2048	50	224x224	177
ResNet101	44.7M	2048	101	224x224	347
SqueezeNet	1.2M	1000	18	227x227	68
Darknet-19	15.7M	1000	19	256x256	64
Darknet-53	61.5M	1000	53	256x256	184
InceptionResNet	55.8M	1536	164	299x299	824
Xception	22.9M	2048	71	299x299	171
MobileNet	4.2M	1280	28	224x224	154
DenseNet201	20.2M	1920	201	224x224	709
Shufflenet	3.4M	544	50	224x224	173
NASNetMobile	5.3M	1056	-	224x224	914
NASNetLarge	88.9M	4032	-	331x331	1244

EcientNetB0	5.3M	1280	237	224x224	290
Places365 Googlenet	4M	1024	22	224x224	144

Tableau 5.4. Informations sur les différentes architectures CNN utilisées.

Le **Tableau 5.5** suivant illustre les paramètres des classifieurs utilisés.

Classifieur	Paramètre
Fine Tree	Divisions maximales : 100 divisions
Medium Tree	Divisions maximales : 20 divisions
Coarse Tree	Divisions maximales : 4 divisions
Linear SVM	Fonction noyau : méthode multi-classes linéaire : un contre un
Kernel function	Fonction noyau : quadrature multi-classes Méthode : un contre un
Cubic SVM	Fonction noyau : cubique multi-classes Méthode : un contre un
Fine KNN	Nombre de voisins : 1 métrique de distance : pondération de la distance euclidienne
Medium KNN	Nombre de voisins : 10 métriques de distance : pondération de la distance euclidienne
Weighted KNN	Nombre de voisins : 10 métriques de distance : pondération de la distance euclidienne : inverse au carré
Boosted Trees	Méthode d'ensemble : AdaBoost type d'apprenant : arbre de décision divisions maximales : 20 nombre d'apprenants : 30
Subspace KNN	Méthode d'ensemble : sous-espace type d'apprenant : plus proche voisin nombre d'apprenants : 30
RUSBoosted Trees	Méthode d'ensemble : RUSBoost type d'apprenant : arbre de décision nombre d'apprenants : 30 fractionnements maximum : 20

Tableau 5.5. Paramètre des classifieurs.

5.1.1. Méthode proposée en utilisant le jeu de données original.

Les **Tableau 5.6** et **Tableau 5.7** présentent les taux d'exactitude de classification obtenus à partir de la base ISIC 2019 originale.

	Densenet 201	Resnet18	Resnet50	Resnet101	Shufflenet	Squeezenet	Darknet19	Darknet53
Fine Tree	59,24	58,40	59,60	60,09	58,40	59,20	57,10	57,50

Medium Tree	58,53	56,90	59,30	58,76	56,90	57,50	56,80	56,90
Coarse Tree	54,32	55,10	57,30	54,70	55,50	54,30	54,20	54,10
Linear Discriminant	73,41	66,50	71,10	71,97	67,40	67,40	67,40	69,70
Gaussian Naive Bayes	48,14	41,20	42,00	44,49	34,20	34,50	40,70	38,50
Kernel Naive Bayes	51,15	54,60	45,80	42,01	35,40	38,60	45,00	43,70
Linear SVM	72,40	68,00	71,80	73,31	69,10	67,60	66,00	68,10
Quadratic SVM	78,68	72,70	75,60	77,22	73,70	72,90	72,10	73,70
Cubic SVM	80,87	74,50	77,30	78,27	75,50	74,00	73,90	75,30
Fine Gaussian SVM	51,00	51,20	51,00	51,60	52,30	57,20	51,70	53,00
Medium Gaussian	75,40	71,80	73,60	74,27	72,00	69,30	70,90	71,60
Coarse SVM	67,82	64,80	66,80	68,22	64,90	64,30	64,30	64,60
Fine KNN	77,28	72,30	73,70	73,88	70,90	69,20	70,20	70,70
Medium KNN	61,90	66,80	67,30	69,50	66,10	66,30	65,70	66,20
Coarse KNN	64,10	62,40	63,20	63,28	61,00	61,90	62,80	61,10
Cosine KNN	69,90	67,50	69,70	71,33	67,50	66,00	66,70	66,90
Cubic KNN	68,23	66,50	67,50	71,65	65,30	66,20	66,00	65,90
Weighted KNN	72,80	71,00	71,30	68,61	69,30	69,50	69,70	70,10
Boosted Trees	61,00	58,30	61,00	62,50	59,20	59,40	58,90	59,20
Bagged Trees	65,10	63,60	64,10	67,53	63,60	65,20	65,00	64,30
Subspace Discriminant	72,60	66,10	70,30	72,17	66,60	67,50	67,30	68,50
Subspace KNN	76,4	73,70	76,30	74	75,00	69,20	70,30	70,70
RUSBoosted Trees	45,00	43,10	44,60	46,88	47,90	40,00	42,60	40,00
Random forest	65,77	63,48	64,17	64,07	63,11	65,52	66,68	64,88

Tableau 5.6. Résultat d'évaluation par l'utilisation de 8 architectures (%).

	Googlenet	Mobilenet v2	Xception	Nasnetmobile	Inceptionv3	Efficientb0	InceptionResnet	NasnetLarge	Places365Googlenet
Fine Tree	58,10	57,30	55,70	56,10	55,60	58,30	57,70	56,90	58,60
Medium Tree	56,60	56,70	54,80	55,70	55,80	56,70	56,40	56,20	58,20
Coarse Tree	55,00	54,60	53,20	54,80	52,90	53,60	54,80	55,20	55,70
Linear Discriminant	65,00	66,60	67,60	65,00	67,10	69,40	67,90	Fail	65,30
Gaussian Naive Bayes	38,60	38,50	42,50	40,00	41,00	39,80	42,60	Fail	Fail
Kernel Naive Bayes	46,50	49,30	42,60	42,02	50,70	51,70	49,80	48,90	52,10
Linear SVM	67,40	68,90	67,20	66,00	66,90	71,30	69,20	68,20	66,80
Quadratic SVM	71,50	72,30	71,00	69,70	72,10	75,40	73,70	70,80	70,30
Cubic SVM	72,60	74,50	72,90	71,20	74,00	76,90	75,40	71,90	70,90
Fine Gaussian SVM	51,30	51,10	51,00	51,00	51,00	51,50	51,70	52,30	53,50
Medium Gaussian	69,50	71,20	70,00	68,70	69,90	73,10	72,00	67,70	68,10
Coarse SVM	64,20	65,70	63,60	63,90	63,70	66,30	65,20	62,90	62,70
Fine KNN	69,20	72,10	70,90	66,50	70,20	73,60	70,20	65,90	67,00
Medium KNN	65,20	66,60	64,80	63,40	65,10	67,60	66,00	62,10	64,70
Coarse KNN	61,50	61,10	60,00	60,30	60,60	62,10	61,60	58,50	61,90

Cosine KNN	65,70	67,80	66,20	64,70	66,00	68,00	67,30	63,90	64,80
Cubic KNN	64,30	65,70	64,20	63,80	65,20	67,20	65,90	62,30	64,30
Weighted KNN	68,30	69,50	67,80	66,20	68,50	71,90	69,30	64,80	69,00
Boosted Trees	59,20	58,40	57,60	57,50	58,80	59,80	58,70	58,40	59,60
Bagged Trees	64,40	62,60	61,30	61,60	62,00	65,00	64,00	62,10	64,60
Subspace Discriminant	65,90	67,40	67,60	65,60	67,10	69,10	68,60	69,30	65,30
Subspace KNN	71,40	74,30	71,80	66,90	71,10	75,10	69,90	Fail	68,10
RUSBoosted Trees	38,40	42,10	39,10	35,10	38,80	41,80	40,20	40,10	39,70
Random forest	64,51	66,89	60,96	59,40	63,55	65,84	64,80	64,68	64,03

Tableau 5.7. Résultat d'évaluation par l'utilisation de 9 architectures (%).

La **Figure 5.5** illustre sous forme graphique les différents taux d'exactitude de classification obtenue à partir la base ISIC 2019 par l'utilisation de l'architecture Densenet201.

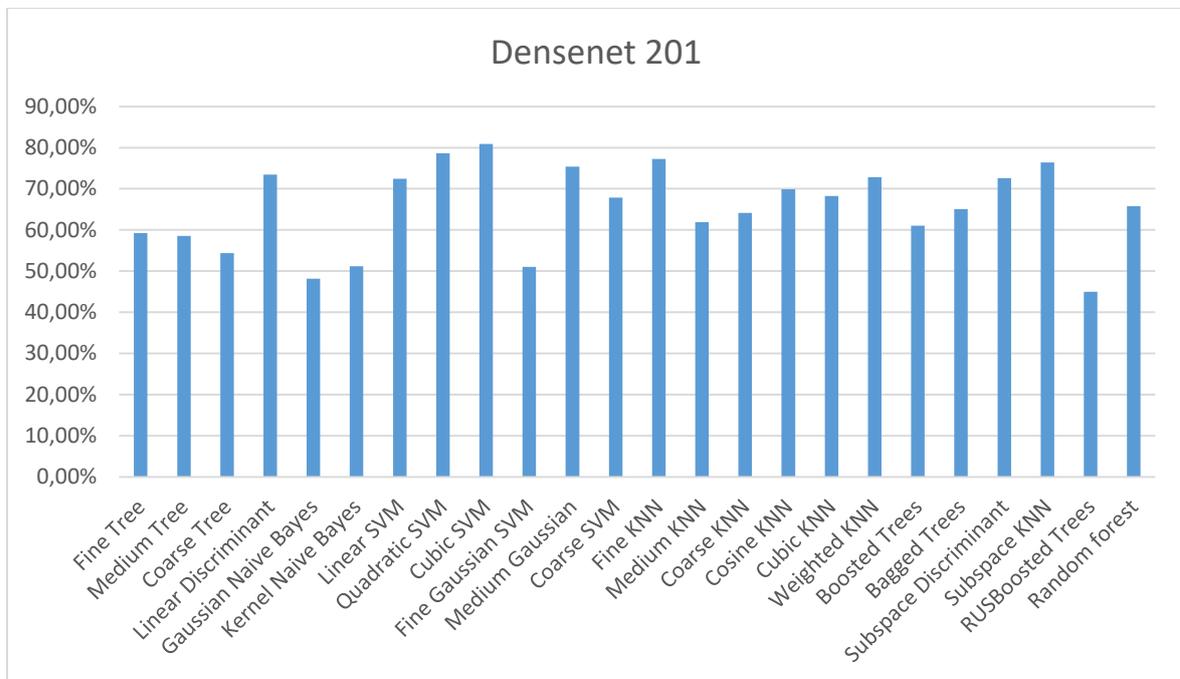


Figure 5.5. Taux de classification par DenseNet201 sans prétraitement de la base.

D'après les résultats obtenus à partir de cette expérience, nous pouvons voir que le taux de classification obtenu à partir du classificateur SVM cubique est supérieur à tous les autres taux obtenus à partir des autres classifieurs. D'autre part, lorsqu'on compare le taux de classification obtenu à partir des différents CNN, nous observons que DenseNet201 obtient le plus grand taux avec le meilleur classifieur. D'après les résultats de cette expérience, nous pouvons observer que la combinaison de DenseNet201 avec le classifieur SVM cubique offre le plus grand taux avec 80,87%.

5.1.2. Méthode proposée avec prétraitement de la base ISIC 2019

La phase de prétraitement vise à améliorer la qualité des images dermoscopiques pour assurer une meilleure efficacité de détection des lésions (Figure 5.6). Nous avons utilisé DullRazor (Kiani & Sharafat, 2011) pour supprimer les artefacts capillaires.

Au total 25331 images améliorées de l'ensemble de données ISIC 2019 est divisé en deux parties. La première la partie correspond à 80% de la base (20265 images) dédiée à l'apprentissage et la deuxième partie correspond à 20% (5066 images) utilisées pour le test.

Pour établir l'efficacité du modèle, chaque classificateur a été évalué sur le même ensemble de test contenant le prétraitement images. Le Tableau 5.8 et Tableau 5.9 présente les taux de classification sur ISIC 2019 prétraitée.

	Densenet 201	Resnet18	Resnet50	Resnet101	Shufflenet	Squeezenet	Darknet19	Darknet53
Fine Tree	60,70	58,40	60,40	60,40	58,20	58,20	57,70	58,70
Medium Tree	58,30	57,60	58,60	59,20	57,00	58,10	57,50	57,10
Coarse Tree	55,40	55,40	56,10	55,70	54,20	55,30	55,90	54,80
Linear Discriminant	72,50	66,10	69,70	70,00	66,40	66,80	67,60	68,90
Gaussian Naive Bayes	50,50	40,80	42,50	43,90	Fail	39,60	42,60	40,30
Kernel Naive Bayes	53,30	54,40	44,40	42,00	40,20	39,30	46,10	45,00
Linear SVM	72,00	67,70	70,70	71,60	67,50	67,20	68,20	68,80
Quadratic SVM	77,80	71,90	75,00	75,30	71,30	71,40	72,10	73,50
Cubic SVM	79,50	73,20	75,90	76,70	72,10	72,50	73,60	75,50
Fine Gaussian SVM	67,40	51,50	51,10	51,20	52,70	56,70	52,10	52,80
Medium Gaussian	75,00	71,30	72,40	73,00	69,70	68,50	71,30	72,20
Coarse SVM	67,40	65,10	66,30	66,70	64,40	63,50	64,60	65,40
Fine KNN	76,20	71,90	73,10	72,80	70,60	68,20	68,60	71,40
Medium KNN	69,30	66,70	67,50	67,50	65,20	65,50	66,40	67,00
Coarse KNN	65,20	63,80	64,60	63,10	62,50	62,00	63,10	63,00
Cosine KNN	69,80	67,80	69,10	68,50	67,10	65,20	65,80	68,10
Cubic KNN	69,30	66,90	67,40	66,70	64,40	65,90	65,90	67,00
Weighted KNN	73,40	70,00	70,90	70,30	68,10	68,90	69,30	71,40
Boosted Trees	61,30	59,40	60,50	61,40	59,40	59,30	59,70	60,00
Bagged Trees	66,60	63,60	65,70	66,60	63,80	64,80	65,50	66,40
Subspace Discriminant	71,90	66,50	69,10	69,40	65,80	66,60	67,50	68,10
Subspace KNN	76,70	73,50	76,00	74,40	74,80	67,40	68,80	71,20

RUSBoosted Trees	47,80	44,70	47,20	45,50	44,70	42,10	44,60	41,00
Random forest	68,18	65,20	67,24	67,14	65,54	67,88	67,02	66,36

Tableau 5.8. Résultat d'évaluation par l'utilisation de 8 architectures avec prétraitement de la base (%).

	Googlenet	Mobilenet v2	Xception	Nasnetmobile	Inceptionv3	Efficient b0	Inception Resnet	Nasnet Large	Places365 Googlenet
Fine Tree	58,80	56,50	55,90	56,60	56,80	58,70	57,40	57,60	58,70
Medium Tree	57,60	56,80	55,00	56,10	56,00	57,50	56,80	56,50	57,30
Coarse Tree	54,00	54,80	54,60	54,40	53,40	55,20	54,30	55,10	54,00
Linear Discriminant	65,90	66,70	68,30	64,60	67,00	69,80	68,50	67,5	66,00
Gaussian Naive Bayes	39,91	34,20	39,40	36,10	42,60	40,70	40,10	39,3	38,9
Kernel Naive Bayes	48,10	41,00	43,50	46,10	51,30	51,80	49,10	48,00	50,70
Linear SVM	67,80	68,80	68,70	66,40	67,60	70,90	68,60	67,30	66,10
Quadratic SVM	71,60	72,50	71,80	69,70	71,60	75,90	73,00	69,90	70,20
Cubic SVM	71,70	72,70	73,80	70,80	73,10	77,90	74,10	70,30	71,20
Fine Gaussian SVM	51,70	51,10	51,10	51,10	51,00	51,50	51,50	52,80	53,30
Medium Gaussian	68,80	70,70	70,00	69,00	70,30	73,10	70,70	67,60	67,90
Coarse SVM	64,50	65,40	64,80	64,00	64,50	66,30	64,80	63,30	61,90
Fine KNN	68,20	71,10	69,80	64,70	69,90	71,80	69,00	65,50	66,60
Medium KNN	65,40	65,90	66,30	64,20	66,20	67,20	65,40	63,30	64,20
Coarse KNN	62,20	61,10	61,40	60,40	62,30	63,10	61,90	59,60	62,40
Cosine KNN	66,60	68,10	67,20	64,80	67,20	68,50	65,90	64,90	64,80
Cubic KNN	64,50	65,30	65,90	64,00	66,10	66,70	64,90	63,30	63,90
Weighted KNN	67,90	69,40	69,00	66,30	69,20	71,30	68,60	65,70	68,30
Boosted Trees	59,60	58,70	58,40	58,20	58,50	60,80	59,00	58,80	58,40
Bagged Trees	64,10	63,30	62,50	61,20	62,30	65,40	64,00	62,40	65,50
Subspace Discriminant	65,60	67,30	68,00	64,60	67,30	69,40	68,00	65,4	66,20
Subspace KNN	69,80	73,56	71,00	63,90	71,00	73,70	69,50	70,2	67,60
RUSBoosted Trees	41,90	40,70	40,50	34,80	39,30	43,5	40,50	39,60	38,90
Random forest	65,98	67,40	63,77	61,09	66,60	66,84	65,70	66,06	64,82

Tableau 5.9. Résultat d'évaluation par l'utilisation de 9 architectures avec prétraitement de la base (%).

La **Figure 5.7** illustre sous forme graphique les différents taux d'exactitude de classification obtenue à partir la base ISIC 2019 par l'utilisation de l'architecture Densenet201 après prétraitement de la base.

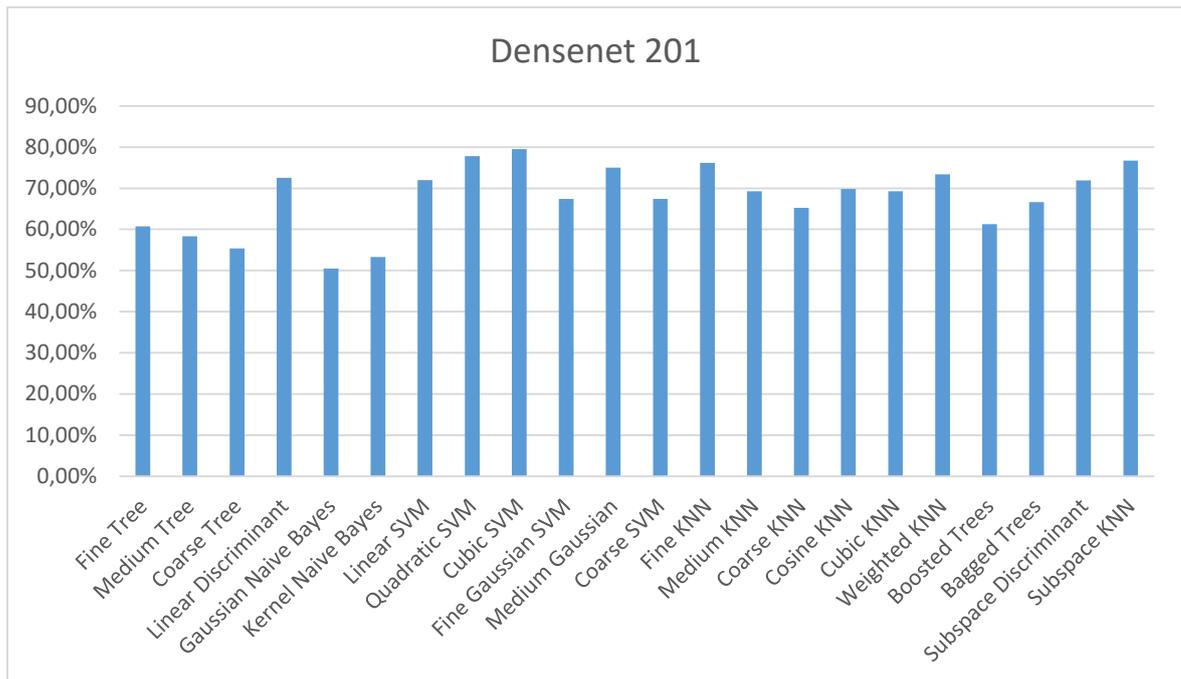


Figure 5.6. Taux de classification par DenseNet201 après prétraitement de la base.

D'après les résultats obtenus à partir de cette expérience, nous pouvons voir que le taux de classification obtenu à partir du classificateur SVM cubique est supérieur à tous les autres taux obtenus à partir des autres classificateurs. D'autre part, lorsqu'on compare le taux de classification obtenu à partir des différents CNN, nous observons que DenseNet201 obtient le plus grand taux avec le meilleur classificateur. D'après les résultats de cette expérience, nous pouvons observer que la combinaison de DenseNet201 avec le classificateur SVM cubique offre le plus grand taux avec 79,50 %.

5.1.3. Méthode proposée avec augmentation de la base ISIC 2019

L'augmentation des données consiste à appliquer certaines transformations à l'ensemble d'apprentissage initial pour créer de nouveaux exemples artificiels. Les transformations appliquées ne changent pas la nature de la classe détectée et créent ainsi de nouveaux exemples. L'augmentation est effectuée par l'utilisation du logiciel Augmentor¹⁰. Les différentes opérations effectuées sont illustrées dans le **Tableau 5.10** suivant :

Opération	Paramètres
Rotation	sans recadrage (1, 45,45), sans recadrage (1, 90,90), sans recadrage (1, 180,45), sans recadrage (1,45,180)

¹⁰<https://github.com/mdbloice/Augmentor>

Changement de contraste	(1, 1,2.5)
Zoom	(1, 1,1.7), (1, 1,2.5)
Rotation de haut en bas	///
Rotation de gauche à droite	///
Etirement	(1, 1,15), (1, 1,20)
Changement intensité RVB	(1, 2,2), (1, 1,2)
Changement de luminosité	(1,0.99, 1.3), (1,0.4, 1)
Distorsion aléatoire	(1, 20, 25,15)
Recadrage centré	(1,0.9)
Incliner gauche droite Incliner les coins	///

Tableau 5.10 : Différents paramètres des opérations d'augmentation utilisées.

Il est clairement remarquable que le nombre d'images dans chaque classe de l'ensemble de données ISIC 2019 soit significativement différent. Il y a un énorme écart dans le nombre d'images dans chaque classe. La classe NV contient 12875 images contrairement AKIEC, DF, VASC et le SCC. Par exemple, la classe NV est 54 fois supérieure à la classe DF. Avec l'intention d'augmenter le nombre d'images d'entraînement et de réduire les effets négatifs d'importants déséquilibres de classe et de améliorer l'efficacité de CNN, nous avons suivi certaines des techniques d'augmentation telles que les rotations, zoom, cisaillement, recadrage, retournements (haut-bas, gauche-droite), inclinaison gauche-droite et coin, contraste, ajout aléatoire couleur et luminosité aléatoire.

Le **Tableau 5.11** indique le nombre d'échantillons d'ensembles de données ISIC 2019 avant et après l'augmentation. Après l'augmentation des données la taille du jeu de données ISIC 2019 augmente d'un facteur 2,5 avec un total de 63661 images au total. Dans cette troisième expérience, nous avons divisé le nouveau jeu de données obtenu en deux parties. La première partie était de 80% équivaut à 50929 images de l'ensemble de données pour apprentissage et la deuxième partie était de 20% équivaut à 12732 images utilisées pour les tests. Pour établir l'efficacité du modèle, chaque classificateur a été évalué sur le même ensemble de test contenant Images ISIC 2019.

	Originale	Augmentée
MEL	4522	9044
NV	12875	10000
BCC	3323	6646

AK	867	7803
BKL	2624	7872
DF	239	7170
VASC	253	7590
SCC	628	7536
Totale	25331	63661

Tableau 5.11. Répartition des données avant et après l'augmentation.

Les [Tableau 5.12](#) et [Tableau 5.13](#) présentent les résultats de classification de la base ISIC 2019 après augmentation.

	Densenet 201	Resnet18	Resnet50	Resnet101	Shufflenet	SqueezeNet	Darknet19	Darknet53
Fine Tree	41,11	37,14	41,43	41,16	40,84	39,25	39,30	37,46
Medium Tree	35,46	34,90	37,70	38,76	34,80	35,60	34,76	34,97
Coarse Tree	26,72	26,30	29,46	26,37	27,80	26,40	26,43	32,12
Linear Discriminant	76,29	60,34	71,77	72,31	16,76	61,02	65,20	67,29
Gaussian Naive Bayes	48,60	41,40	42,30	44,45	34,60	34,54	40,32	38,71
Kernel Naive Bayes	38,50	42,20	33,84	29,40	24,12	26,86	33,64	31,70
Linear SVM	83,12	79,52	82,12	84,20	71,42	71,67	72,94	75,34
Quadratic SVM	88,52	79,82	84,35	84,10	75,79	79,68	78,25	80,56
Cubic SVM	91,71	83,79	86,87	86,95	81,06	82,87	83,53	84,98
Fine Gaussian SVM	52,30	52,60	51,62	52,97	53	58,04	53,10	53,60
Medium Gaussian	76,52	72,84	74,62	75,12	72,94	71,30	74,30	72,07
Coarse SVM	86,69	81,04	68,85	80,10	82,36	80,70	82,34	81,76
Fine KNN	92,34	87,49	88,07	88,01	83,92	86,24	86,26	86,73
Medium KNN	70,40	74,80	81,70	77,50	74,10	77,20	73,70	74,20
Coarse KNN	73,24	71,40	76,30	72,28	70	75,30	71,80	70,10
Cosine KNN	77,72	74,50	70,20	78,33	74,50	68,90	73,70	73,90
Cubic KNN	78,03	75,50	78,70	80,65	74,30	75	75	74,90
Weighted KNN	79,20	77	73,50	74,61	75,30	72,20	75,70	76,10
Boosted Trees	69,40	65,30	78,30	69,50	66,20	76,50	65,90	66,20
Bagged Trees	74,24	72,60	70	76,53	72,60	68,40	74	73,30
Subspace Discriminant	73,22	57,94	79,45	69,20	59,47	58,82	62,46	63,53
Subspace KNN	77,52	75,64	78,40	74,62	76,43	71,55	71,23	72,94
RUSBoosted Trees	38,23	37,11	38,40	40,01	41,62	34,10	36,71	34,12
Random forest	70,50	68,30	69,54	70,34	68,74	70,28	70,22	68,56

Tableau 5.12. Résultat d'évaluation par l'utilisation de 8 architectures avec augmentation de la base (%).

	Googlenet	Mobilenet v2	Xception	Nasnetmobile	Inceptionv3	Efficient b0	Inception Resnet	Nasnet Large	Places365 Googlenet
Fine Tree	37,37	37,66	35,97	35,16	36,19	39,49	37,62	36,63	37,93
Medium Tree	34,59	35,25	32,40	33,75	33,46	32,82	34,42	34,23	36,32
Coarse Tree	27,10	26,58	25,22	26,92	24,90	26,14	26,83	27,13	27,72
Linear Discriminant	59,19	64,72	67,66	59,77	66,55	69,82	68,54	Fail	58,78
Gaussian Naive Bayes	38,86	38,64	42,91	40,34	41,24	40	42,76	Fail	Fail
Kernel Naive Bayes	33,40	37,41	41,30	31,40	38,70	39,30	37,40	36,94	40,10
Linear SVM	72,94	75,46	76,32	72,83	74,68	81,62	79,83	77,61	70,08
Quadratic SVM	76,23	79,68	80,70	75,91	80,14	85,65	82,46	79,85	73,49
Cubic SVM	80,39	83,29	83,99	79,91	84,56	89,36	86,86	82,53	78,57
Fine Gaussian SVM	52,04	52,11	52,30	52,61	51,60	51,87	52,43	53,10	53,10
Medium Gaussian	72,30	73,80	71,64	72,01	70,40	73,40	72,90	70,34	70,38
Coarse SVM	79,50	80,94	80,13	70,66	80,61	80,71	81,46	80,16	78,94
Fine KNN	83,39	86,70	86,00	81,14	87,06	89,63	86,56	83,15	80,77
Medium KNN	73,20	74,60	72,80	71,40	73,10	75,60	74	70,10	72,70
Coarse KNN	70,50	70,10	69	69,30	69,60	71,10	70,60	67,50	70,90
Cosine KNN	72,70	74,80	73,20	71,70	73	75	74,30	72,57	71,80
Cubic KNN	73,30	74,70	73,20	72,80	74,20	76,20	74,90	71,35	73,30
Weighted KNN	74,30	75,50	73,80	72,20	74,50	77,90	75,30	70,73	75
Boosted Trees	66,20	65,40	64,60	64,50	65,80	66,80	65,70	65,41	66,60
Bagged Trees	73,40	71,60	70,30	70,60	71	74	73	71,23	73,60
Subspace Discriminant	56,12	62,26	65,55	57,60	64,33	66,96	65,94	68,65	57,01
Subspace KNN	72,24	74,83	72,03	67,32	72,26	76,00	70,14	Fail	70,12
RUSBoosted Trees	32,60	36,23	33,24	30,23	32,82	35,65	34,54	34,73	33,42
Random forest	68,18	69,60	65,97	63,29	69,70	69,14	68,10	68,36	66,22

Tableau 5.13. Résultat d'évaluation par l'utilisation de 9 architectures avec augmentation de la base (%).

La **Figure 5.7** illustre sous forme graphique les différents taux d'exactitude de classification obtenus à partir la base ISIC 2019 par l'utilisation de l'architecture Densenet201 avec augmentation de la base.

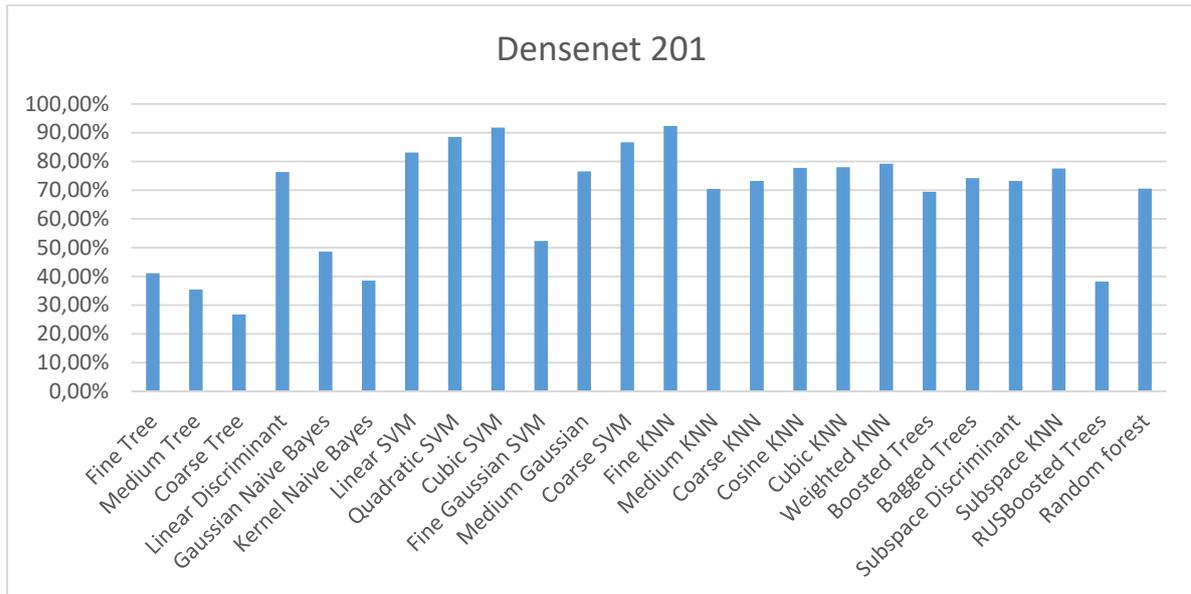


Figure 5.7. Taux de classification par DenseNet201 après augmentation de la base.

En comparant les résultats obtenus à partir de chaque CNN séparément avec les 24 classificateurs, nous pouvons observer que le taux de classification obtenu à partir du Fine KNN combiné avec le classificateur DenseNet201 CNN est supérieure à toute précision obtenu à partir des autres classificateurs avec un taux de classification de 92,34 % et le taux de classification obtenu à partir du cubique SVM combiné avec le classificateur DenseNet201 est 91,71 %.

5.2. Réduction des caractéristiques de base ISIC 2019

L'analyse des données consiste essentiellement à établir quelles sont les relations existant entre les observations, entre les variables. L'objectif de l'ACP est de réduire le nombre de fonctionnalités incluses tout en capturant les informations clés, ou la propagation des données, telle que mesurée par la variance. Nous allons générer ce nouvel ensemble de fonctionnalités plus petit en combinant linéairement les fonctionnalités au sein de l'ensemble de fonctionnalités d'origine. Les nouvelles fonctionnalités sont appelées composants principaux. La première composante principale est la ligne qui capture la majorité de la variance dans l'ensemble de données, et la deuxième composante principale capture la deuxième variance la plus importante, et ainsi de suite. Chacune de ces composantes principales est orthogonale à la suivante.

La [Figure 5.8](#) illustre l'organigramme de l'approche proposée. Le processus de classification est composé de quatre parties successives. Le premier composant en

entrée correspond aux images dermoscopiques, le second est la partie d'extraction de caractéristiques avec 17 architectures de réseaux de neurones convolutifs (CNN) pré-entraînés. A partir des caractéristiques obtenues, une réduction par ACP est effectuée, ce que nous permet d'avoir deux bases réduites : une base d'apprentissage et une base de test. Le troisième composant du système est l'étape d'apprentissage par un ensemble de classificateurs, chacun étant effectué individuellement. Enfin, la dernière est l'étape de validation à partir du jeu de données de test.

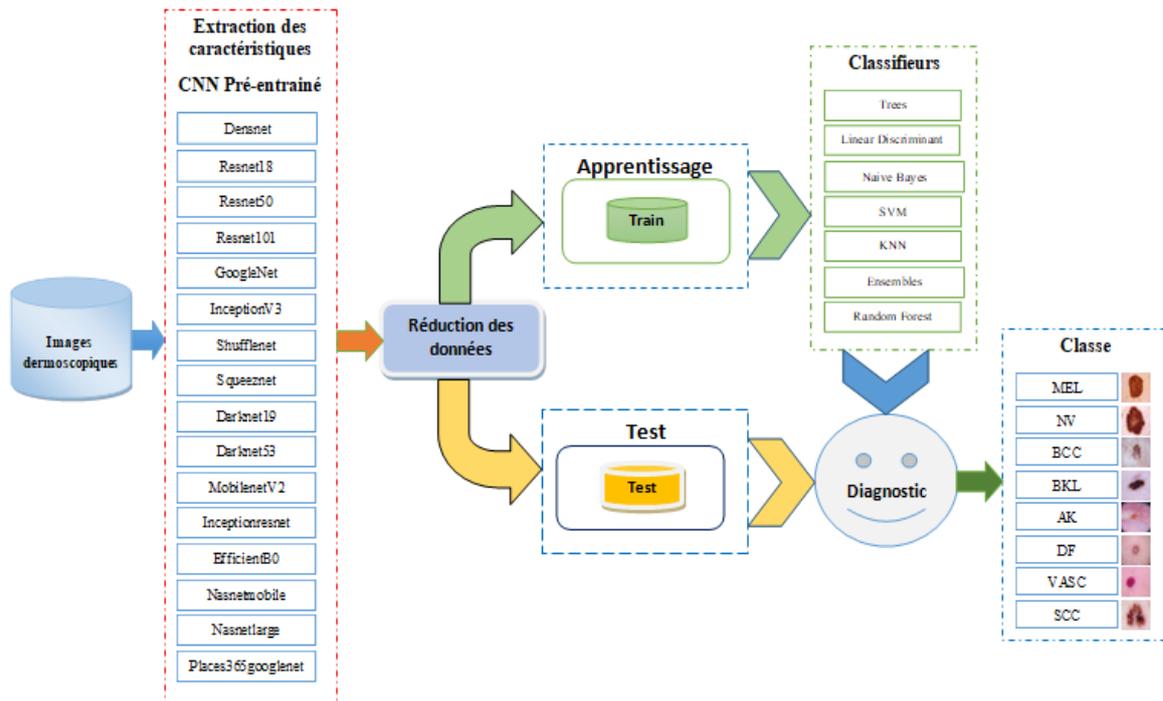


Figure 5.8 : Méthodologie de l'approche réduction des caractéristiques.

Le **Tableau 5.14** indique le nombre de caractéristiques avant et après l'application de l'ACP.

Architecture	Nombre de caractéristiques avant ACP	Nombre de caractéristiques apres ACP
Googlenet	1024	239
InceptionV3	2048	504
ResNet18	512	169
ResNet50	2048	333
ResNet101	2048	318
SqueezeNet	1000	44
Darknet-19	1000	82

Darknet-53	1000	95
InceptionResNet	1536	202
Xception	2048	600
MobileNet	1280	535
DenseNet201	1920	323
Shufflenet	544	175
NASNetMobile	1056	367
NASNetLarge	4032	539
EfficientNetB0	1280	373
Places365 Google net	1024	84

Tableau 5.14. Nombre de caractéristiques avant et après l'application de l'ACP.

Les [Tableau 5.15](#) et [Tableau 5.16](#) présentent les résultats de classification de la base ISIC 2019 après réduction des caractéristiques avec ACP.

	Densenet 201	Resnet18	Resnet50	Resnet101	Shufflenet	Squeeznet	Darknet19	Darknet53
Fine Tree	59,44	57,78	59,69	58,82	57,74	57,74	58,01	58,09
Medium Tree	59,04	56,22	58,72	68,26	56,02	56,02	57,50	55,76
Coarse Tree	56,97	54,80	55,27	55,61	54,11	54,11	53,67	51,16
Linear Discriminant	68,75	66,27	69,21	68,02	66,76	66,76	64,78	65,18
Gaussian Naive Bayes	49,53	39,95	37,35	36,12	42,58	42,58	49,21	47,08
Kernel Naive Bayes	55,41	Fail	46,60	46,49	52,84	52,84	54,30	52,33
Linear SVM	70,59	67,86	71,04	70,35	68,44	68,44	66,30	66,58
Quadratic SVM	75,50	73,35	76,25	75,88	73,51	73,51	71,75	72,13
Cubic SVM	77,79	75,05	77,50	77,77	74,63	74,63	73,00	72,66
Fine Gaussian SVM	65,85	51,70	67,29	67,43	56,28	56,28	72,38	72,68
Medium Gaussian	70,19	72,19	69,68	69,09	70,08	70,08	65,24	66,48
Coarse SVM	52,23	64,45	52,86	52,88	60,66	60,66	53,47	53,26
Fine KNN	76,35	73,79	76,51	76,15	72,09	72,09	71,22	70,90
Medium KNN	61,78	67,69	68,04	68,26	65,87	65,87	67,07	65,12
Coarse KNN	52,03	62,65	58,17	58,98	56,91	56,91	61,47	60,24
Cosine KNN	72,92	68,14	72,76	71,28	70,47	70,47	68,36	69,01
Cubic KNN	61,63	67,55	67,41	67,07	65,10	65,10	67,00	65,04
Weighted KNN	64,86	71,36	72,23	Fail	69,50	69,50	71,12	69,46
Boosted Trees	59,69	59,12	60,40	59,59	57,92	57,92	58,43	56,26

Bagged Trees	61,31	64,13	62,22	62,40	63,07	63,07	63,62	62,95
Subspace Discriminant	68,71	Fail	69,27	68,04	66,90	66,90	64,77	65,24
Subspace KNN	75,80	74,85	74,63	74,06	73,06	73,06	70,00	70,57
RUSBoosted Trees	42,06	42,20	40,43	34,58	29,02	29,02	37,74	35,49
Random forest	59,44	57,78	59,69	58,82	57,74	57,74	58,01	58,09

Tableau 5.15. Résultat d'évaluation par l'utilisation de 8 architectures avec réduction de la base (%)

	Googlenet	MobileNet v2	Xception	Nasnetmobile	Inceptionv3	Efficient b0	Inception Resnet	Nasnet Large	Places365 Googlenet
Fine Tree	57,46	57,46	56,63	56,20	57,64	58,01	56,22	56,30	41,30
Medium Tree	56,83	55,94	55,74	55,53	57,78	57,24	54,07	55,82	42,70
Coarse Tree	53,47	53,53	52,94	52,25	54,60	53,65	51,16	53,24	46,00
Linear Discriminant	65,22	67,04	67,55	64,13	66,60	68,71	66,03	65,57	34,00
Gaussian Naive Bayes	33,81	40,09	35,98	33,97	35,61	34,41	47,71	30,06	Fail
Kernel Naive Bayes	43,96	51,82	46,66	40,21	39,91	43,13	53,81	34,52	49,30
Linear SVM	67,17	68,50	68,57	65,48	67,94	70,13	68,28	66,92	33,90
Quadratic SVM	72,05	72,25	72,35	69,31	71,40	75,48	73,19	70,59	29,80
Cubic SVM	73,43	73,90	74,24	70,81	72,78	77,42	74,34	72,03	28,80
Fine Gaussian SVM	61,37	51,11	53,61	53,30	55,86	58,07	69,50	65,99	46,70
Medium Gaussian	67,92	69,40	68,67	66,21	67,41	70,35	67,49	65,91	32,10
Coarse SVM	56,12	59,20	52,13	54,62	51,82	56,83	52,47	51,44	38,10
Fine KNN	70,96	71,89	71,10	65,55	69,29	74,50	72,60	68,77	33,40
Medium KNN	64,47	60,42	62,87	60,98	61,86	65,71	65,95	96,51	35,80
Coarse KNN	57,84	51,60	56,47	57,76	58,94	57,48	58,33	58,03	37,60
Cosine KNN	68,54	70,67	70,19	66,09	69,40	71,46	69,31	67,63	35,20
Cubic KNN	63,98	60,24	60,24	60,32	61,09	65,18	62,93	62,02	36,10
Weighted KNN	67,84	63,01	66,56	63,98	65,85	69,90	69,21	65,57	31,70
Boosted Trees	59,06	57,50	57,6	57,34	58,88	58,72	56,73	57,58	41,60
Bagged Trees	61,51	57,15	57,94	56,47	56,55	60,92	60,78	58,74	34,50
Subspace Discriminant	65,26	67,05	67,55	64,11	66,70	68,63	66,19	65,61	33,80
Subspace KNN	69,78	74,34	64,19	64,69	70,81	73,83	71,61	66,86	32,40
RUSBoosted Trees	36,93	35,67	36,48	36,40	29,21	37,41	37,15	34,25	61,10
Random forest	57,46	57,46	56,63	56,20	57,64	58,01	56,22	56,30	41,30

Tableau 5.16. Résultat d'évaluation par l'utilisation de 9 architectures avec réduction de la base (%)

La **Figure 5.9** illustre sous forme graphique les différents taux d'exactitude de classification obtenus à partir la base ISIC 2019 par l'utilisation de l'architecture DenseNet201 avec réduction des caractéristiques.

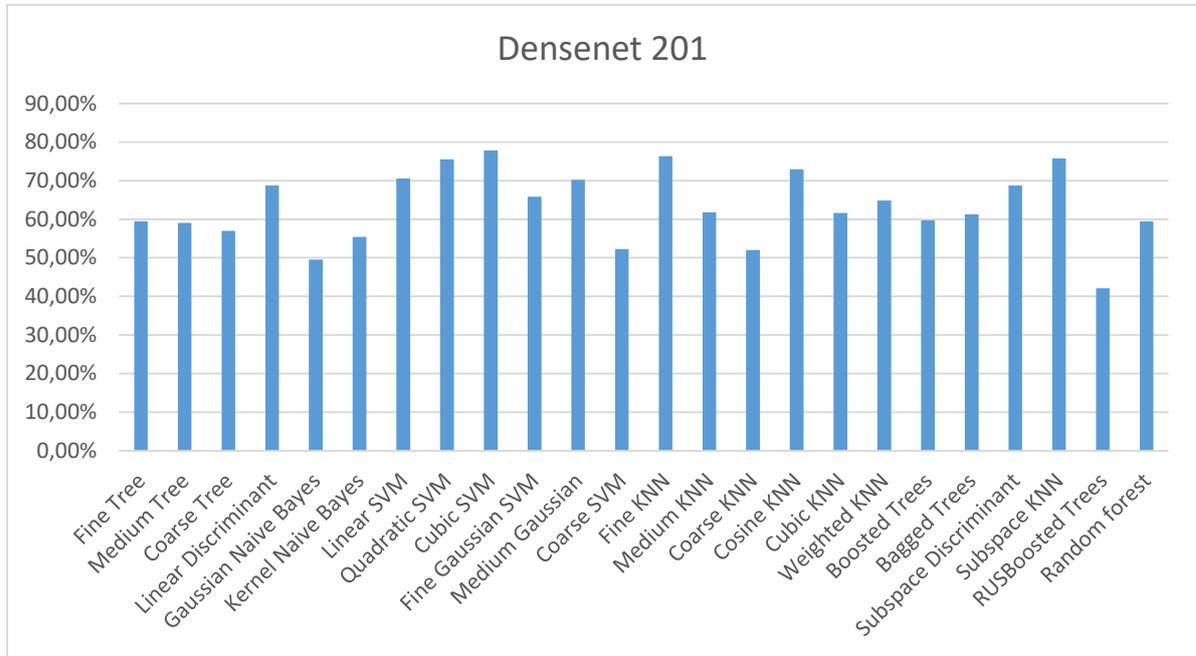


Figure 5.9. Taux de classification par DenseNet201 après réduction des caractéristiques.

5.3. Comparaison

Le **Tableau 5.17** présente les résultats des différentes métriques issues de l'utilisation de plusieurs classifieurs sur l'architecture DenseNet201 avec augmentation de données.

	Précision	Sensibilité	Spécificité	F1-score	Exactitude balancée	Exactitude
Fine Tree	39,38	39,83	91,18	37,8	39,83	41,11
Medium Tree	36,6	34,13	90,01	33,28	34,13	35,46
Coarse Tree	11,38	25,24	89,5	14,94	25,24	26,72
Linear Discriminant	71,61	76,38	94,83	73,12	76,38	76,29
Gaussian Naive Bayes	41,95	43,93	91,53	42,56	43,93	44,61
Kernel Naive Bayes	43,59	39,55	90,91	39,09	39,55	41,3
Linear SVM	72,24	77,89	94,72	73,62	77,89	77,74

Quadratic SVM	81,99	88,86	96,12	84,09	88,86	88,52
Cubic SVM	84,82	92,04	96,4	86,82	92,04	91,71
Fine Gaussian SVM	66,92	49,6	91,17	46,65	49,6	51,29
Medium Gaussian	84,18	91,82	96,26	86,02	91,82	91,56
Coarse SVM	31,33	76,69	80,75	44,49	86,93	86,69
Fine KNN	85,22	92,75	96,38	86,96	92,75	92,34
Medium KNN	19,43	76,52	69,29	30,99	43,63	45,53
Coarse KNN	34,38	27,01	88,51	18,31	27,01	29,12
Cosine KNN	75,4	81,43	95,8	77,31	81,43	80,97
Cubic KNN	72,11	77,11	94,89	73,58	77,11	76,96
Weighted KNN	74,47	60,74	92,77	58,11	60,74	62,43
Boosted Trees	37,89	38,75	91,08	37,93	38,75	39,96
Bagged Trees	68,65	73,15	94,85	70,34	73,15	73,04
Subspace Discriminant	68,79	73,18	94,46	70,18	73,18	73,22
Subspace KNN	69,82	74,53	94,54	71,2	74,53	74,4
RUSBoosted Trees	36,68	37,92	90,81	36,92	37,92	38,23

Tableau 5.17. Résultat d'évaluation par plusieurs classifieurs sur l'architecture DenseNet201 (%)

Le **Figure 5.10** présente les résultats des différentes métriques issues de l'utilisation de plusieurs classifieurs sur l'architecture DenseNet201 avec augmentation de données.

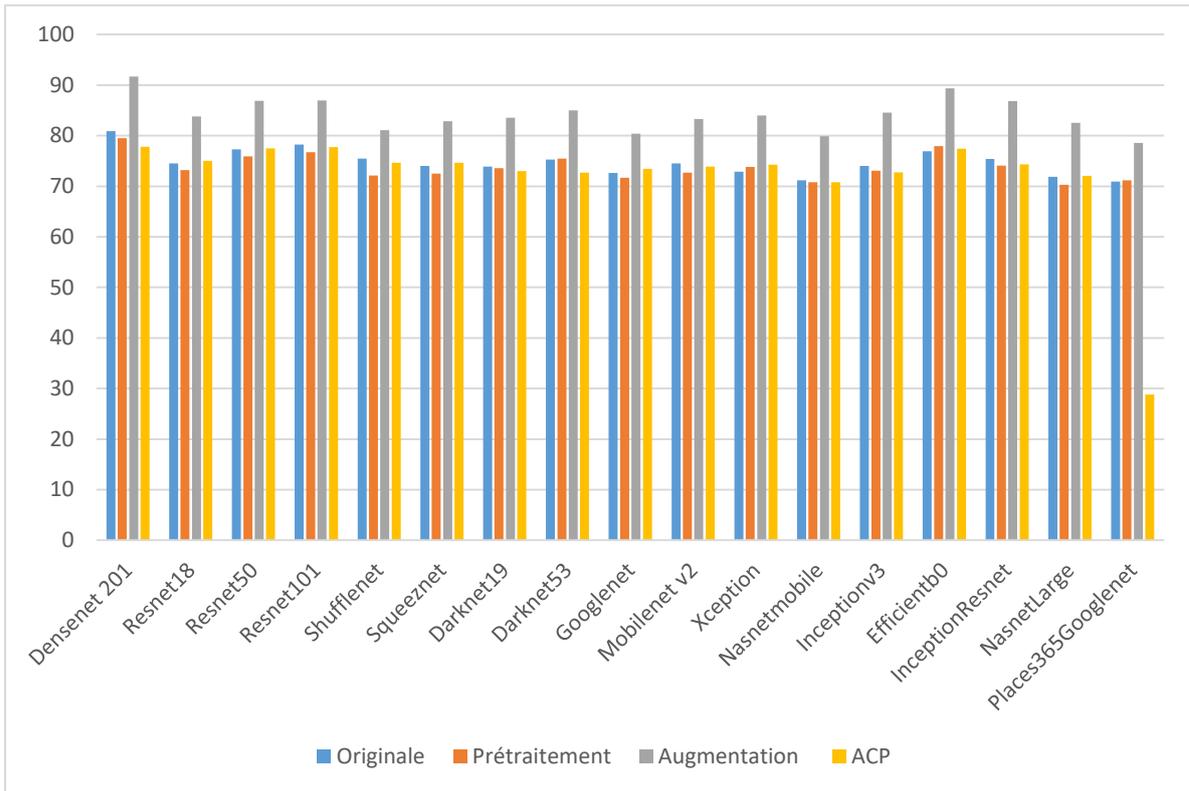


Figure 5.10. Taux de classification par DenseNet201 avec Cubic SVM (%).

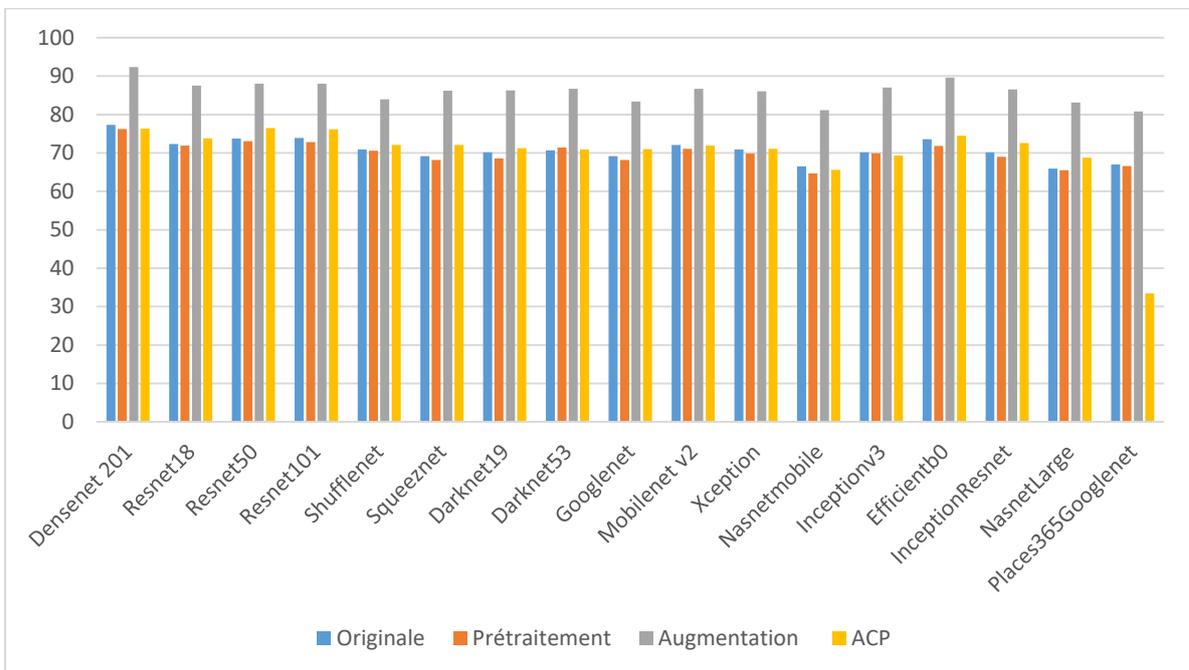


Figure 5.11. Taux de classification par DenseNet201 Fine KNN(%).

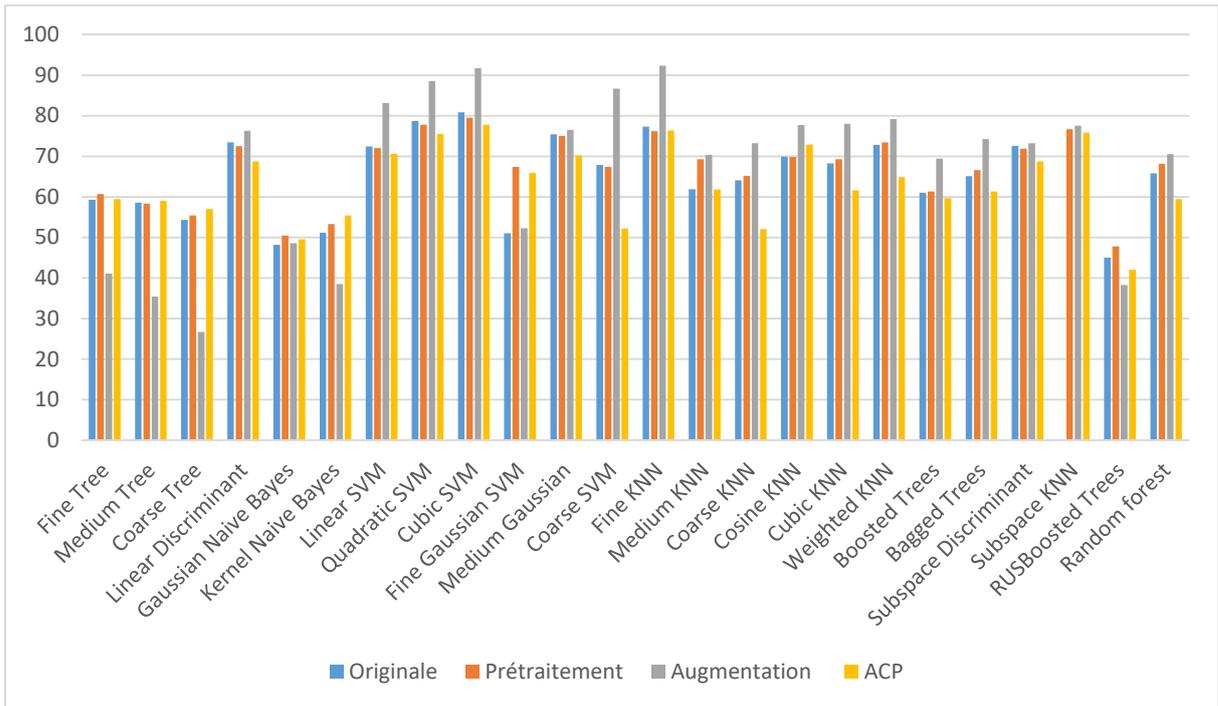


Figure 5.12. Taux de classification de DenseNet201 avec différent classifieurs (%).

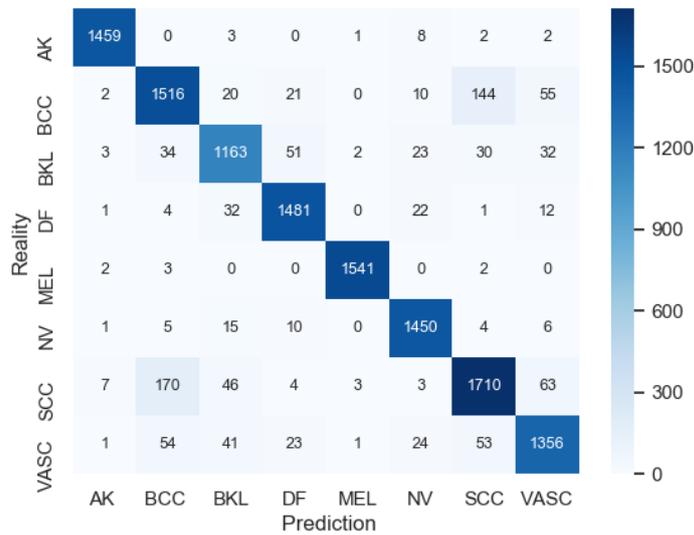


Figure 5.13. Matrice de confusion de l'architecture DenseNet201 avec Cubic SVM.

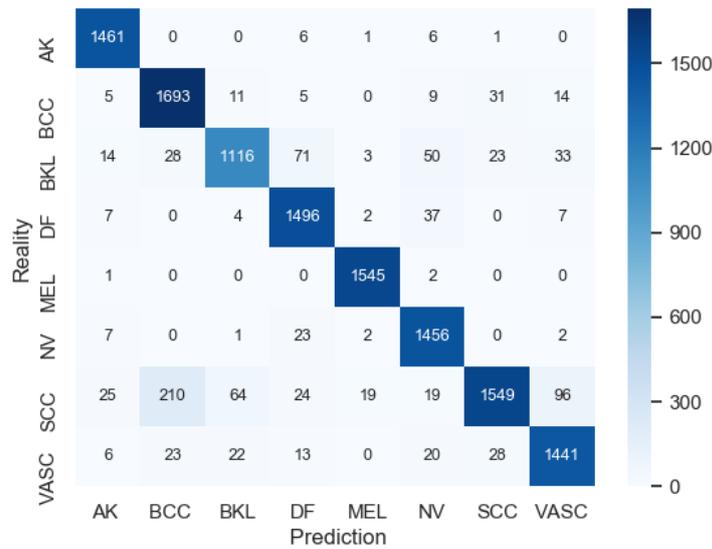


Figure 5.14. Matrice de confusion de l'architecture DenseNet201 avec Fine KNN.

En comparant les résultats des expériences présentées dans les [Tableaux 5.6, 5.7, 5.8, 5.9, 5.13, 5.13, 5.15, 5.16](#) et les [Figures 5.10, 5.11, 5.12](#). Dans l'ensemble, les performances de classification des différents types de lésions cutanées sont améliorées lors de l'application de données d'augmentation dans divers classificateurs et architectures CNN.

L'architecture Densenet201 présente la plus grande exactitude par rapport aux autres architectures utilisées ; et SVM cubique et Fine KNN représentent les meilleurs classificateurs lorsque nous le comparons avec les autres classificateurs. Cependant, la combinaison de Densenet201 en tant qu'extracteur de caractéristiques et de SVM cubique atteint un taux d'exactitude de 91,71 % lors de la combinaison de Densenet201 en tant qu'extracteur de caractéristiques avec le classificateur FineKNN atteint un taux d'exactitude de 92,34 % en utilisant l'augmentation de données, D'autre part en combinant Densenet201 et cubique SVM atteint les meilleurs taux d'exactitude de 80,87%, 79,50%, 77,79% dans l'ensemble original et prétraité et avec ACP respectivement.

Pour juger notre proposition et discuter ses caractéristiques par rapport à l'état de l'art, nous avons examiné différents travaux de l'état de l'art présentés dans le [Tableau 5.18](#) qui adresse le même problème de classification avec le même ensemble de données ISIC 2019, d'après les performances obtenues notre proposition est fiable et efficace par rapport aux autres travaux.

Auteur	Méthode	Nb de classes	Exactitude
Alizadeh & Mahloojifar , 2021	CNN, texture features	2	96.30%
El-khatib et al., 2020	Decision fusion, features based CNN, SVM	2	93.00%
Kassem et al., 2020	GoogleNet+ SVM	8	94.92%
Pacheco et al., 2019	Ensemble of 13 CNNs	8	89%
Ahmed et al., 2020	Ensembles (Xception, Inception- ResNet-V2, and NasNetLarge)	8	90.6%
Guissois , 2019	Ensemble models (DenseNet201 and Inception V3)	8	91.00%
Méthode proposée	DenseNet201, Cubic SVM	8	91.71%
Méthode proposée	DenseNet201, Fine KNN	8	92.34%

Tableau 5.18. Résultat d'évaluation de notre approche avec l'état de l'art.

5.4. Approche pipeline en deux étapes avec la base PH2¹¹

Dans cette partie portant sur un diagnostic de mélanome, deux scénarios d'expériences sont proposés et illustrés par la [Figure 5.15](#) en utilisant le jeu de données PH2 :

- Classification des lésions cutanées en deux types de lésions : mélanome ou non mélanome
- Classification des lésions cutanées en trois types : mélanome, naevus atypique ou naevus commun

¹¹ Samia Benyahia, Boudjelal Meftah, Olivier Lézoray, Skin lesion classification using convolutional neural networks based on Multi-Features Extraction, The 19th International Conference on Computer Analysis of Images and Patterns (CAIP 2021), Cyprus, Grèce, 27 Septembre-01 Octobre 2021

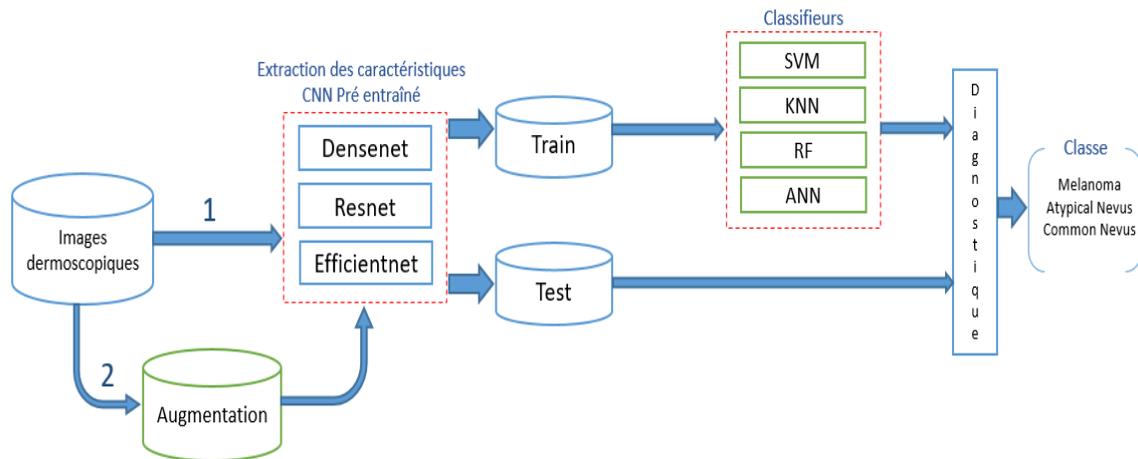


Figure 5.15. Méthodologie de l’approche pipeline en deux étapes avec la base PH2

La base de données PH2 contient un total de 200 images que nous avons divisé en deux parties.

- 80% du jeu de données (160 images) est utilisé pour l’apprentissage
- 20% du jeu de données (40 images) est utilisé pour tester le modèle.

Pour le deuxième scénario concernant la classification en trois classes, une technique d'augmentation des données est incluse avec les opérations suivantes : rotations, zoom, cisaillement, retournements (haut-bas, gauche-droite), biais-gauche-droite, amélioration du contraste. Toutes les expériences ont été réalisées avec le même ensemble de données d'apprentissage et testées avec le même ensemble de tests.

Le **Tableau 5.19** et la **Figure 5.16** décrivent les résultats de performance obtenus pour classer l'ensemble de données PH2 en deux classes : mélanome ou non-mélanome basant sur les architectures DenseNet201, ResNet50, EfficientB0 et quatre classifieurs : ANN, SVM, KNN et Random Forest.

	DenseNet201				ResNet50				EfficientB0			
	Acc	Sen	Spec	Précision	Acc	Sen	Spec	Précision	Acc	Sen	Spec	Précision
SVM	99	99	99	99	97.5	83.33	83.33	98.68	92.5	65.31	65.31	72.36
KNN	97.5	98.64	98.64	87.5	95	81.98	91.98	91.98	97.5	98.64	98.64	87.5
ANN	95	96.96	96.96	88.88	95	97.36	97.36	75	87.12	78.12	78.12	81.16

Random Forest	99	99	99	99	97.5	83.33	83.33	98.68	95	97.29	97.29	80
---------------	----	----	----	----	------	-------	-------	-------	----	-------	-------	----

Tableau 5.19. Résultats d'évaluation du mélanome et non-mélanome

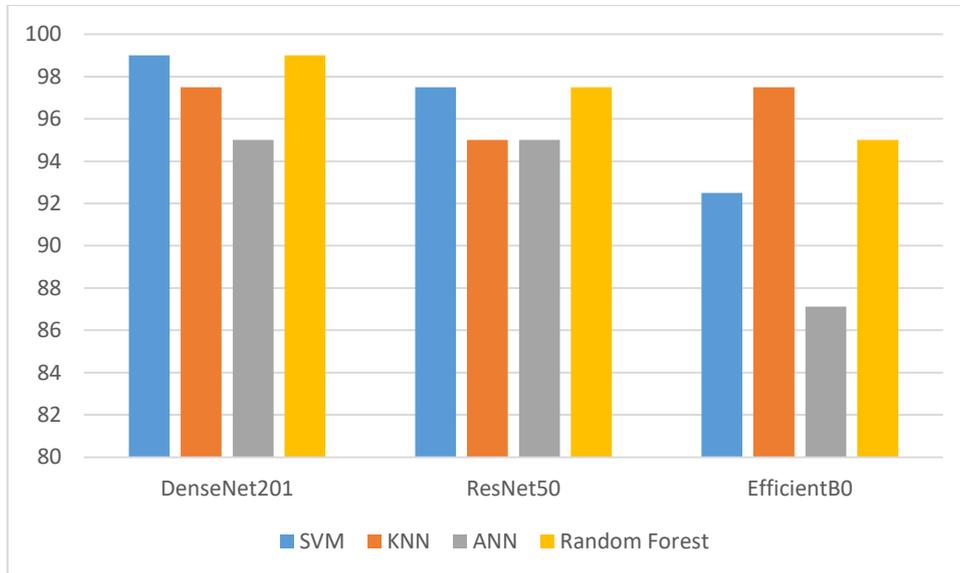


Figure 5.16. Précision selon DenseNet201, ResNet50 et EfficientB0 avec quatre classifieurs pour mélanome ou non-mélanome.

Le **Tableau 5.20** et la **Figure 5.17** décrivent les résultats de performance obtenus pour classer l'ensemble de données PH2 en 3 classes : mélanome, naevus atypique ou naevus commun sans augmentation basant sur les architectures DenseNet201, ResNet50, EfficientB0 et quatre classifieurs : ANN, SVM, KNN et Random Forest.

	DenseNet201				ResNet50				EfficientB0			
	Acc	Sen	Spec	Précision	Ac	Sen	Spec	Précision	Acc	Sen	Spec	Précision
SVM	62	62.77	78.38	73.19	60	60.20	77.16	71.44	62.5	40.09	76.82	60.45
KNN	70	70.37	81.48	75.91	60	57.06	75.59	61.11	62.5	41.14	76.2	62.19
ANN	70	72.80	83.33	72.87	60	28.57	74.04	34.43	50	30.51	72.17	57.44
Random Forest	57.5	58.69	75.3	69.59	70	44.75	82.09	76.5	70	47.78	82.38	67.35

Tableau 5.20. Résultats d'évaluation du mélanome, naevus atypique et naevus commun sans augmentation.

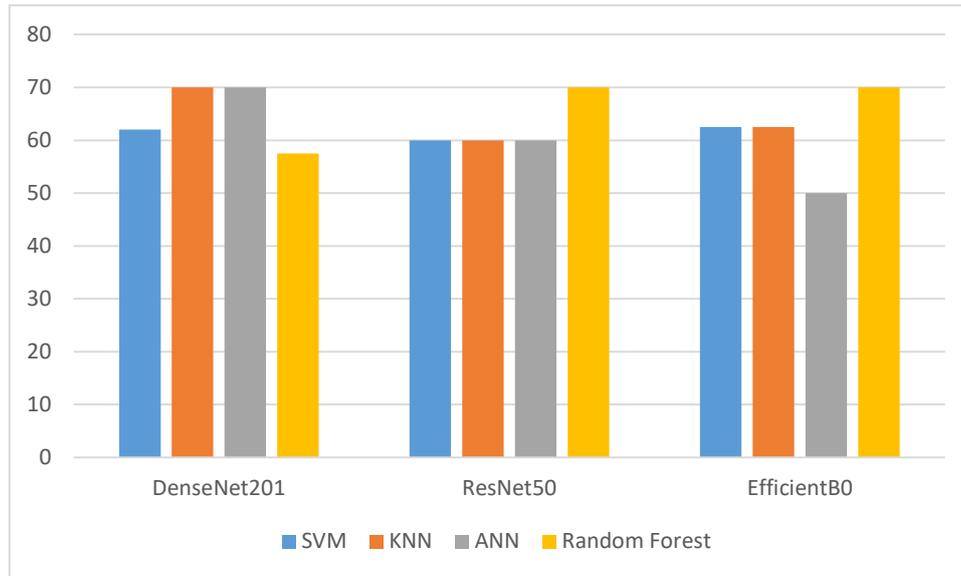


Figure 5.17. Précision selon DenseNet201, ResNet50 et EfficientB0 avec quatre classificateurs pour le mélanome, le naevus atypique ou le naevus commun sans augmentation.

Le **Tableau 5.21** et la **Figure 5.18** illustrent les résultats de performance obtenus pour classer l'ensemble de données PH2 en 3 classes : mélanome, naevus atypique ou naevus commun avec augmentation basant sur les architectures DenseNet201, ResNet50, EfficientB0 et quatre classifieurs : ANN, SVM, KNN et Random Forest.

	DenseNet201				ResNet50				EfficientB0			
	Acc	Sen	Spec	Précision	Acc	Sen	Spec	Précision	Acc	Sen	Spec	Précision
SVM	95	96.67	97.22	96.29	87.5	91.25	92.77	91.02	90	92.91	94.16	92.59
KNN	87.5	91.25	93.51	86.31	82.5	80	90.37	77.14	82.50	87.50	89.72	87.30
ANN	92.5	87.77	93.88	92.50	95	96.29	96	96.66	85	88.88	90.90	89.16
Random Forest	85	88.75	91.57	83.57	82.5	87.50	90.92	79.39	82.50	87.50	90.04	82.47

Tableau 5.21. Résultats d'évaluation du mélanome, naevus atypique et naevus commun avec augmentation.

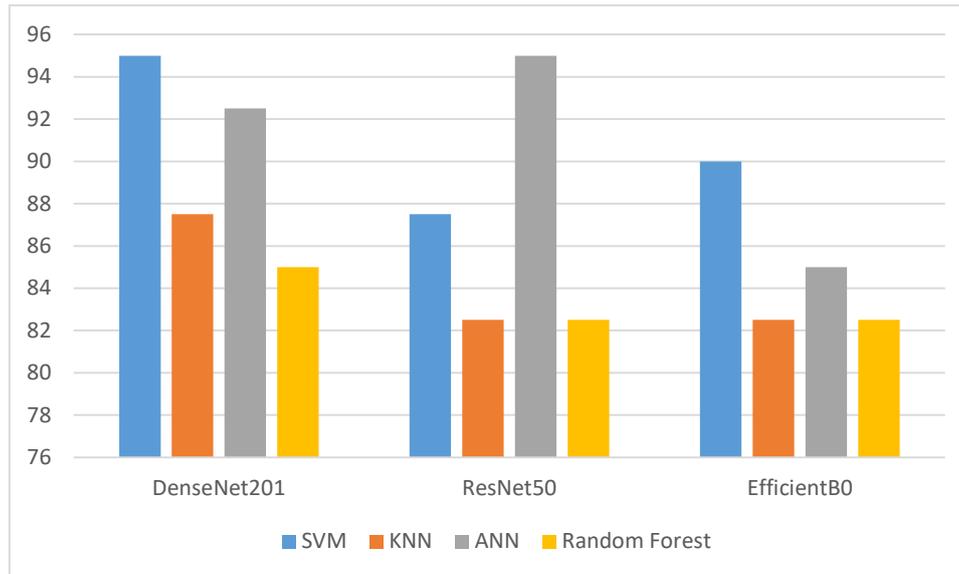


Figure 5.18. Précision selon DenseNet201, Resnet50 et Efficient 0 avec quatre classificateurs pour le mélanome, le naevus atypique ou le naevus commun avec augmentation.

A travers l'évaluation et les résultats obtenus à partir des [Tableau 5.19](#), [Tableau 5.20](#) et [Tableau 5.21](#), nous constatons que le modèle DenseNet201 a obtenu les meilleurs résultats par rapport aux autres modèles CNN. De l'autre côté, le classifieur cubique SVM a obtenu les meilleurs résultats par rapport aux autres classifieurs ANN, KNN, Random Forest.

À travers les [Figure 5.16](#), [Figure 5.17](#) et [Figure 5.18](#), nous remarquons que l'augmentation des données améliore considérablement les résultats de classification pour trois classes.

D'après les [Tableau 5.19](#), [Tableau 5.20](#) et [Tableau 5.21](#), les statistiques montrent que les classifieurs SVM offrent la plus grande précision de 99 % pour la détection des mélanomes et des non-mélanomes, tandis que 95 % pour le mélanome, le naevus atypique ou le naevus commun avec augmentation combinée à l'architecture DenseNet.

Les [Tableaux 5.22](#) et [5.23](#) décrivent la comparaison de l'approche proposée avec l'état d'art en termes de précision pour l'ensemble de données PH2. On peut constater que notre approche proposée est compétitive par rapport aux autres méthodes.

Auteurs	Nb de classes	Classifieur	Taux
Ghasem et al., 2020	2	SBS	98.5%
Ghasem et al., 2020	2	HSBS	96.7%
Salido & Ruiz, 2018	2	AlexNet	93%
Filali et al., 2020	2	SVM	98%
Sanket & Chandra, 2019	2	SVM	92%
Méthode proposée	2	DenseNet201 +SVM	99%

Tableau 5.22. Comparaison d'évaluation de 2 classes avec l'état d'art

Auteurs	Nb de classes	Classifieur	Taux
Oekan & Koklu, 2017	3	MLP	92.5%
Salido & Ruiz, 2018	3	AlexNet	67.5%
Singh et al., 2020	3	SVM	92.5%
Khalid et al., 2018	3	AlexNet	98.61%
Méthode proposée	3	DenseNet201 +SVM	95%

Tableau 5.23. Comparaison d'évaluation de 3 classes avec l'état d'art

En prenant en compte les expériences réalisées précédemment sur le jeu de données ISIC 2019, nous avons appliqué plusieurs classifieurs sur la base des caractéristiques obtenues par le modèle DenseNet201 pré-entraîné. Le [Tableau 5.24](#) présente l'évaluation en termes d'exactitude.

Méthode	Exactitude
Fine tree	90.00%
Medium tree	90.00%
Coarse tree	90.00%
Linear Discriminant	98.00%
Gaussian Naive Bayes	80.00%
Kernel Naive Bayes	88.00%
Linear SVM	97.00%
Quadratic SVM	99.00%
Cubic SVM	99.00%
Fine Gaussian SVM	90.00%
Medium Gaussian SVM	95.00%
Coarse SVM	90.00%

Fine KNN	95.00%
Medium KNN	88.00%
Coarse KNN	95.00%
Cosine KNN	90.00%
Cubic KNN	90.00%
Weighted KNN	88.00%
Boosted trees	90.00%
Bagged trees	83.00%
Subspace discriminant	93.00%
Subspace KNN	93.00%
RUSBoostedtrees	88.00%

Tableau 5.24. Résumé des résultats sur PH2

5.5. Approche pipeline de bout en bout avec augmentation de données avec la base ISIC 2019

L'utilisation d'un réseau de neurones convolutionnel déjà conçu et entraîné nous permet de tirer parti de ce que le modèle a déjà appris sans avoir à le développer à partir de zéro.

Plusieurs expériences ont été effectuées par l'utilisation de multiple CNN pré-entraînés. Nous avons utilisé une architecture de réseau d'apprentissage profond existante Densenet201 qui a été mise en œuvre avec des poids d'apprentissage par transfert sur la base ImageNet pour être appliquée sur la base ISIC 2019. Notons que ImageNet consiste en un ensemble de données d'environ 1,2 million d'images pour l'apprentissage, 50000 pour la validation et 100000 pour les tests, appartenant à 1000 catégories.

La [Figure 5.19](#) montre l'architecture CNN DenseNet201 utilisé. Deux couches entièrement connectée ont ensuite été ajoutée avec 4096 poids suivi d'une fonction activation « Elu », une couche drop out et une fonction d'activation Softmax. Les poids des deux couches entièrement connectée ont été initialisés de manière aléatoire et pré-entraînés sur les images de l'ensemble d'apprentissage. Au cours de l'apprentissage, une métrique de perte d'entropie croisée a été définie avec un optimiseur Adam (Learning Rate= e^{-5}) et une taille de lot de 16. Au cours de l'apprentissage du modèle, l'augmentation des données a été effectuée en introduisant une quantité aléatoire d'images par zoom, rotation et réflexion horizontale et verticale.

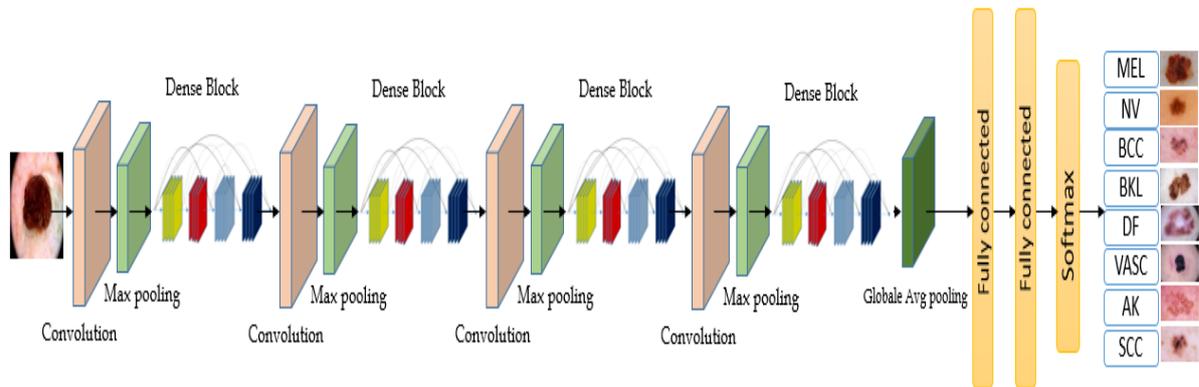


Figure 5.19 : Architecture DenseNet201 affiné de bout en bout

D'après les résultats obtenus à partir de cette expérience, nous pouvons voir que taux de classification obtenue est 91.10% (Tableau 5.25).

L'objectif de cette expérience est de montrer la différence des résultats obtenus de l'approche bout en bout par rapport à l'approche pipeline en deux étapes (extraction de caractéristiques avec apprentissage profond + classification avec apprentissage automatique conventionnel).

La Figure 5.20 montre la matrice de confusion de l'architecture DenseNet201 affiné de bout en bout avec augmentation de données

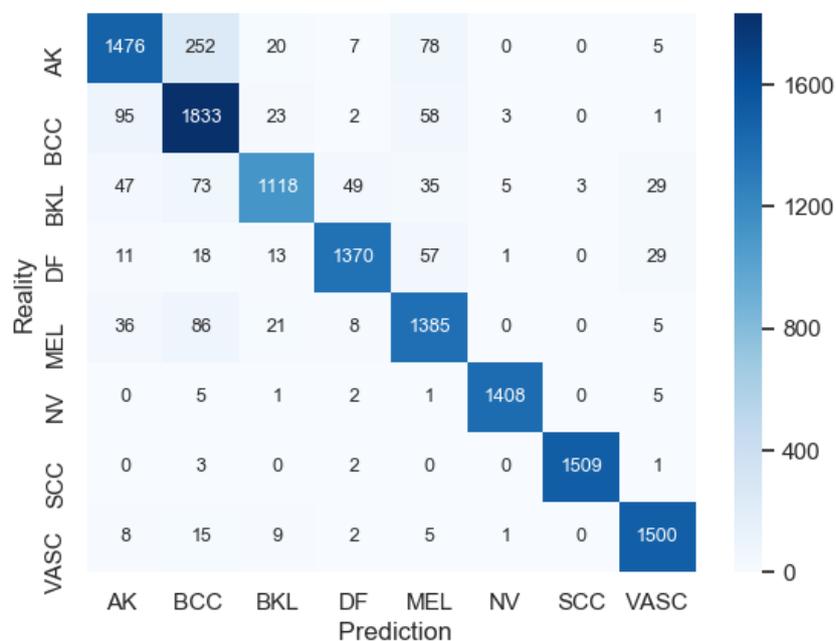


Figure 5.20. Matrice de confusion de l'architecture DenseNet201 affiné de bout en bout avec augmentation de données

Métrique Méthode	Précision	Recall	Spécificité	F1-score	Balanced Accuracy	Accuracy
DenseNet201	88.22%	80.30%	98.19%	84.07%	91.33%	91.10%

Tableau 5.25. Résultat d'évaluation du modèle DenseNet201 affiné de bout en bout avec augmentation de données

5.6. Approche pipeline hiérarchique¹²

Les étapes globales de notre modèle proposé sont illustrées dans la [Figure 5.21](#). Dans la première étape, nous divisons l'ensemble de données d'images dermoscopiques en deux parties : la première représente les données d'entraînement et la seconde représente les données de test. La phase d'entraînement est composée de cinq CNN indépendants organisés en arbre hiérarchique, et chacun effectue individuellement une décision pour un type de lésion cutanée spécialisé et spécifique.

Le premier CNN est utilisé pour distinguer les lésions mélanocytaires et non mélanocytaires. Les lésions mélanocytaires identifiées sont injectées dans le deuxième CNN pour faire la distinction entre les mélanocytes malins (mélanome) et mélanocytaires bénins (naevus). Au contraire, les lésions non mélanocytaires identifiées sont transmises au troisième CNN pour faire la distinction entre les lésions malignes non mélanocytaires et les lésions bénignes non mélanocytaires. Le quatrième CNN reçoit des lésions malignes non mélanocytaires en entrée et distingue le carcinome épidermoïde (SCC) du carcinome basal (BCC).

Enfin, le cinquième CNN acquiert le bénin non mélanocytaire en entrée et distingue le dermatofibrome (DF), la lésion vasculaire (VASC), la kératose actinique (AK), la kératose bénigne (BKL). Une fois la phase d'apprentissage terminée, nous évaluons le modèle avec un jeu de données de test.

¹² Samia Benyahia, Boudjelal Meftah, Olivier Lézoray. Hierarchical Approach for the Classification of Multi-class Skin lesions Based on Deep Convolutional Neural Networks. Third International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI 2022), Paris, France, 1-3 Juin, 2022

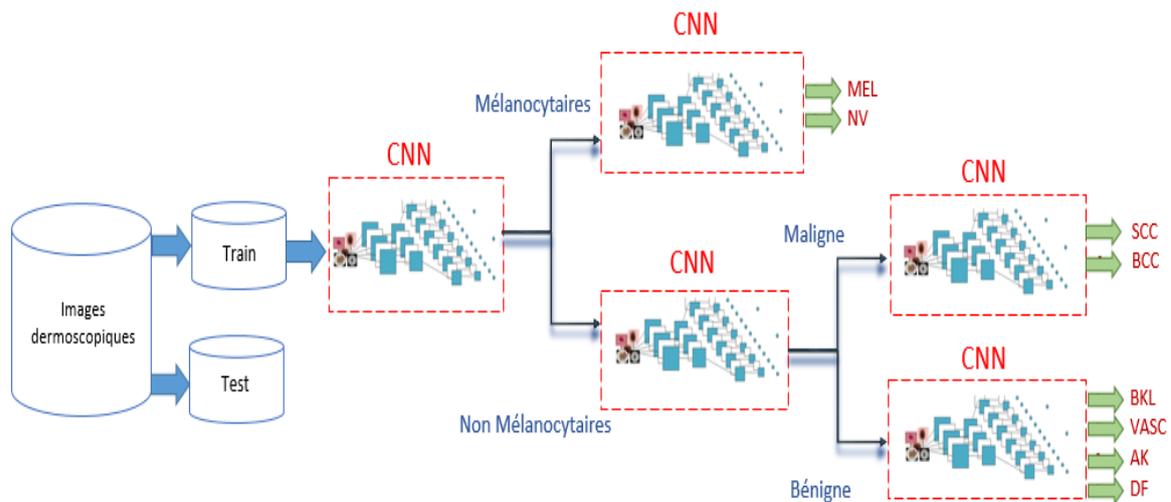


Figure 5.21. Organigramme de la méthode proposée

Plusieurs expériences ont été menées avec le jeu de données ISIC 2019. L'ensemble de données a été divisé en trois parties distinctes : 80%, 10% et 10%, respectivement pour les ensembles de données d'entraînement, de validation et de test. Nous précisons qu'aucun prétraitement des données n'a été effectué.

Nous avons mené des expériences avec deux scénarios : les approches de classification séquentielle et hiérarchique. Chaque scénario est testé sans augmentation de données avec deux CNN : DenseNet201 et ResNet50. Pour l'approche séquentielle, nous utilisons directement le CNN pour effectuer la classification de bout en bout. Dans l'approche hiérarchique, nous formons cinq CNN où chacun effectue individuellement une classification pour un type spécifique de lésion cutanée de manière hiérarchiquement structurée telle que :

- Le premier CNN est utilisé pour distinguer les mélanocytaires des non mélanocytaires
- Pour les mélanocytaires, le deuxième CNN est utilisé pour faire la distinction entre le mélanome et le naevus
- Pour les non mélanocytaires, le troisième CNN est utilisé pour faire la distinction entre malin non mélanocytaire et bénin non mélanocytaire
- Pour les malins non mélanocytaires, le quatrième CNN est utilisé pour faire la distinction entre le carcinome épidermoïde et le carcinome basal
- Pour les bénignes non mélanocytaires, le cinquième CNN est utilisé pour distinguer le dermatofibrome, la lésion vasculaire, la kératose actinique et la kératose bénigne.

Le **Tableau 5.26** décrit les résultats de performance obtenus pour classer l'ensemble de données ISIC 2019 en huit classes avec ResNet50 et DenseNet201 pour les architectures séquentielles et hiérarchiques sans augmentation. À travers les résultats obtenus à partir du **Tableau 5.26**, nous pouvons observer que le modèle hiérarchique proposé obtient les meilleurs résultats par rapport au modèle séquentiel pour les deux architectures DenseNet201 et ResNet50. De l'autre côté, nous constatons que le modèle DenseNet201 fournit les meilleurs résultats par rapport à ResNet50.

Métrique	ResNet 50 Séquentiel	ResNet 50 Hiérarchique	DenseNet201 Séquentiel	DenseNet201 Hiérarchique
Accuracy	64.03%	77.69%	66.69%	79.43%
Sensitivity	37.33%	76.6%	38.21%	76.37%
Specifcity	94.32%	95.83%	94.40%	95.89%
Precision	35.14%	66.35%	35.86%	69.14%

Tableau 5.26. Synthèse des résultats obtenus avec Resnet séquentiel et hiérarchique50 etArchitectures DenseNet201

Le **Tableau 5.27** présente les différentes augmentations effectuées sur l'ensemble d'apprentissage.

	ISIC 2019	Train 80%	opérations	Nombre D'images après augmentation
NV	12875	10300	-3000	7300
MEL	4522	3618	*2	7236
BCC	3328	2657	/	2657
SCC	628	502	*6	3012
DF	239	191	*10	1910
VASC	253	203	*10	2030
AK	867	694	*3	2082
BKL	2624	2099	/	2099

Tableau 5.27. Différentes augmentations effectuées sur l'ensemble d'apprentissage.

Le **Tableau 5.28** présente les résultats obtenus pour classer l'ensemble de données ISIC 2019 en huit classes avec ResNet50 et DenseNet201 pour les architectures séquentielles et hiérarchiques avec augmentation.

Métrique	ResNet 50 Séquentiel	ResNet 50 Hiérarchique	DenseNet201 Séquentiel	DenseNet201 Hiérarchique
Accuracy	77.81%	78.91%	74.37%	81.11%
Sensitivity	71.52%	67.43%	73.12%	70.59%
Specifcity	95,84%	95,88%	95.34%	96.06%
Precision	63.43%	61.29%	60.96%	67.70%

Tableau 5.28. Synthèse des résultats obtenus avec ResNet50 séquentiel et hiérarchique et l'architectures DenseNet201 avec augmentation des données.

A travers l'évaluation et aux résultats obtenus à partir des **Tableau 5.27** et **Tableau 5.29**, nous pouvons remarquer que l'augmentation des données améliore la précision de tous les modèles, qu'ils soient séquentiels ou hiérarchiques. Les **Figure 5.22** et **Figure 5.23** présentent une visualisation des métriques de performance obtenues avec DenseNet201 et ResNet50 respectivement pour les approches séquentielles et hiérarchiques.

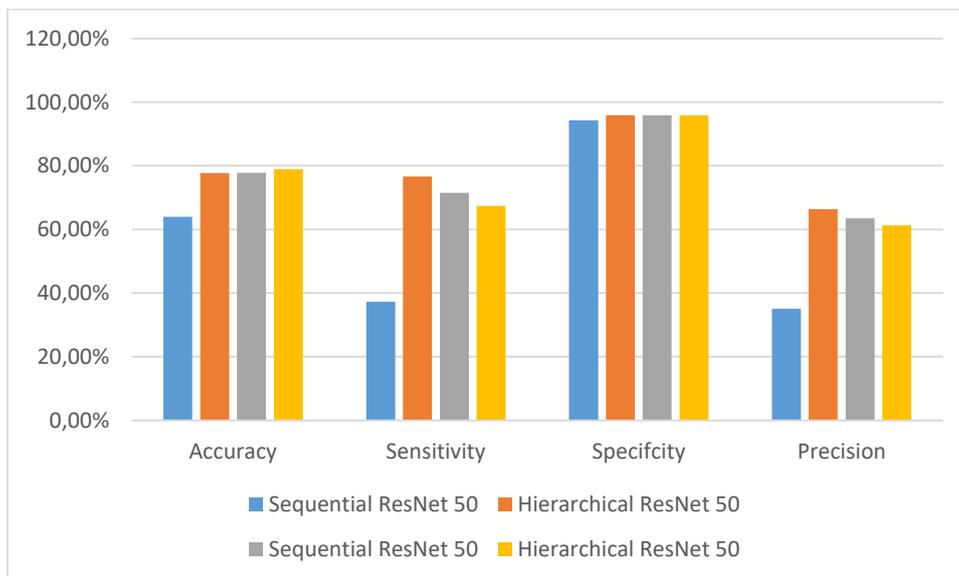


Figure 5.22. Résultats d'évaluation obtenus avec DenseNet201.

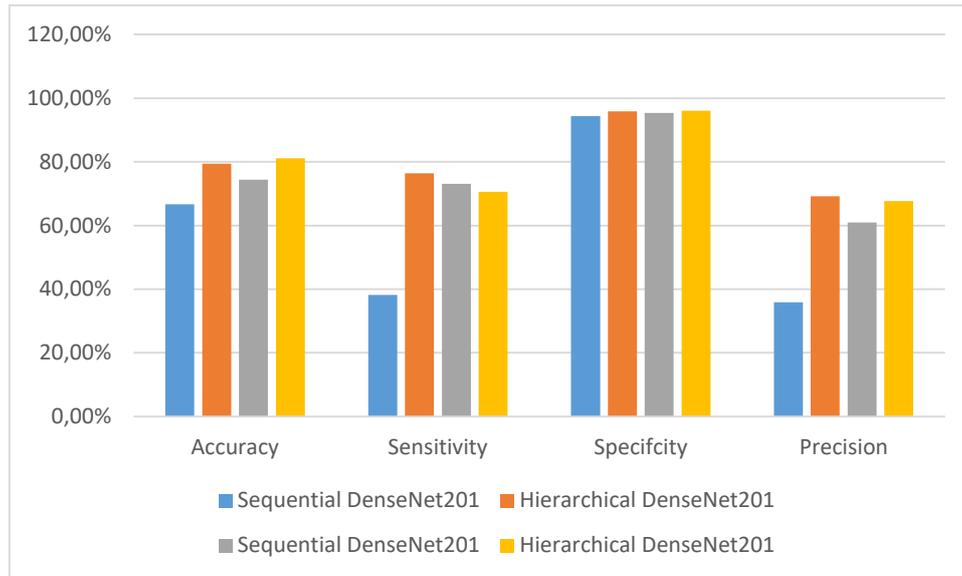


Figure5.23. Résumé des résultats d'évaluation obtenus avec ResNet201.

6. Conclusion

Ce chapitre a dressé l'ensemble des expériences effectuées par l'approche pipeline en deux étapes où nous étudions l'efficacité de l'utilisation des architectures de réseaux de neurones convolutifs (CNN) couramment pré-entraînés comme extracteurs de caractéristiques et de nombreux classificateurs d'apprentissage automatique pour évaluer la classification des lésions cutanées à partir de deux bases de données ISIC 2019 et PH2. D'autre part nous avons effectué des expériences sur l'approche hiérarchique et séquentielle à partir de la base de données ISIC 2019.

Conclusion générale

Le travail présenté tout au long de ce manuscrit a proposé une réponse à la problématique de diagnostic des lésions de la peau en dermatologie, par l'utilisation des réseaux de neurones convolutionnels (CNN). En s'inspirant notamment des avancées sur les méthodes d'apprentissage profond, ce manuscrit a établi une réflexion sur ce sujet permettant la mise en place d'outils d'aide à la détection précoce de pathologies de la peau. L'incidence ne cesse de croître au fil des années, et pour lesquelles la réalisation d'un diagnostic clinique par le praticien peut être ardue selon leurs configurations respectives ou stade d'avancement.

Afin de parvenir à cet objectif, nous avons proposé une présentation des éléments essentiels à la réalisation de ce sujet dont : (i) une explication clinique des principales pathologies de la peau, (ii) une présentation des algorithmes d'intelligence artificielle, (iii) et une description des méthodes d'apprentissage sollicitées dans ce manuscrit pour permettre la mise en place de méthodes de diagnostic. En outre, cette thèse a servi à présenter les caractéristiques de deux bases de données employées pour évaluer les méthodes d'aide au diagnostic par ordinateur proposées (ISIC 2019 et PH2). Par ailleurs, cette présentation des deux bases de données a servi à comprendre les éléments-clés identifiés par les dermatologues sur chacune des modalités afin de caractériser le cancer de la peau.

Enfin, le quatrième et le cinquième chapitre ont permis de répondre à la problématique de ce manuscrit en proposant au lecteur diverses méthodes dont le principe permet d'extraire les caractéristiques des lésions cutanées par les réseaux de neurones convolutionnels (CNN) puis appliqué divers classifieurs issues de l'apprentissage machine. Les expérimentations effectuées sur les bases ISIC 2019 et PH2 ont montré l'efficacité de l'approche pipeline en deux étapes par rapport à l'approche fine tuning de bout en bout.

Notre travail réalisé au cours de notre parcours de doctorat, nous a permis de dégager d'autres issues et perspectives en matière de diagnostic assisté par ordinateur.

- Le premier réflexe est d'appliquer notre approche sur d'autres bases de données médicales tels que : COVID, Chest, Brain, Cell,...etc.
- S'orienter vers la détection de contours et la segmentation des tumeurs cancéreuses de la peau en utilisant la base ISIC 2020 par l'utilisation des nouvelles approches des réseaux de neurones convolutionnels.

Conclusion générale

- Exploiter les métadonnées de la base ISIC 2020 qui correspondent aux informations liées à la pathologie et aux patients, avec notre approche afin de parvenir à un diagnostic meilleur.
- Utiliser l'approche ensembliste (plusieurs CNN) en diagnostic médical.

Références bibliographiques

Abrahams-Gessel, S., & Ernstoff, M.S. (2013). Recent skin self-examination and doctor visits in relation to melanoma risk and tumour depth. *British Journal of Dermatology*, 168.

Adamson, A.S., Jarmul, J.A., & Pignone, M.P. (2019). Screening for Melanoma in Men: a Cost-Effectiveness Analysis. *Journal of General Internal Medicine*, 35, 1175-1181.

Ahmed, S.A., Yanikoglu, B.A., Göksu, Ö., & Aptoula, E. (2020). Skin Lesion Classification With Deep CNN Ensembles. *2020 28th Signal Processing and Communications Applications Conference (SIU)*, 1-4.

Ahn, E., Bi, L., Jung, Y., Kim, J., Li, C., Fulham, M.J., & Feng, D.D. (2015). Automated saliency-based lesion segmentation in dermoscopic images. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 3009-3012.

Alpaydin, E. (2014) *Introduction to Machine Learning*. MIT Press.

Alizadeh, S.M., & Mahloojifar, A. (2021). Automatic skin cancer detection in dermoscopy images by combining convolutional neural networks and texture features. *International Journal of Imaging Systems and Technology*, 31, 695 - 707.

Argenziano, G., Fabbrocini, G., Carli, P., de Giorgi, V., Sammarco, E., & Delfino, M. (1998). Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions. Comparison of the ABCD rule of dermatoscopy and a new 7-point checklist based on pattern analysis. *Archives of dermatology*, 134 12, 1563-70.

Arora, G., Dubey, A.K., Jaffery, Z.A., & Rocha, Á. (2020). Bag of feature and support vector machine based early diagnosis of skin cancer. *Neural Computing and Applications*, 1 - 8.

Asha Deepika, P.V., B.C.P.T.V.R.A. Yamini, 2020. Classification of skin lesions using svm via deep learning feature network. *Int. J. Adv. Sci. Technol.* 29, 4526–4538.

Avila, S. (2020). Data, depth, and design: Learning reliable models for skin lesion analysis. *Neurocomputing*, 383, 303-313.

Barata, C., Celebi, M.E., & Marques, J.S. (2019). A Survey of Feature Extraction in Dermoscopy Image Analysis of Skin Cancer. *IEEE Journal of Biomedical and Health Informatics*, 23, 1096-1109.

Bengio, Y. (2012). Practical Recommendations for Gradient-Based Training of Deep Architectures. *Neural Networks: Tricks of the Trade*.

Bengio, Y. (2012). Deep Learning of Representations for Unsupervised and Transfer Learning. In *ICML Unsupervised and Transfer Learning*. 27, 17-36

Bengio, Y. Courville, A. Goodfellow, I. (2018). *L'apprentissage profond*. Massot editions.

Bezdan, T., & Bačanin Džakula, N. (2019). Convolutional Neural Network Layers and Architectures. *Proceedings of the International Scientific Conference - Sinteza 2019*.

Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. In *Proceedings of COMPSTAT'2010*.

Boonen, B. Geenen, M. Nève, P. (2016). *Ma peau, ma santé: Guide sur le cancer de la peau et sa prévention*. Lannoo

Bottou, L. (2012). Stochastic Gradient Descent Tricks. *Neural Networks: Tricks of the Trade*.

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, 24, 123-140.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45, 5-32.

Campos-do-Carmo, G., & Ramos-e-Silva, M. (2008). Dermoscopy: basic concepts. *International Journal of Dermatology*, 47.

Capdehourat, G., Corez, A., Bazzano, A., Alonso, R., & Musé, P. (2011). Toward a combined tool to assist dermatologists in melanoma detection from dermoscopic images of pigmented skin lesions. *Pattern Recognit. Lett.*, 32, 2187-2196.

Chakravorty, R., Liang, S., Abedini, M., & Garnavi, R. (2016). Dermatologist-like feature extraction from skin lesion for improved asymmetry classification in PH2 database. *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 3855-3858.

Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800-1807.

Clark, W.H., From, L., Bernardino, E.A., & Mihm, M.C. (1969). The histogenesis and biologic behavior of primary human malignant melanomas of the skin. *Cancer research*, 29 3, 705-27.

Codella, N.C., Cai, J., Abedini, M., Garnavi, R., Halpern, A., & Smith, J.R. (2015). Deep Learning, Sparse Coding, and SVM for Melanoma Recognition in Dermoscopy Images. *MLMI*.

Cox, N.H., & Coulson, I.H. (2008). Diagnosis of Skin Disease. *Rook's Textbook of Dermatology*, Oxford: Blackwell Science 5.

Cueva, W.F., Muñoz, F.C., Vásquez, G., & Delgado., G. (2017). Detection of skin cancer "Melanoma" through computer vision. *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 1-4.

Dauphin, Y., Pascanu, R., Gülçehre, Ç., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *NIPS*.

Dervaux, J. L. (2011). *Les maladies de la peau*. Dangles

Dhivyaa, C.R., Sangeetha, K., Balamurugan, M., Amaran, S., Vetriselvi, T., &

Dietterich, T.G. (2000). Ensemble Methods in Machine Learning. *Multiple Classifier Systems, 1857*, 1-15.

Duchi, J.C., Hazan, E., & Singer, Y. (2010). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *Annals of Statistics*, 7, 1-26.

Efron, B. Tibshirani, R.J. (1993). *An introduction to the Bootstrap*. Chapman & Hall.

El-khatib, H., Popescu, D., & Ichim, L. (2020). Deep Learning-Based Methods for Automatic Diagnosis of Skin Lesions †. *Sensors (Basel, Switzerland)*, 20.

Fausett, L. V. (1994). *Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*. Prentice-Hall.

Fernández, D.R., Berenguer, V.J., Payá, A.S., & Sánchez, B. (2011). A decision support system for the diagnosis of melanoma: A comparative approach. *Expert Syst. Appl.*, 38, 15217-15223.

Filali, Y., El Khoukhi, H., Sabri, M.A., Yahyaouy, A., & Aarab, A. (2019). Texture Classification of skin lesion using convolutional neural network. *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 1-5.

Filali, Y., El Khoukhi, H., Sabri, M.A., & Aarab, A. (2020). Efficient fusion of handcrafted and pre-trained CNNs features to classify melanoma skin cancer. *Multimedia Tools and Applications*, 1 - 20

Fisher, R.A. (1936). The Use Of Multiple Measurements In Taxonomic Problems. *Annals of Human Genetics*, 7, 179-188.

Fitzpatrick, T.B. (1988). The validity and practicality of sun-reactive skin types I through VI. *Archives of dermatology*, 12(6), 869-71.

Fleury, G. Gondran, M. Lacomme, P. Samir, C. (2021). *Découverte du Machine Learning: Les outils de l'apprentissage automatique*. Ellipses.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, 193-202.

Fumera, G., Roli, F. (2005). A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 942-956.

Gallo, R.L. (2017). Human Skin is the Largest Epithelial Surface for Interaction with Microbes. *The Journal of investigative dermatology*, 137(6), 1213-1214.

Ghalejoogh, G.S., Kordy, H.M., & Ebrahimi, F. (2020). A hierarchical structure based on Stacking approach for skin lesion classification. *Expert Syst. Appl.*, 145, 113-127.

Glasmachers, T. (2017). Limits of End-to-End Learning. *ACML*.

Glazer, A.M., Rigel, D.S., Winkelmann, R.R., & Farberg, A.S. (2017). Clinical Diagnosis of Skin Cancer: Enhancing Inspection and Early Recognition. *Dermatologic clinics*, 35(4), 409-416.

Goodfellow, I., Bengio, Y., Courville, A. (2016), *Deep Learning*, MIT Press,

Griffiths, C., Barker, J., Bleiker, T., Chalmers, R. & Creamer, D. (2016). *Rook's textbook of dermatology*, John Wiley & Sons.

Grivokostopoulou, F. (Eds.), *Advances in Integrations of Intelligent Methods*. Series Title: Smart Innovation, Systems and Technologies, 170, 85-98.

Goyal, M., Knackstedt, T., Yan, S., & Hassanpour, S. (2020). Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities. *Computers in biology and medicine*, 127, 104065.

Guo, Y., Liu, Y., Oerlemans, A.A., Lao, S., Wu, S., & Lew, M.S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187, 27-48.

Goussous, A.E. (2019). Skin Lesion Classification Using Deep Neural Network. *ArXiv, abs/1911.07817*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

Hinton, G., Srivastava, N. & Swersky, K. (2012). Neural networks for machine learning overview of mini-batch gradient descent.

Hosny, K.M., Kassem, M.A., & Foaud, M.M. (2018). Skin Cancer Classification using Deep Learning and Transfer Learning. *2018 9th Cairo International Biomedical Engineering Conference (CIBEC)*, 90-93.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv, abs/1704.04861*.

Huang, G., Liu, Z., & Weinberger, K.Q. (2017). Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261-2269.

Hubel, D.H., & Wiesel, T.N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195, 215-243.

Husham, A., Hazim Alkawaz, M., Saba, T., Rehman, A., & Saleh Alghamdi, J. (2016). Automated nuclei segmentation of malignant using level sets. *Microscopy Research and Technique*, 79.

Iandola, F.N., Moskewicz, M.W., Ashraf, K., Han, S., Dally, W.J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *ArXiv, abs/1602.07360*.

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ArXiv, abs/1502.03167*.

Jadhav, A.R., Ghontale, A.G., & Shrivastava, V.K. (2018). Segmentation and Border Detection of Melanoma Lesions Using Convolutional Neural Network and SVM. *Computational Intelligence: Theories, Applications and Future Directions - Volume I*.

Javed, R., Rahim, M.S., Saba, T., & Rehman, A. (2019). A comparative study of features selection for skin lesion detection from dermoscopic images. *Network Modeling Analysis in Health Informatics and Bioinformatics, 9*.

Johnpaul, P. (2020). Skin lesion classification using decision trees and random forest algorithms. *Journal of Ambient Intelligence and Humanized Computing, 1-13*.

Kassem, M.A., Hosny, K.M., & Fouad, M.M. (2020). Skin Lesions Classification Into Eight Classes for ISIC 2019 Using Deep Convolutional Neural Network and Transfer Learning. *IEEE Access, 8, 114822-114832*.

Khan, A., Sohail, A., Zahoor, U., & Qureshi, A.S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review, 1 - 62*.

Kingma, D.P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *CoRR, abs/1412.6980*.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60, 84 - 90*.

Krogh, A., & Hertz, J.A. (1991). A Simple Weight Decay Can Improve Generalization. In *Advances in Neural Information Processing Systems*, 950-957

Kaware, S.R. Wadne, V. S. (2020). *Approche comparative des algorithmes d'apprentissage automatique*. Editions Notre Savoir.

Kiani, K., & Sharafat, A.R. (2011). E-shaver: An improved DullRazor® for digitally removing dark and light-colored hairs in dermoscopic images. *Computers in biology and medicine*, 41 3, 139-45.

Kumar, S.D., & Chandra, J. (2019). Skin Cancer Classification using Machine Learning for Dermoscopy Image 1457.

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., & Jackel, L.D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1, 541-551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE* 86(11): 2278-2324.

Leiter, U., Keim, U., & Garbe, C. (2020). Epidemiology of Skin Cancer: Update 2019. *Advances in experimental medicine and biology*, 1268, 123-139.

Li, X., Wu, J., Jiang, H., Chen, E.Z., Dong, X., & Rong, R. (2018). Skin Lesion Classification Via Combining Deep Learning Features and Clinical Criteria Representations. *bioRxiv*.

Litjens, G.J., Kooi, T., Bejnordi, B.E., Setio, A.A., Ciompi, F., Ghafoorian, M., Laak, J.V., Ginneken, B.V., & Sánchez, C.I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60-88.

Losquadro, W.D. (2017). Anatomy of the Skin and the Pathogenesis of Nonmelanoma Skin Cancer. *Facial plastic surgery clinics of North America*, 25(3), 283-289.

Mahbod, A., Ecker, R., & Ellinger, I. (2019). Skin Lesion Classification Using Hybrid Deep Neural Networks. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1229-1233.

McGuire, S.T., Secrest, A.M., Andrulonis, R., & Ferris, L.K. (2011). Surveillance of patients for early detection of melanoma: patterns in dermatologist vs patient discovery. *Archives of dermatology*, 147(6), 673-678.

Melbin, K., & Raj, Y.J. (2021). Integration of modified ABCD features and support vector machine for skin lesion types classification. *Multim. Tools Appl.*, 80, 8909-8929.

Mélinopoulos, A. & Levacher, C. (2012). *La peau : Structure et Physiologie (2e édition)*. Lavoisier.

Menzies, S.W., Ingvar, C., Crotty, K.A., & McCarthy, W. (1996). Frequency and morphologic characteristics of invasive melanomas lacking specific surface microscopic features. *Archives of dermatology*, 132 10, 1178-82.

Menzies, S.W., Neuber, N.M., Paoli, J., Rabinovitz, H.S., Rinner, C., Scope, A.,

Mermillod, M. (2016). *Réseaux de neurones biologiques et artificiels. Vers l'émergence de systèmes artificiels conscients?*. de boeck press.

Moldovanu, S., Damian Michis, F.A., Biswas, K.C., Culea-Florescu, A., & Moraru, L. (2021). Skin Lesion Classification Based on Surface Fractal Dimensions and

Statistical Color Cluster Features Using an Ensemble of Machine Learning Techniques. *Cancers*, 13.

Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269, 543-547.

Mporas, I., Perikos, I., & Paraskevas, M. (2020). Color Models for Skin Lesion Classification from Dermatoscopic Images. In: Hatzilygeroudis, I., Perikos, I.,

Mughal, B., Muhammad, N., Sharif, M., Saba, T., & Rehman, A. (2017). Extraction of breast border and removal of pectoral muscle in wavelet domain. *Biomedical Research-tokyo*, 28, 5041-5043.

Muhlbaier, M., Topalis, A., Polikar, R. (2005). Ensemble Confidence Estimates Posterior Probability. *Multiple Classifier Systems*, 3541, 326-335.

Murty, M.N., Devi, V.S. (2011). Bayes Classifier. In: *Pattern Recognition. Undergraduate Topics in Computer Science*, Springer, London.

Murugan, A., Nair, S.A., & Kumar, K.P. (2019). Detection of Skin Cancer Using SVM, Random Forest and kNN Classifiers. *Journal of Medical Systems*, 43, 1-9.

Mutlag, W.K., Ali, S.K., Aydam, Z.M., & Taher, B.H. (2020). Feature Extraction Methods: A Review.

Nachbar, F., Stolz, W., Merkle, T., Cognetta, A.B., Vogt, T., Landthaler, M., Bilek, P., Braun-falco, O., & Plewig, G. (1994). The ABCD rule of dermatoscopy. High prospective value in the diagnosis of doubtful melanocytic skin lesions. *Journal of the American Academy of Dermatology*, 30 4, 551-9.

Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269, 543-547.

Oliveira, R.B., Pereira, A.S., & Tavares, J.M. (2018). Computational diagnosis of skin lesions from dermoscopic images using combined features. *Neural Computing and Applications*, 1-21.

Ozkan, I.A., & Koklu, M. (2017). Skin Lesion Classification using Machine Learning Algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, 5, 285-289.

Pagano, J. O. A. (2022). *Guérir du psoriasis*. Macro

Pacheco, A.G., Ali, A., & Trappenberg, T.P. (2019). Skin cancer detection based on deep learning and entropy to detect outlier samples. *ArXiv, abs/1909.04525*.

Parrish, J.A., Fitzpatrick, T.B., Tanenbaum, L., & Pathak, M.A. (1974). Photochemotherapy of psoriasis with oral methoxsalen and longwave ultraviolet light. *The New England journal of medicine*, 291(23), 1207-1211.

Pathan, S., Prabhu, K.G., & Siddalingaswamy, P.C. (2018). Techniques and algorithms for computer aided diagnosis of pigmented skin lesions - A review. *Biomed. Signal Process. Control*. 39, 237-262.

Patil, M.V., & Dongre, N.M. (2020). Melanoma Detection Using HSV with SVM Classifier and De-duplication Technique to Increase Efficiency. *Computing Science, Communication and Security*, 1235, 109 - 120.

Perez, L., & Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *ArXiv, abs/1712.04621*.

Perez, F., Vasconcelos, C.N., Avila, S., & Valle, E. (2018). Data Augmentation for Skin Lesion Analysis.

Perez, F., Avila, S., & Valle, E. (2019). Solo or Ensemble? Choosing a CNN Architecture for Melanoma Classification. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2775-2783.

Pickover, C. A. (2021). *La fabuleuse histoire de l'intelligence artificielle*. Dunod

Pomarede, N. (2016). *Psoriasis et eczema, les solutions*. Alpen

Rao, C.R. (1948). The Utilization of Multiple Measurements in Problems of Biological Classification. *Journal of the royal statistical society series b-methodological*, 10, 159-193.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517-6525.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *ArXiv*, *abs/1804.02767*.

Ruela, M., Barata, C., Marques, J.S., & Rozeira, J. (2017). A system for the detection of melanomas in dermoscopy images using shape and symmetry features. *Comput. methods Biomech. Biomed. Eng. Imaging Vis.*, 5, 127-137.

Salido, J.A., & Ruiz, C.R. Using Deep Learning to Detect Melanoma in Dermoscopy Images.

Schapire, R.E. (1990). The strength of weak learnability. *Machine Learning*, 5, 197-227.

Schaul, T., Antonoglou, I., & Silver, D. (2014). Unit Tests for Stochastic Optimization. *CoRR*, *abs/1312.6055*.

Shrestha, A.K., & Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures. *IEEE Access*, *7*, 53040-53065.

Sellam, S. (2008). *Les maladies de la peau*. Berangel

Shorten, C., & Khoshgoftaar, T.M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*, 1-48.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, *abs/1409.1556*.

Singh, L., Janghel, R.R., & Sahu, S.P. (2020). Designing a Retrieval-Based Diagnostic Aid using Effective Features to Classify Skin Lesion in Dermoscopic Images. *Procedia Computer Science*, *167*, 2172-2180.

Someya, T. & Amagai, M. (2019). Toward a new generation of smart skins. *Nature Biotechnology*, *37*, 382-388.

Soyer, H.P., Sinz, C., Thomas, L., Zalaudek, I., & Kittler, H. (2019). Expert-Level Diagnosis of Nonpigmented Skin Cancer by Combined Convolutional Neural Networks. *JAMA Dermatology*, *155*, 58-65.

Stolz, W. Riemann, A. Cagnetta, A. Pillet, L. Abmayr & W. Hoelzel (1994). ABCD rule of dermatoscopy: a new practical method for early recognition of malignant melanoma. *European Journal of Dermatology*, *4*(7), 521-527.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014) Dropout : a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929-1958.

Sung, H., Ferlay, J., Siegel, R.L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*, 71, 209 - 249.

Sutskever, I., Martens, J., Dahl, G.E., & Hinton, G.E. (2013). On the importance of initialization and momentum in deep learning. *ICML*.

Surówka, G., & Ogorzałek, M. (2019). Naive Bayes Learning of Dermoscopy Images. *ICAISC*. 11509, 294-304.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818-2826.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A.A. (2017). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *AAAI*.

Tan, M., & Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv, abs/1905.11946*.

Taylor, L., & Nitschke, G.S. (2018). Improving Deep Learning with Generic Data Augmentation. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1542-1547.

Titus, L.J., Clough-Gorr, K., Mackenzie, T., Perry, A.E., Spencer, S.K., Weiss, J.,

Travel, U. C. (2015). *Melanoma: signs and prevention*

Tschandl, P., Rosendahl, C., Akay, B.N., Argenziano, G., Blum, A., Braun, R.P., Cabo, H., Gourhant, J.Y., Kreusch, J., Lallas, A., Lapins, J., Marghoob, A.A.,

Valle, E., Fornaciali, M., Menegola, A., Tavares, J., Bittencourt, F.V., Li, L.T., &

Vannieuwenhuyze, A. (2019), *Intelligence artificielle vulgarisée, la machine learning et le deep learning par la pratique*, ENI.

Van Smeden, J., Janssens, M., Gooris, G., & Bouwstra, J.A. (2014). The important role of stratum corneum lipids for the cutaneous barrier function. *Biochimica et biophysica acta*, 1841 3, 295-313.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

Whited, J.D., & Grichnik, J.M. (1998). The rational clinical examination. Does this patient have a mole or a melanoma? *JAMA*, 279 9, 696-701 .

Wu, Y., Ianakiev, K., Govindaraju, V. (2002). Improved k-nearest neighbor classification. *Pattern recognition*, 35(10), 2311-2318.

Yang, Q. Zhang, Y. Dai, W. Pan, S.J. (2020). *Transfer Learning*. Cambridge University Press.

Yu, L., Chen, H., Dou, Q., Qin, J., & Heng, P. (2017). Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks. *IEEE Transactions on Medical Imaging*, 36, 994-1004.

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6848-6856.

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. (2018). Learning Transferable Architectures for Scalable Image Recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8697-8710.

Sommaire

La peau subit des agressions externes, physiques, chimiques et bactériologiques. Elle y répond par des réactions vasculaires et tissulaires non spécifiques, suivant un mécanisme parfois allergique. Les lésions cutanées se rapportent à un terme général qui caractérise une partie de la peau ayant, en comparaison de la peau l'entourant, une croissance/apparence ou structure qui diffère de tissus considérés comme sains. Pour diagnostiquer une lésion cutanée, une variété d'indices visuels peuvent être utilisés tels que la morphologie lésionnelle individuelle, la distribution du site corporel, la couleur, la desquamation et la disposition des lésions. Lorsque les éléments individuels sont analysés séparément, le processus de reconnaissance peut être assez complexe. Pour certaines maladies, leurs lésions sont si similaires qu'elles ne peuvent être distinguées visuellement. Les limites du dépistage visuel des lésions cutanées peuvent être surmontées grâce à l'utilisation du diagnostic assisté par ordinateur (DAO). Les systèmes DAO tentent d'imiter les performances des dermatologues pour déterminer la lésion cutanée en tant que normale ou anormale. Ces systèmes utilisent des algorithmes d'extraction de caractéristiques conçus par l'homme en combinaison avec un classifieur. Cependant, lorsque nous étendons les maladies de la peau à une gamme plus large, où les caractéristiques sont si complexes que la conception de caractéristiques artisanales devient irréalisable, l'approche traditionnelle échoue. C'est pour cette raison, qu'on fait recours aux techniques d'apprentissage profond. La percée de l'apprentissage profond a conduit à des progrès dans de nombreuses tâches de vision par ordinateur. Le problème fondamental abordé par cette thèse peut être énoncé comme une question : comment utiliser les derniers développements en apprentissage profond pour mettre en œuvre différents classifieurs capable d'examiner une image contenant une lésion cutanée et de prédire un résultat (malin ou bénin).

ملخص

يتعرض الجلد لهجمات خارجية وفيزيائية وكيميائية وبكتريولوجية. يستجيب مع تفاعلات الأوعية الدموية والأنسجة غير المحددة، وفي بعض الأحيان يتبع آلية الحساسية. تشير الآفات الجلدية إلى مصطلح عام يميز جزءاً من الجلد له، مقارنة بالجلد المحيط، نمو/مظهر أو بنية تختلف عن الأنسجة التي تعتبر صحية. لتشخيص الآفة الجلدية، يمكن استخدام مجموعة متنوعة من القرائن البصرية مثل مورفولوجيا الآفة الفردية، وتوزيع موقع الجسم، واللون، والتقشر، وترتيب الآفات. عندما يتم تحليل العناصر الفردية بشكل منفصل، يمكن أن تكون عملية التعرف معقدة للغاية. بالنسبة لبعض الأمراض، تكون آفاتهم متشابهة لدرجة أنه لا يمكن تمييزها بصرياً. يمكن التغلب على قيود فحص آفات الجلد المرئية من خلال استخدام التشخيص بمساعدة الكمبيوتر (CAD). تحاول أنظمة CAO محاكاة أداء أطباء الجلد في تحديد الآفة الجلدية على أنها طبيعية أو غير طبيعية. تستخدم هذه الأنظمة خوارزميات استخراج الميزات من تصميم الإنسان جنباً إلى جنب مع المصنف. ومع ذلك، لكن عندما نوسع الأمراض الجلدية إلى نطاق أوسع، حيث تكون الميزات معقدة للغاية بحيث يصبح تصميم الميزات المصنوعة يدوياً غير ممكن، فإن النهج التقليدي يفشل. هذا هو سبب استخدام تقنيات التعلم العميق. يمكن طرح المشكلة الأساسية التي تتناولها هذه الأطروحة على شكل سؤال: كيفية استخدام أحدث التطورات في التعلم العميق لتنفيذ مصنفات مختلفة قادرة على فحص صورة تحتوي على آفة جلدية والتنبؤ بنتيجة (خبيثة أو حميدة)

Summary

The skin is subject to external, physical, chemical and bacteriological attacks. It responds with non-specific vascular and tissue reactions, sometimes following an allergic mechanism. Cutaneous lesions refer to a general term that characterizes a part of the skin having, in comparison to the surrounding skin, a growth/appearance or structure that differs from tissue considered healthy. To diagnose a skin lesion, a variety of visual clues can be used such as individual lesion morphology, body site distribution, color, desquamation, and arrangement of lesions. When the individual elements are analyzed separately, the recognition process can be quite complex. For some diseases, their lesions are so similar that they cannot be visually distinguished. The limitations of visual skin lesion screening can be overcome through the use of computer-aided diagnosis (CAD). DAO systems attempt to mimic the performance of dermatologists in determining the skin lesion as normal or abnormal. These systems use human-designed feature extraction algorithms in combination with a classifier. However, when we extend skin diseases to a wider range, where the features are so complex that designing handmade features becomes unfeasible, the traditional approach fails. This is why deep learning techniques are used. The breakthrough in deep learning has led to advancements in many computer vision tasks. The fundamental problem addressed by this thesis can be stated as a question: how to use the latest developments in deep learning to implement different classifiers capable of examining an image containing a skin lesion and predicting an outcome (malignant or benign).