



l'École Nationale Supérieure d'Arts et Métiers de Rabat
Ingénierie Numérique en Data Science et Intelligence Artificielle
(INDIA)

SUJET :

Année universitaire : 2021/2022

SOUAD Ayoub

Pr. JOUDAR Nour-eddine -Encadrant
 Pr. BEKKALI Mohammed -Examineur

Remerciements

Au terme de ce travail, nous voudrions exprimer nos remerciements et notre profonde reconnaissance à tout ceux qui ont contribués de prêt ou de loin à sa réalisation :

Nous voudrions, en premier lieu, exprimer notre profonde reconnaissance à notre encadrant, Professeur **JOUDAR Nour-eddine**, qui, grâce à ses orientations précieuses, ses encouragements valorisants et son orientation compétente, nous avons pu réalisé ce travail et s'initier à la recherche, pour sa direction compétente de ce mémoire, son soutien dans les moments d'incertitude et sa disponibilité inconditionnelle.

Nous tenons à remercier toute l'équipe pédagogique et administrative de l'École Nationale des Arts et Métiers de Rabat, qui nous a aidés par ses qualités humaines, sa rigueur professionnelle.

Nous remercions les enseignants-chercheurs pour leurs efforts de transmission aux étudiants de leurs connaissances professionnelles.

Nous exprimons notre gratitude à tous nos collègues de la classe INDIA pour leur soutien amical durant ces années d'étude.

Resumé

Le diagnostic assisté par ordinateur pourrait remplacer ou au moins aider de manière optimale les ophtalmologistes à diagnostiquer les anomalies oculaires les plus courantes, comme la néovascularisation choroïdienne (NVC), l'œdème maculaire diabétique (OMD) et le DRUSEN ; ces maladies présentent un risque considérable dans la perte de notre vision. La tomographie par cohérence optique OCT est une technique d'imagerie utilisée pour avoir un aperçu détaillé des différentes couches rétiniennes, grâce à la combinaison des scans OCT, du traitement d'image et des techniques de segmentation. Cela va permettre l'ophtalmologue à faire son diagnostic au fond de l'œil. Notre objectif dans ce travail est d'automatiser le diagnostic le maximum possible, afin de réduire les erreurs dues à la fatigue et le temps nécessaire pour le diagnostic. Notre méthode est basée sur l'apprentissage profond en adoptant une base de données public des images OCT avec des étiquettes indiquant les types des maladies, et en appliquant des modèles de réseaux de neurones convolutifs pertinents avec des techniques d'apprentissage par transfert. Il reste une partie plus importante pour la confidentialité du patient au sein de l'hôpital, c'est le cryptage de ces images d'une manière sécurisée et les stockées dans une base de données.

Abstract

Computer-aided diagnosis could replace or at least optimally assist ophthalmologists in diagnosing the most common ocular anomalies, such as choroidal neovascularisation (CNV), diabetic macular oedema (DME) and DRUSEN; these diseases pose a considerable risk to our vision loss. Optical Coherence Tomography (OCT) is an imaging technique used to gain a detailed insight into the different layers of the retina, using a combination of OCT scans, image processing and segmentation techniques. This will allow the ophthalmologist to make a diagnosis at the back of the eye. Our goal in this work is to automate the diagnosis as much as possible, in order to reduce errors due to fatigue and the time needed for diagnosis. Our method is based on deep learning by adopting a public database of OCT images with labels indicating disease types, and applying relevant convolutional neural network models with transfer learning. A more important part of patient confidentiality within the hospital is encrypting these images in a meaningful way and storing them in a database.

Table des matières

Remerciements	2
Resumé	3
Abstract	4
INTRODUCTION GÉNÉRALE	10
1 Réseaux de neurones convolutifs	11
1.1 Introduction	11
1.2 CNN layers	11
1.2.1 Convolutional Layer	11
1.2.2 Padding	12
1.2.3 Filtres	14
1.2.4 Pooling layer	14
1.2.5 Fully connected Layer	15
1.3 Exemple d'architecture convolutive : LeNet-5	17
1.4 Aller plus profond avec CNN et la réduction des coûts de computation . .	19
1.4.1 apprentissage résiduel	19
1.4.2 1x1 Convolution	20
1.5 Transfer learning	23
1.5.1 Définition	23
1.5.2 Apprentissage par transfert avec des modèles prétraités	24
1.6 Conclusion	25
2 Cryptage des images numériques par les systèmes chaotiques	26
2.1 Introduction	26
2.2 Logistic map	26
2.3 Algorithme de cryptage d'image basé sur logistic map	27
2.4 Simulation et discussion	29
2.4.1 Histogramme	29
2.4.2 Sensibilité de la clé	30
2.5 Conclusion	31
3 OCT diagnostic	32
3.1 Introduction	32
3.2 Présentation du projet	32
3.2.1 Contexte du projet	32

3.2.2	Problématique	32
3.2.3	Objectif	33
3.2.4	Pathologies maculaires : classes traitées	33
3.3	Matériel et méthodes	34
3.3.1	Base de données	34
3.3.2	Méthode proposée	34
3.3.3	Matériel pour l'implémentation du modèle	34
3.3.4	Méthode de classification	35
3.4	Résultats et Discussion	36
3.4.1	Critères d'évaluation	36
3.4.2	Résultats de la méthode proposée	37
3.5	Implémentation de la solution	40
3.5.1	Présentation de l'application	40
3.5.2	Technologies Utilisées	40
3.5.3	Guide d'utilisation de l'application web	41
3.6	Conclusion	44
CONCLUSION GÉNÉRALE ET PERSPECTIVES		45

Liste des figures

1.1	Couches CNN avec champs réceptifs locaux rectangulaires	12
1.2	Connexions entre les couches et zero padding	13
1.3	Réduction de la dimensionnalité en utilisant un stride de 2	13
1.4	Appliquer deux filtres différents pour obtenir deux cartes de caractéristiques	14
1.5	Max Pooling	15
1.6	Average Pooling	15
1.7	Fully connected layer - la première couche est connecté à chaque nœud de la deuxième couche	16
1.8	Architecture de LeNet un réseau de neurones convolutif ici pour la reconnaissance de chiffres	17
1.9	Sommaire de l'architecture LeNet5	18
1.10	Erreur de test et de training pour les 20 layers et 56 layers	19
1.11	Apprentissage résiduel : bloc résiduel	19
1.12	Convolution 1x1	20
1.13	Appliquer un filtre 5x5	21
1.14	Réduction de la dimensionnalité en appliquant une convolution 1x1	21
1.15	Architecture Inception	22
1.16	La couche "Bottle-neck"	23
1.17	Approche traditionnelle vs. Approche de Transfert Learning	24
1.18	Transfert Learning appliquée au modèle VGG16	25
2.1	Évolution de différentes conditions initiales en fonction de r	27
2.2	Shéma de Cryptage et décryptage	27
2.3	Résultats des images de test	29
2.4	les histogrammes : image a ; image b ; image cryptée c de a ; image cryptée d de b	30
2.5	RMSE de l'image originale et l'image cryptée en fonction de r	31
3.1	Schéma d'entraînement du modèle	36
3.2	Courbes de l'exactitude et de l'erreur pendant l'entraînement et la validation : modèle VGG16	37
3.3	Courbes de l'exactitude et de l'erreur pendant l'entraînement et la validation : modèle InceptionV3	37
3.4	Paramètres de l'évaluation : VGG16	38
3.5	Paramètres de l'évaluation : InceptionV3	38
3.6	Matrice de Confusion VGG16	39
3.7	Matrice de Confusion InceptionV3	39

3.8 Schéma d'implémentation d'application	40
---	----

Liste des tableaux

3.1	La base de données OCT	34
3.2	Séparation de la base de données	35

INTRODUCTION GÉNÉRALE

Les pathologies rétiniennes[13] sont la cause la plus fréquente de infantile dans le monde. La détection rapide et automatique des maladies est essentielle et urgente pour réduire la charge de travail de l'ophtalmologiste. Les ophtalmologistes diagnostiquent les maladies en se basant sur la reconnaissance des formes par la visualisation directe ou indirecte de l'œil et de ses structures environnantes. La dépendance au fond de l'œil et à son analyse rend le domaine de l'ophtalmologie parfaitement adapté pour bénéficier des algorithmes d'apprentissage profond [1].

Plusieurs techniques d'apprentissage automatique ont été appliquées dans ce contexte pour la classification et la segmentation automatiques des images OCT [7]. Certaines études utilisent la technique d'apprentissage profond qui montre une haute performance et un faible coût ; en contrepartie, elle nécessite l'utilisation d'une immense base de données avec la même caractéristique.

Dans ce contexte, l'ENSAM Rabat contribue efficacement dans le développement du domaine de l'intelligence artificielle dans la santé, en créant une filière d'ingénierie numérique en data science et intelligence artificielle, qui va permettre aux ingénieurs d'état biomédicaux en data science et IA de faire bénéficier de leur formation pointue dans la gestion des problématiques AI.

Dans cette période de semestre 4 on étaient travaillé sur la prédiction des maladies de la rétine on utilisant le "deep learning" [10]et le cryptage des données du patient par les systèmes chaotiques[11][12].

Ce rapport est constitué de trois chapitres : Le premier chapitre présente une étude explicative sur les réseaux de neurones convolutifs réseaux neurons convolutifs[2] [3] [4] [5] [6]. Le deuxième parle et discute le cryptage de l'image par les systèmes chaotiques. Enfin, une application web de la classification des maladies de la rétine en se basant sur les réseaux de neurones convolutifs et qui respecte la confidentialité du patient.

Réseaux de neurones convolutifs

1.1 Introduction

Les réseaux de neurones convolutifs (CNN) sont nés de l'étude du cortex visuel du cerveau et sont utilisés dans la reconnaissance d'images depuis les années 1980. Ces dernières années, grâce à l'augmentation de la puissance de calcul, à la quantité de données d'entraînement disponibles et aux astuces d'entraînement des réseaux profonds, les CNN ont réussi à atteindre des performances surhumaines sur certaines tâches visuelles complexes. Ils alimentent des services de recherche d'images, des voitures à conduite autonome, des systèmes de classification automatique de vidéos, etc. De plus, les CNN ne se limitent pas à la perception visuelle : ils réussissent également de nombreuses autres tâches, comme la reconnaissance vocale et le traitement du langage naturel. Cependant, nous nous concentrons sur les applications visuelles pour l'instant.

1.2 CNN layers

1.2.1 Convolutional Layer

La couche convolutive est la composante la plus importante d'un CNN : les neurones de la première couche convolutive ne sont pas connectés à chaque pixel de l'image d'entrée, mais uniquement aux pixels de leur champ récepteur (voir figure 1). À son tour, chaque neurone de la deuxième couche convolutive est connecté uniquement aux neurones situés dans un petit rectangle de la première couche. Cette architecture permet au réseau de se concentrer sur de petites caractéristiques de bas niveau dans la première couche cachée, puis de les assembler en caractéristiques de plus haut niveau dans la couche cachée suivante, et ainsi de suite. Cette structure hiérarchique est courante dans les images du monde réel, et c'est l'une des raisons pour lesquelles les réseaux CNN fonctionnent si bien pour la reconnaissance d'images.

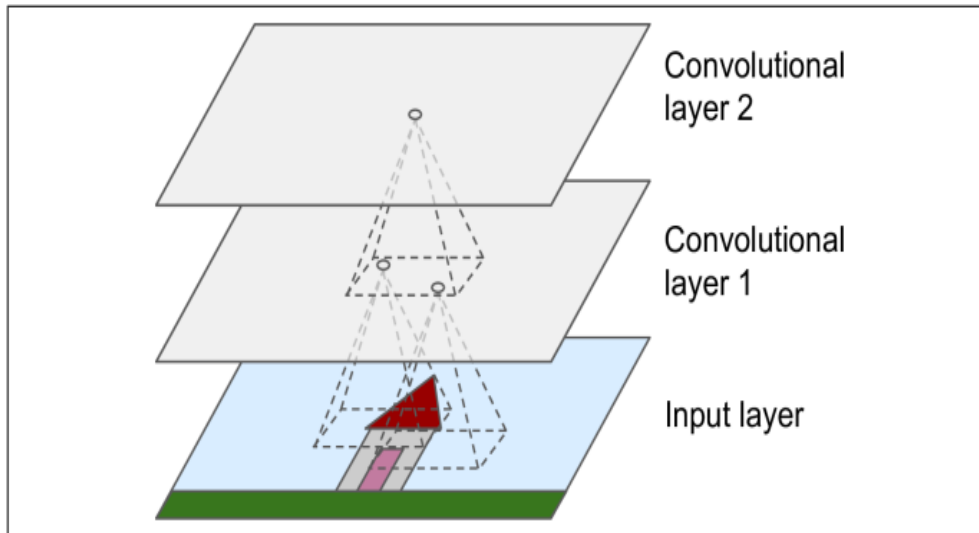


FIGURE 1.1 – Couches CNN avec champs réceptifs locaux rectangulaires

Dans la couche convolutionnelle, une matrice appelée noyau est passée sur la matrice d'entrée pour créer une carte de caractéristiques pour la couche suivante. Nous exécutons une opération mathématique appelée convolution en faisant glisser la matrice Kernel sur la matrice d'entrée. À chaque emplacement, une multiplication matricielle par élément est effectuée et additionne le résultat sur la carte de caractéristiques. La convolution est un type spécialisé d'opération linéaire qui est largement utilisé dans divers domaines, notamment le traitement des images, les statistiques et la physique. La convolution peut être appliquée sur plus d'un axe. Si nous avons une image bidimensionnelle en entrée, I , et un filtre à noyau bidimensionnel, K , la convolution peut être appliquée sur plus d'un axe. 2 dimensions, K , l'image convoluée est calculée de la manière suivante comme suit :

$$S(i, j) = \sum_m \sum_n I(m, n) k(i - m, j - n) \quad (1.1)$$

1.2.2 Padding

Un neurone situé dans la ligne i , colonne j d'une couche donnée est connecté aux sorties des neurones de la couche précédente situés dans les lignes i à $i + f_h - 1$, colonnes j à $j + f_w - 1$, où f_h et f_w sont la hauteur et la largeur du champ réceptif (voir Figure 2). Pour qu'une couche ait la même hauteur et la même largeur que la couche précédente, il est nécessaire d'ajouter des zéros autour des entrées, comme le montre le schéma. C'est ce qu'on appelle le "zero padding".

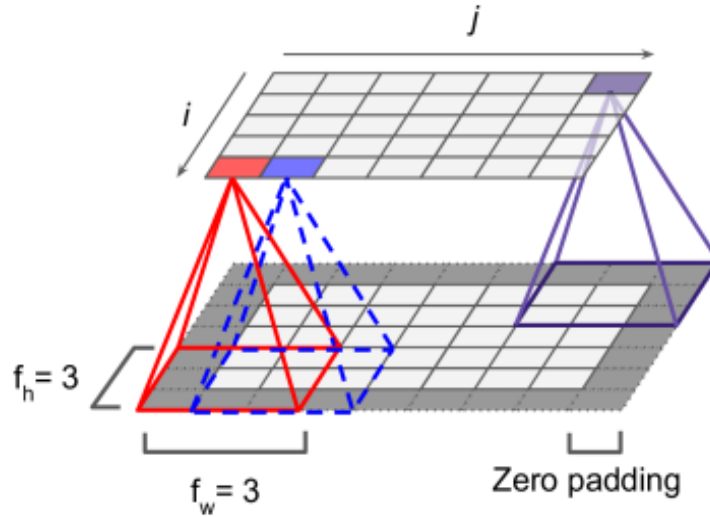


FIGURE 1.2 – Connexions entre les couches et zero padding

Il est également possible de connecter une grande couche d'entrée à une couche beaucoup plus petite en espaçant les champs réceptifs, comme le montre la figure 1.3. Cela réduit considérablement la complexité de calcul du modèle. Le passage d'un champ réceptif à l'autre est appelé le stride. Dans le schéma, une couche d'entrée de 5×7 (plus un zero padding) est connectée à une couche de 3×4 , avec des champs réceptifs de 3×3 et un stride de 2 (dans cet exemple, le stride est le même dans les deux sens, mais ce n'est pas obligatoire). Un neurone situé dans la ligne i , colonne j de la couche supérieure est connecté aux sorties des neurones de la couche précédente situées dans les lignes $i \times s_h$ à $i \times s_h + f_h - 1$, colonnes $j \times s_w$ à $j \times s_w + f_w - 1$, où s_h et s_w sont les pas verticaux et horizontaux.

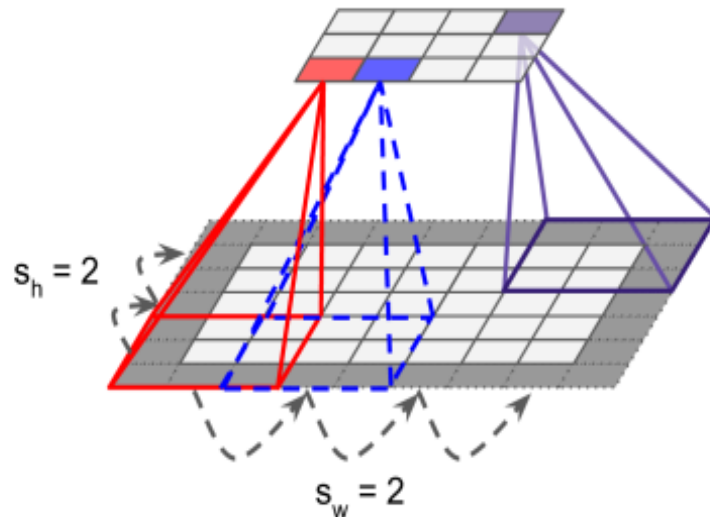


FIGURE 1.3 – Réduction de la dimensionnalité en utilisant un stride de 2

Utilisons une image de dimension $(n \times n)$ et un filtre de dimension $(f \times f)$ avec un padding p et un stride s . La taille de sortie aura la dimension :

$$\left[\frac{n + 2p - f}{s} + 1 \right] \times \left[\frac{n + 2p - f}{s} + 1 \right]$$

1.2.3 Filtres

Les poids d'un neurone peuvent être représentés par une petite image de la taille du champ réceptif. Par exemple, la figure 1.4 montre deux ensembles possibles de poids, appelés filtres (ou kernel de convolution). Le premier est représenté par un carré noir avec une ligne verticale blanche au milieu (il s'agit d'une matrice 7×7 remplie de 0, à l'exception de la colonne centrale, qui est remplie de 1) ; les neurones utilisant ces poids ignoreront tout ce qui se trouve dans leur champ récepteur, à l'exception de la ligne verticale centrale (puisque toutes les entrées seront multipliées par 0, à l'exception de celles situées dans la ligne verticale centrale). Le deuxième filtre est un carré noir avec une ligne blanche horizontale au milieu. Une fois encore, les neurones utilisant ces poids ignoreront tout ce qui se trouve dans leur champ réceptif, à l'exception de la ligne horizontale centrale. Maintenant, si tous les neurones d'une couche utilisent le même filtre de ligne verticale (et le même terme de biais), et que vous fournissez au réseau l'image d'entrée illustrée à la figure 4 (l'image du bas), la couche produira l'image du haut à gauche. Remarquez que les lignes blanches verticales sont renforcées alors que le reste est flou. De même, l'image en haut à droite est ce que vous obtenez si tous les neurones utilisent le même filtre de lignes horizontales ; remarquez que les lignes blanches horizontales sont renforcées alors que le reste est flou. Ainsi, une couche pleine de neurones utilisant le même filtre produit une carte de caractéristiques, qui met en évidence les zones d'une image qui activent le plus le filtre. Bien entendu, il n'est pas nécessaire de définir les filtres manuellement : au cours de l'apprentissage, la couche convolutive apprend automatiquement les filtres les plus utiles pour sa tâche, et les couches supérieures apprennent à les combiner en des modèles plus complexes.

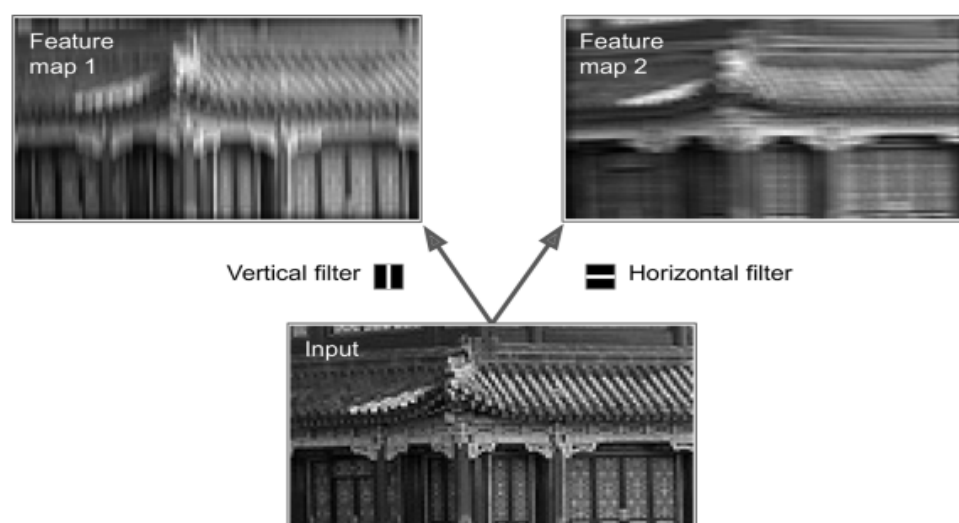


FIGURE 1.4 – Appliquer deux filtres différents pour obtenir deux cartes de caractéristiques

1.2.4 Pooling layer

Cette couche a pour but de sous-échantillonner l'image d'entrée afin de réduire la charge de calcul, l'utilisation de la mémoire et le nombre de paramètres (limitant ainsi le risque de sur-ajustement). Cette opération peut être effectuée par deux méthodes différentes : La première consiste à choisir la valeur maximale dans chacune des régions de calcul (couvertes par le filtre) de la matrice d'entrée, comme le montre la figure 1.5 :

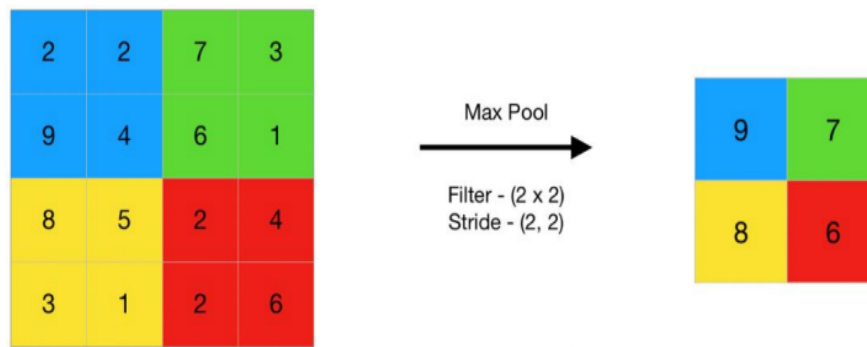


FIGURE 1.5 – Max Pooling

La seconde consiste à calculer la mise en commun moyenne de chaque région couverte de la matrice d'entrée, comme le montre la figure 1.6 :

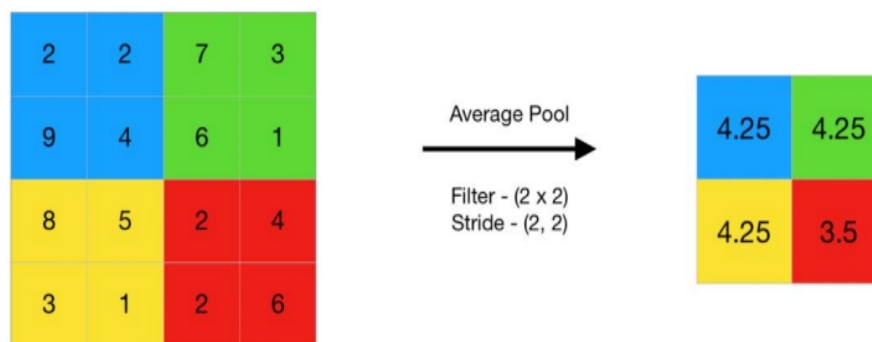


FIGURE 1.6 – Average Pooling

1.2.5 Fully connected Layer

À la fin d'un réseau neuronal convolutif, la sortie de la dernière couche sert d'entrée à la couche entièrement connectée. Il peut y avoir une ou plusieurs de ces couches. Fully connected signifie que chaque nœud de la première couche est connecté à chaque nœud de la deuxième couche comme dans la figure 1.7.

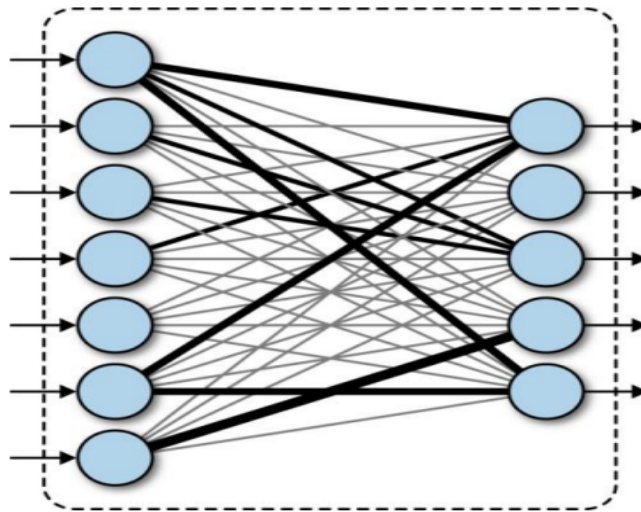


FIGURE 1.7 – Fully connected layer - la première couche est connecté à chaque nœud de la deuxième couche

1.3 Exemple d'architecture convolutive : LeNet-5

L'architecture de LeNet-5 se compose de deux ensembles de couches convolutives et de couches de average pooling, suivis d'une couche convolutive de flattening, puis de deux couches fully connected et enfin d'un classificateur softmax.

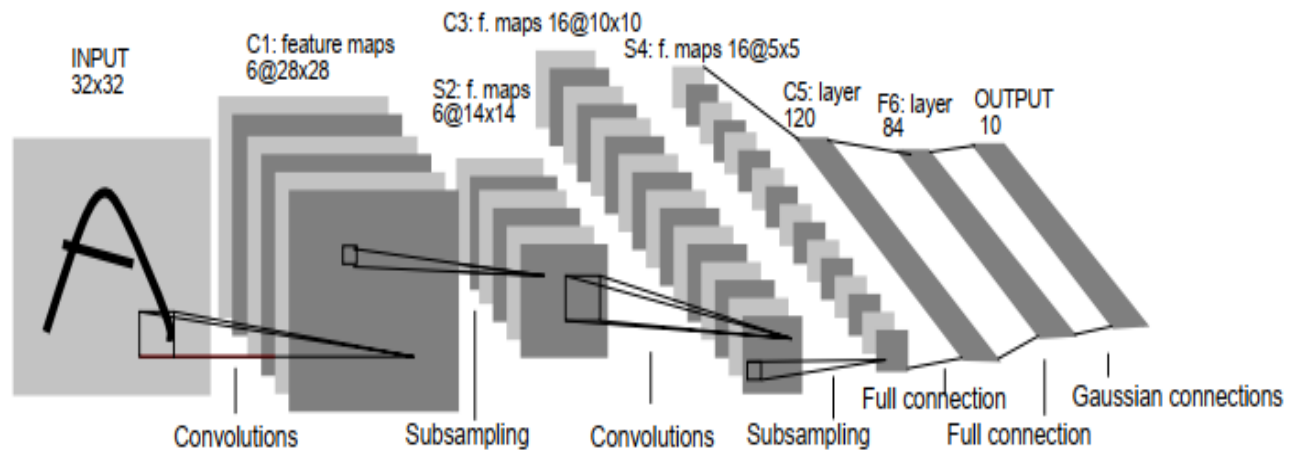


FIGURE 1.8 – Architecture de LeNet un réseau de neurones convolutif ici pour la reconnaissance de chiffres

La première couche est la couche d'entrée avec une carte de caractéristiques de taille 32X32X1. Ensuite, nous avons la première couche de convolution avec 6 filtres de taille 5X5 et un stride de 1. La fonction d'activation utilisée dans cette couche est tanh. La carte de sortie est de 28X28X6. Ensuite, nous avons une couche de average pooling avec des filtres de taille 2X2 et un stride de 1. La carte de caractéristiques résultante est de 14X14X6. Puisque la couche average pooling n'affecte pas le nombre de canaux. Après cela vient la deuxième couche de convolution avec 16 filtres de 5X5 et stride 1. De plus, la fonction d'activation est tanh. La taille de la sortie est maintenant de 10X10X16. Vient encore l'autre couche de average pooling de 2X2 avec stride 2. En conséquence, la taille de la carte de caractéristiques est réduite à 5X5X16. La couche de mise en commun finale comporte 120 filtres de 5X5 avec stride 1 et fonction d'activation tanh. La taille de la sortie est maintenant de 120. La couche suivante est une couche entièrement connectée avec 84 neurones qui donnent une sortie de 84 valeurs et la fonction d'activation utilisée ici est à nouveau tanh. La dernière couche est la couche de sortie avec 10 neurones et la fonction Softmax. La fonction Softmax donne la probabilité qu'un point de données appartienne à une classe particulière. La valeur la plus élevée est alors prédite. Voilà l'architecture complète du modèle Lenet-5. Le nombre de paramètres entraînables de cette architecture est d'environ soixante mille, cela est représenté en bref dans la figure 1.9.

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	
Conv 2	16	5 * 5	1	10 X 10 X 16	tanh
Avg. pooling 2		2 * 2	2	5 X 5 X 16	
Conv 3	120	5 * 5	1	120	tanh
Fully Connected 1	-	-	-	84	tanh
Fully Connected 2	-	-	-	10	Softmax

FIGURE 1.9 – Sommaire de l'architecture LeNet5

1.4 Aller plus profond avec CNN et la réduction des coûts de computation

1.4.1 apprentissage résiduel

L'ajout de couches supplémentaires créerait une fonction plus complexe, l'échec serait donc attribué à un surajustement. Si c'était le cas, des paramètres de régularisation supplémentaires et des algorithmes tels que le dropout ou les L2-norms seraient une approche efficace pour corriger ces réseaux. Cependant, le graphique montre que l'erreur d'apprentissage du réseau à 56 couches est plus élevée que celle du réseau à 20 couches, ce qui met en évidence un phénomène différent expliquant son échec.

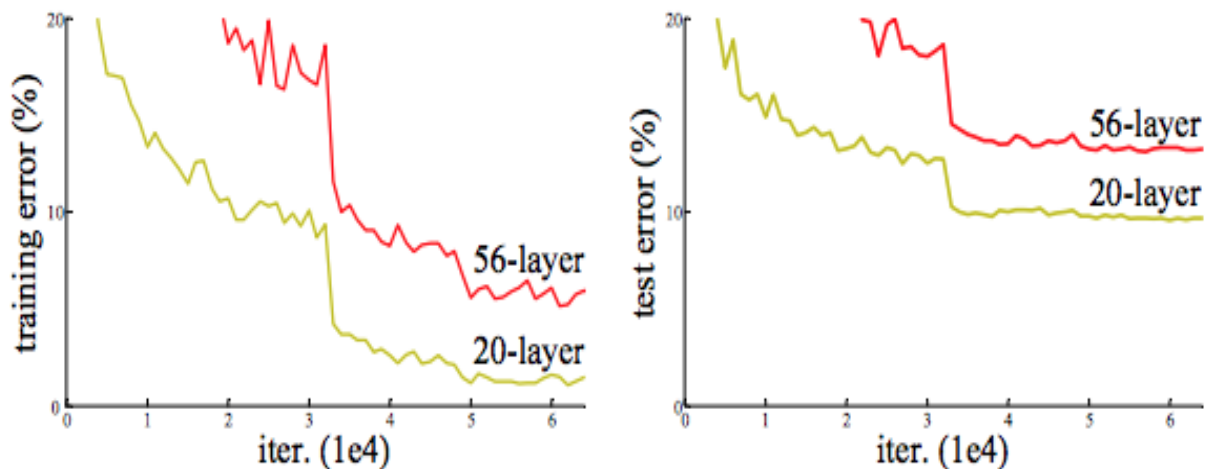


FIGURE 1.10 – Erreur de test et de training pour les les 20 layers et 56 layers

Le problème de la formation de réseaux très profonds a été atténué par l'introduction d'une nouvelle couche de réseau neuronal - le bloc résiduel comme dans la figure 1.11.

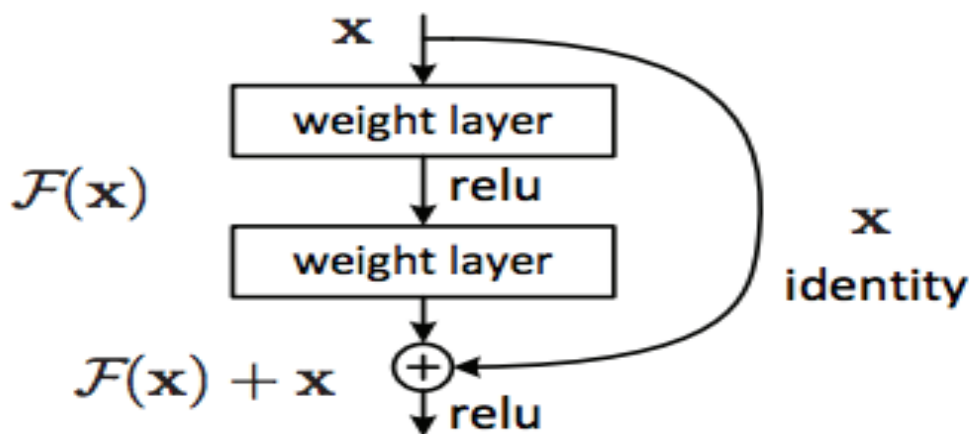


FIGURE 1.11 – Apprentissage résiduel : bloc résiduel

L'image ci-dessus est la chose la plus importante à apprendre. Pour les développeurs qui souhaitent mettre en œuvre et tester rapidement ce système, la modification la plus importante à comprendre est le mappage d'identité "Skip Connection". Ce mappage d'identité n'a pas de paramètres et est juste là pour ajouter la sortie de la couche précédente à la couche suivante. Cependant, il arrive que x et $F(x)$ n'aient pas la même dimension. Rappelons qu'une opération de convolution réduit généralement la résolution spatiale d'une image, par exemple, une convolution 3×3 sur une image 32×32 donne une image 30×30 . Le mappage d'identité est multiplié par une projection linéaire W pour étendre les canaux de raccourci afin de correspondre au résidu. Cela permet de combiner l'entrée x et $F(x)$ comme entrée de la couche suivante.

$$y = F(x, W_i) + W_s \times x \quad (1.2)$$

Équation 1.2 utilisée lorsque $F(x)$ et x ont une dimensionnalité différente, comme 32×32 et 30×30 . Ce terme W_s peut être implémenté avec des convolutions 1×1 , ce qui introduit des paramètres supplémentaires dans le modèle.

1.4.2 1x1 Convolution

La convolution 1×1 signifie simplement que le filtre est de taille 1×1 (Oui - cela signifie un seul nombre par opposition à une matrice comme, disons, un filtre 3×3). Ce filtre 1×1 va convoluer sur l'image d'entrée ENTIÈRE, pixel par pixel. En d'autres termes, la convolution 1×1 a été utilisée pour réduire le nombre de canaux tout en introduisant la non-linéarité.

Considérons maintenant des entrées avec un grand nombre de canaux - 192 par exemple. Si nous voulons réduire la profondeur tout en conservant la même hauteur \times largeur des cartes de caractéristiques (champ de réception), nous pouvons choisir des filtres 1×1 (rappelez-vous que le nombre de filtres = canaux de sortie) pour obtenir cet effet. Cet effet de sous-échantillonnage transversal est appelé "réduction de la dimensionnalité" comme il est montré dans la Figure 1.12.

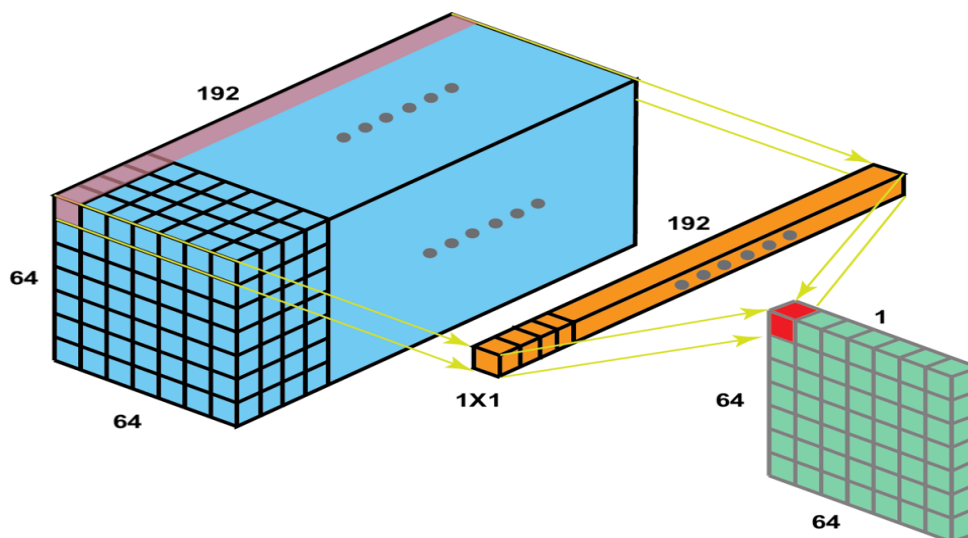
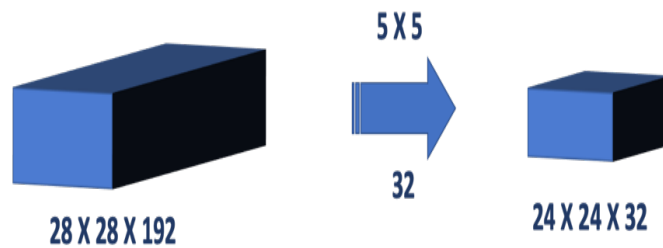


FIGURE 1.12 – Convolution 1x1

Utilisations de 1x1 convolution :

- **Utilisation 1** : Réduction/augmentation de la dimensionnalité :

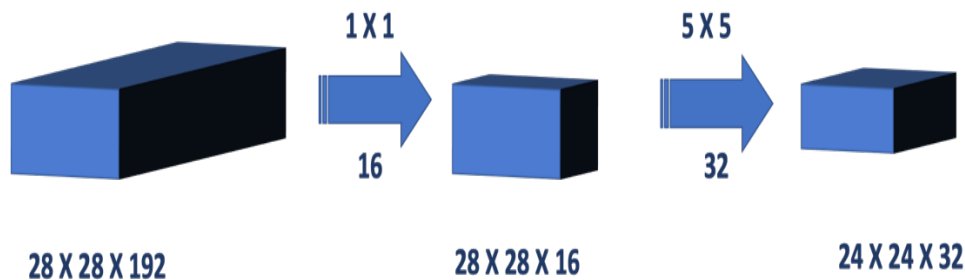
Prenons un exemple pour comprendre comment la réduction de la dimension permet de réduire la charge de calcul. Supposons que nous devions convoluer $28 \times 28 \times 192$ cartes de caractéristiques d'entrée avec $5 \times 5 \times 32$ filtres. Il en résultera 120,422 millions d'opérations' ;



Number of Operations : $(28 \times 28 \times 32) \times (5 \times 5 \times 192) = 120.422 \text{ Million Ops}$

FIGURE 1.13 – Appliquer un filtre 5×5

Faisons quelques calculs avec les mêmes cartes de caractéristiques d'entrée, mais avec une couche Conv 1×1 avant la couche Conv 5×5 .



Number of Operations for 1×1 Conv Step : $(28 \times 28 \times 16) \times (1 \times 1 \times 192) = 2.4 \text{ Million Ops}$

Number of Operations for 5×5 Conv Step : $(28 \times 28 \times 32) \times (5 \times 5 \times 16) = 10 \text{ Million Ops}$

Total Number of Operations = 12.4 Million Ops

FIGURE 1.14 – Réduction de la dimensionnalité en appliquant une convolution 1×1

En ajoutant une couche Conv 1×1 avant la Conv 5×5 , tout en conservant la hauteur et la largeur de la carte des caractéristiques, nous avons réduit le nombre d'opérations par un facteur de 10. Cela réduira les besoins en calcul et, par conséquent, sera plus efficace. Le document de GoogleNet décrit le module comme "Inception Module" (comprenez : DiCaprio dit "We need to go DEEPER" dans le film Inception).

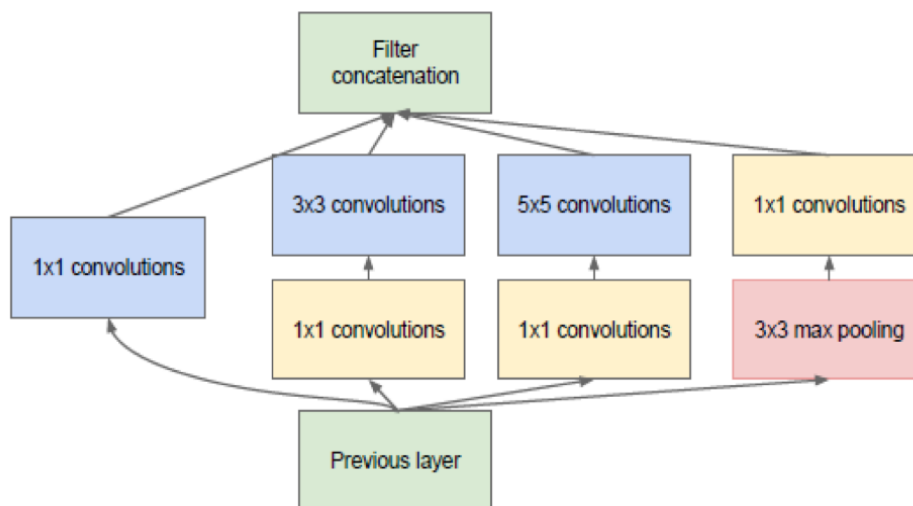


FIGURE 1.15 – Architecture Inception

■ **Utilisation 2** : Construire un réseau plus profond (couche "Bottle-neck") :

Le vainqueur de la classification ILSVRC 2015, ResNet, a obtenu le taux d'erreur le plus faible et a balayé la concurrence en utilisant un réseau très profond à l'aide de "connexions résiduelles" et d'une "couche Bottle-neck". Dans leur article, He et tous expliquent comment une couche à Bottle-neck a été conçue en utilisant une séquence de 3 couches convolutives avec des filtres de taille 1X1, 3X3, puis 1X1 respectivement pour réduire et restaurer la dimension. Le sous-échantillonnage de l'entrée a lieu dans la couche 1X1, ce qui permet d'obtenir des vecteurs caractéristiques plus petits (nombre réduit de paramètres) sur lesquels la couche 3X3 peut travailler. Immédiatement après, la couche 1X1 restaure les dimensions pour qu'elles correspondent à celles de l'entrée, ce qui permet d'utiliser directement les raccourcis d'identité. Pour plus de détails sur les raccourcis d'identité et la connexion par saut, veuillez consulter certaines des revues sur ResNet.

■ **Utilisation 3** : Depth-wise Separable Convolution :

Nous utilisons chaque canal de filtre uniquement sur un canal d'entrée. Dans l'exemple, nous avons des filtres à 3 canaux et des images à 3 canaux. Ce que nous faisons, c'est de décomposer le filtre et l'image en trois canaux différents, puis de convoluer l'image correspondante avec le canal correspondant, avant de les superposer à nouveau ; ensuite, nous utilisons un filtre 1*1 pour couvrir la dimension de profondeur.

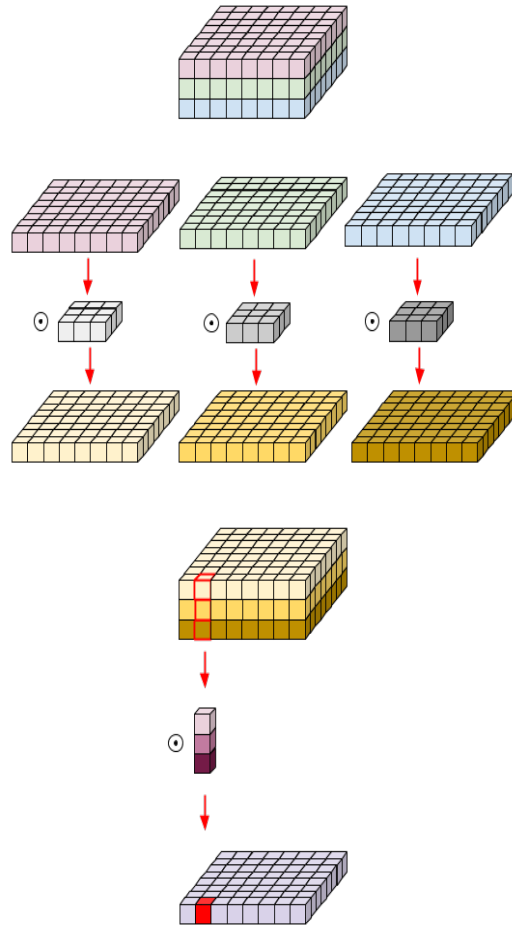


FIGURE 1.16 – La couche "Bottle-neck"

1.5 Transfer learning

1.5.1 Définition

L'apprentissage par transfert est un problème de recherche en apprentissage automatique qui se concentre sur le stockage des connaissances acquises lors de la résolution d'un problème et sur leur application à un problème différent mais connexe. Par exemple, les connaissances acquises en apprenant à reconnaître des voitures pourraient être appliquées à la reconnaissance de camions.

La figure 1.16 montre le contraste entre les processus d'apprentissage automatique classique et l'apprentissage par transfert. Comme nous pouvons le voir, l'apprentissage automatique classique tente d'apprendre chaque tâche disparate séparément avec un système d'apprentissage différent, tandis que l'apprentissage par transfert tente d'extraire les connaissances de tâches sources précédentes à des tâches cibles où ces dernières ont très peu de données étiquetées pour l'apprentissage supervisé. de données étiquetées pour l'apprentissage supervisé.

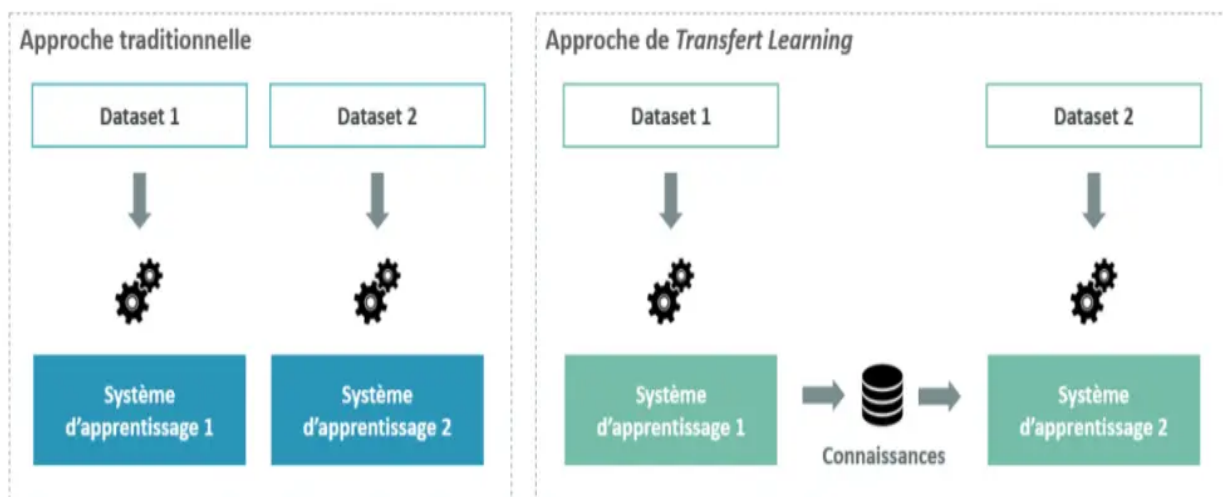


FIGURE 1.17 – Approche traditionnelle vs. Approche de Transfert Learning

1.5.2 Apprentissage par transfert avec des modèles prétraités

ImageNet est un projet de recherche visant à développer une grande base de données d'images avec des annotations, c'est-à-dire des images et leurs étiquettes. Des modèles pré-entraînés comme InceptionV1, Inception V2, VGG-16 et VGG-19 sont déjà entraînés sur ImageNet qui comprend des catégories d'images disparates. Ces modèles sont construits à partir de zéro et entraînés à l'aide de GPU puissants sur des millions d'images comprenant des milliers de catégories d'images. Comme le modèle est entraîné sur un énorme ensemble de données, il a appris une bonne représentation des caractéristiques de bas niveau comme l'espace, les bords, la rotation, l'éclairage, les formes et ces caractéristiques peuvent être partagées pour permettre le transfert de connaissances et agir comme un extracteur de caractéristiques pour de nouvelles images dans différents problèmes de vision par ordinateur. Ces nouvelles images peuvent être de catégories complètement différentes de l'ensemble de données source, mais le modèle pré-entraîné devrait toujours être capable d'extraire des caractéristiques pertinentes de ces images en se basant sur les principes de l'apprentissage par transfert. Dans cet article, nous allons exploiter la puissance de l'apprentissage par transfert en utilisant modèle pré-entraîné - VGG-16 comme un extracteur de caractéristiques efficace pour classer les chiens et les chats, même avec moins d'images d'entraînement.

Prenant l'exemple du modèle VGG16 préentraîné par la base de données ImageNET ; Dans la figure 1.18 ci-dessus, les couches convolutionnelles et pooling layers sont dégelées pour permettre le training. Une couche entièrement connectée est définie pour la formation et la prédiction

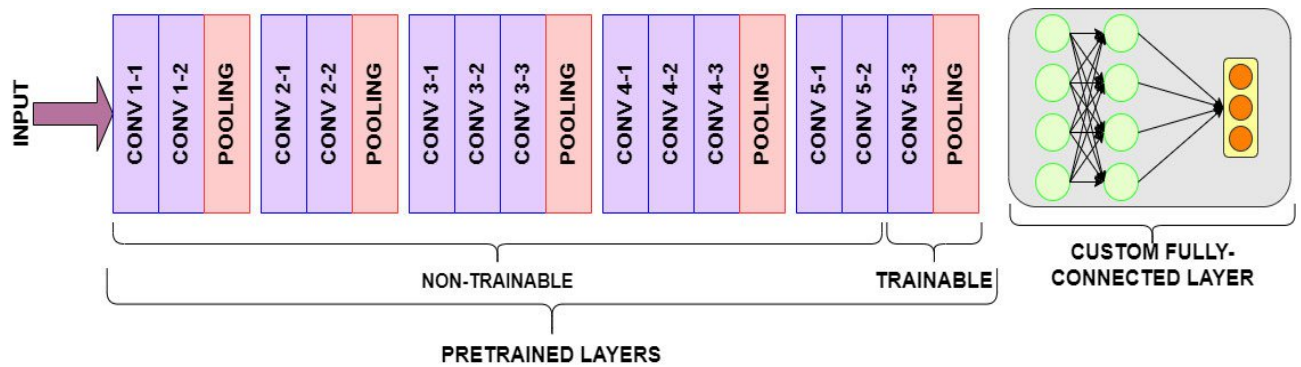


FIGURE 1.18 – Transfert Learning appliquée au modèle VGG16

1.6 Conclusion

D'énormes progrès ont été réalisés dans la reconnaissance d'images, principalement en raison de la disponibilité d'ensembles de données annotés à grande échelle (c.-à-d. ImageNet) et de la récente renaissance des réseaux de neurones convolutifs profonds (CNN). Pour l'apprentissage axé sur les données, des ensembles de données bien annotés à grande échelle avec des caractéristiques de distribution de données représentatives sont essentiels pour apprendre des modèles plus précis ou généralisables. En fin nous examinons quand et pourquoi l'apprentissage par transfert d'ImageNet préformé peut être utile. Nous étudions un problème spécifique de classification des maladies.

Cryptage des images numériques par les systèmes chaotiques

2.1 Introduction

Avec la progression continue de la cybercriminalité, la sécurité des images numériques est devenue un thème important dans le monde des communications. La révolution numérique a engendré des moyens plus faciles pour le traitement, le stockage et la transmission des images numériques. Cependant, elle a aussi engendré des moyens de falsification, de contrefaçons et d'espionnage très avancés. Le risque est encore plus grand dans un environnement ouvert tel que la transmission des images satellitaires. Dans ce chapitre on discute l'un des plus méthodes performantes de cryptage d'images, à savoir cryptage chaotique par logistic map.

2.2 Logistic map

Logistic map est une cartographie polynomiale (équivalente à une relation de récurrence) de degré 2, souvent citée comme un exemple archétypal de la manière dont un comportement complexe et chaotique peut découler d'équations dynamiques non linéaires très simples. La carte a été popularisée dans un article de 1976 du biologiste Robert May, en partie comme un modèle démographique en temps discret analogue à l'équation logistique écrite par Pierre François Verhulst. Mathématiquement, la carte logistique s'écrit :

$$x_{n+1} = rx_n(1 - x_n) \tag{2.1}$$

Où r est le paramètre de contrôle et n désigne le nombre d'itérations. Lorsque $r \in (3, 57, 4]$, le système de l'équation 2.1 présente un comportement chaotique représentée dans la figure 2.1.

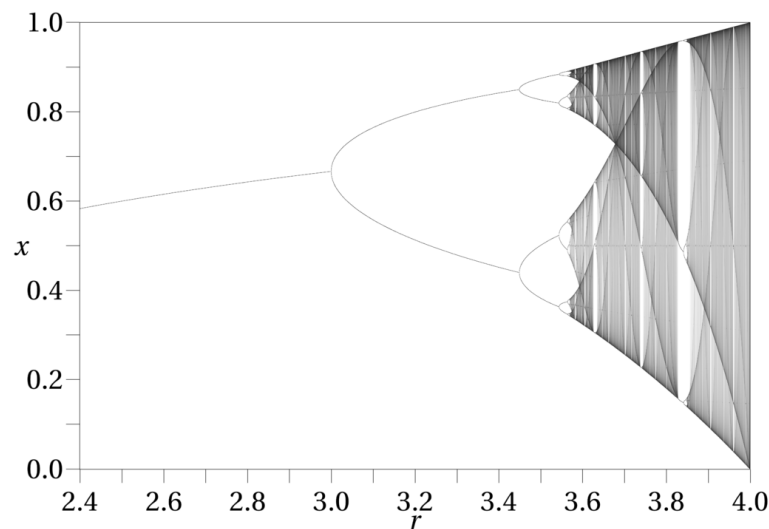


FIGURE 2.1 – Évolution de différentes conditions initiales en fonction de r

Les valeurs au-delà de 3,57 présentent un comportement chaotique, mais il existe encore certaines valeurs r isolées qui semblent présenter un comportement non chaotique, on les appelle parfois des îlots de r qui présentent une oscillation entre trois valeurs, et pour des valeurs légèrement plus élevées de r une oscillation entre 6 valeurs, puis 12, etc.

2.3 Algorithme de cryptage d'image basé sur logistic map

Pour une image originale donnée de taille $M \times N$, l'algorithme proposé de cryptage et de décryptage d'image illustré à la figure 2.2 par logistic map avec la valeur initiale r_0 de la logistic map dans l'équation (2.1).

L'image résultante peut être obtenue en effectuant un XOR de l'image originale avec la séquence aléatoire générée par logistic map.

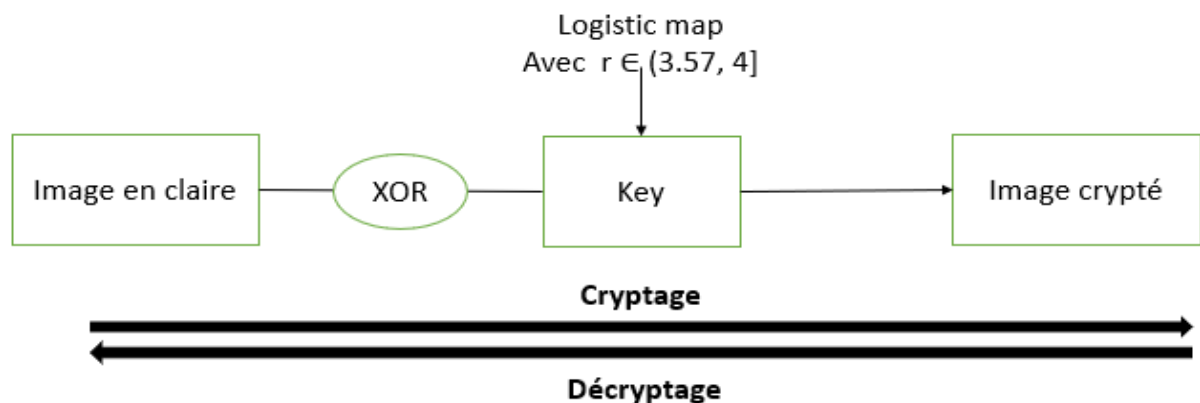


FIGURE 2.2 – Shéma de Cryptage et décryptage

Voici le code source en python :

```
In [ ]: import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
```

```
In [ ]: #generer la cle
def keygen(x,r,size):

    key=[]

    for i in range(size):
        x=r*x*(1-x)
        key.append(int((x*pow(10,16))%256))
    return key
```

```
In [ ]: #encryption

def encryption(img,x,r):
    height=img.shape[0]
    width=img.shape[1]
    key=keygen(x,r,height*width)
    z=0
    encrypted_img= np.zeros(shape=[height,width,3], dtype=np.uint8)
    for i in range(height):
        for j in range(width):
            #pixel xor key
            encrypted_img[i,j]= img[i,j]^key[z]
            z+=1

    plt.imshow(encrypted_img)
    plt.show()
    return encrypted_img
```

```
In [2]: #decryption

def decryption(img_c,x,r):
    height=img_c.shape[0]
    width=img_c.shape[1]
    key=keygen(x,r,height*width)
    z=0
    key=keygen(0.01,r,height*width)
    decrypted_img= np.zeros(shape=[height,width,3], dtype=np.uint8)
    for i in range(height):
        for j in range(width):
            #pixel xor key
            decrypted_img[i,j]= img_c[i,j]^key[z]
            z+=1

    return decrypted_img
```

Alors, le cryptage et le décryptage sont assurés par le schéma proposé. Une applica-

tion en python donne les résultats suivantes :

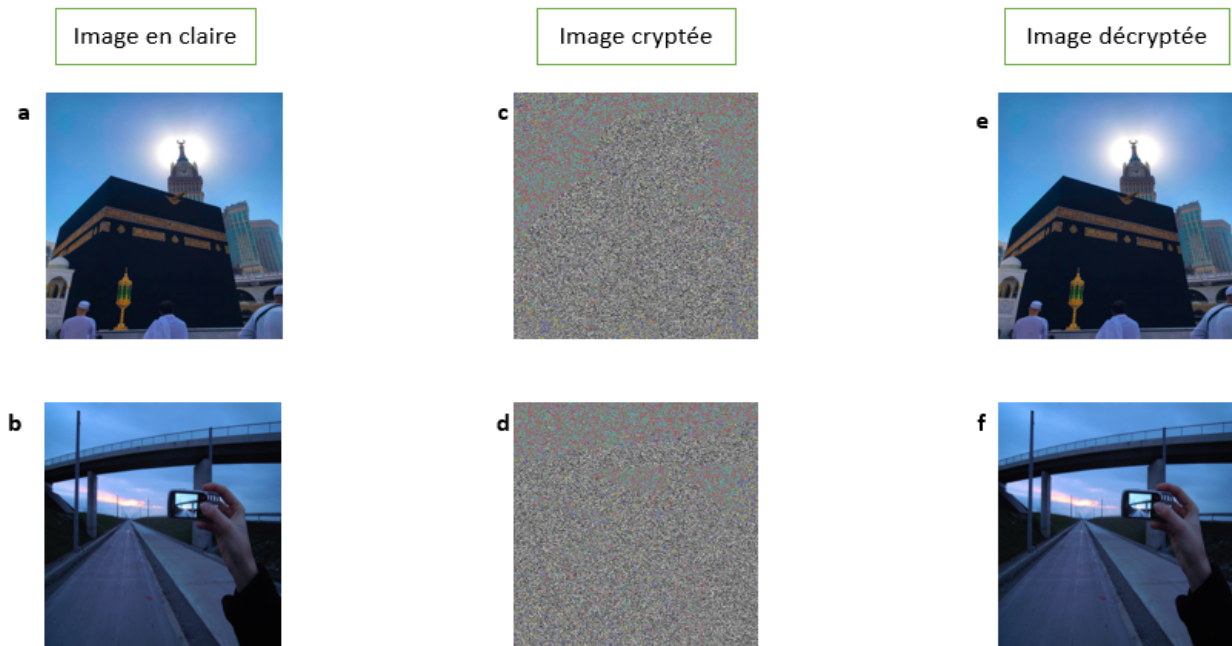


FIGURE 2.3 – Résultats des images de test

2.4 Simulation et discussion

Pour vérifier la faisabilité et l'efficacité du schéma de cryptage d'image proposé basé sur la carte logistique, des simulations sont effectuées avec l'image en clair de la figure 2.3.a et la seconde de la figure 2.3.b. La valeur initiale r_0 de la carte logistique est fixée à 3,951. Les figures 2.3.c et 2.3.d montrent les images en texte chiffré des figures 2.3.a et 2.3.b, respectivement. Les figures 2.3.e et 2.3.f sont les images déchiffrées des figures 2.3.a et 2.3.b avec les clés correctes, respectivement, à partir desquelles les différences entre l'image en texte clair et l'image déchiffrée ne peuvent être distinguées.

2.4.1 Histogramme

Il est normal pour nous d'analyser les performances du schéma de cryptage d'image avec l'histogramme de l'image. Pour un bon schéma de cryptage d'image, les histogrammes des images cryptées sont assez uniformes ou similaires dans leur distribution. Les figures 2.4.a et 2.4.b sont les histogrammes des figures 2.3.a et 2.3.b, respectivement. Les figures 2.4.c et 2.4.d montrent les histogrammes des images cryptées correspondantes.

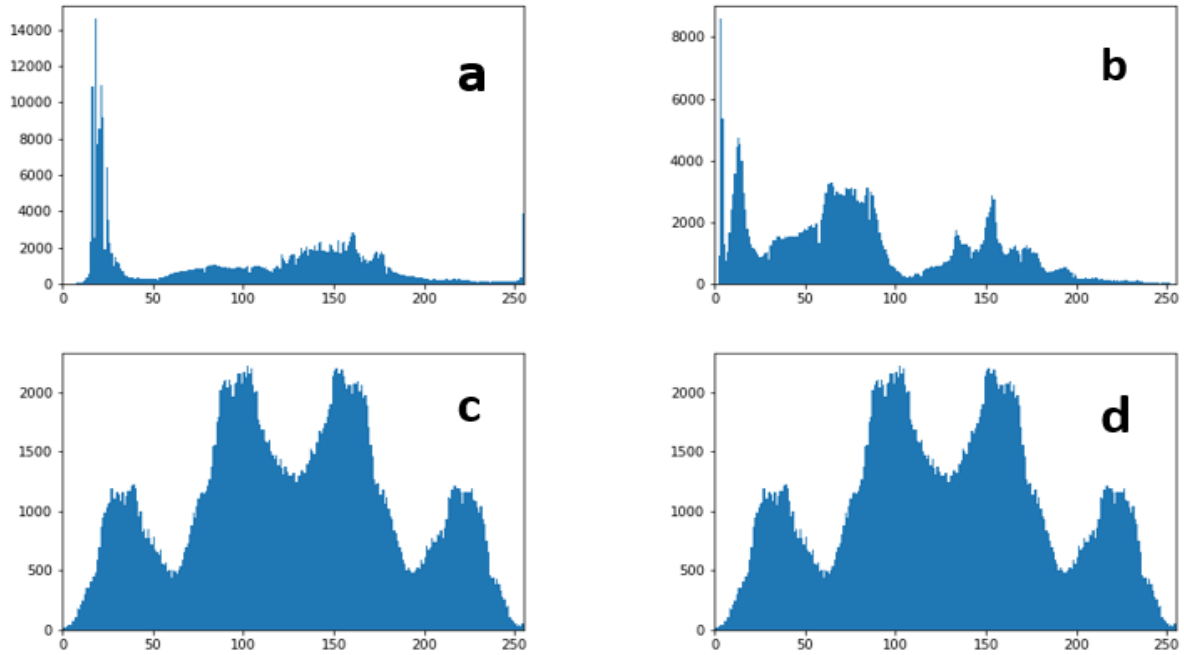


FIGURE 2.4 – les histogrammes : image a ; image b ; image cryptée c de a ; image cryptée d de b

D'après la figure 2.4, les histogrammes des images originales sont apparemment différents les uns des autres, tandis que les images cryptées présentent des propriétés statistiques similaires. Les histogrammes de l'image cryptée sont soumis à une distribution normale approximative. Ainsi, il ne fournit aucune information utile à l'adversaire pour effectuer une attaque d'histogramme sur l'algorithme de cryptage d'image.

2.4.2 Sensibilité de la clé

La fonction suivante d'erreur quadratique moyenne relative (RMSE) est généralement employée pour exprimer quantitativement la différence entre les images décryptées et les images originales.

$$RMSE = \log \frac{\sum_{x=1}^M \sum_{y=1}^N [f(x, y) - \bar{f}(x, y)]^2}{M \times N} \quad (2.2)$$

où $f(x, y)$ et $\bar{f}(x, y)$ désignent les valeurs des pixels de l'image originale et de l'image décryptée, respectivement ; $M \times N$ désigne la taille de l'image.

La sensibilité peut également être testée en observant les changements des images de décryptage lorsque les clés correctes changent légèrement. Les figures 2.3 représentent le RMSE de l'image originale et l'image cryptée en changeant le paramètre de contrôle de la logistic map.

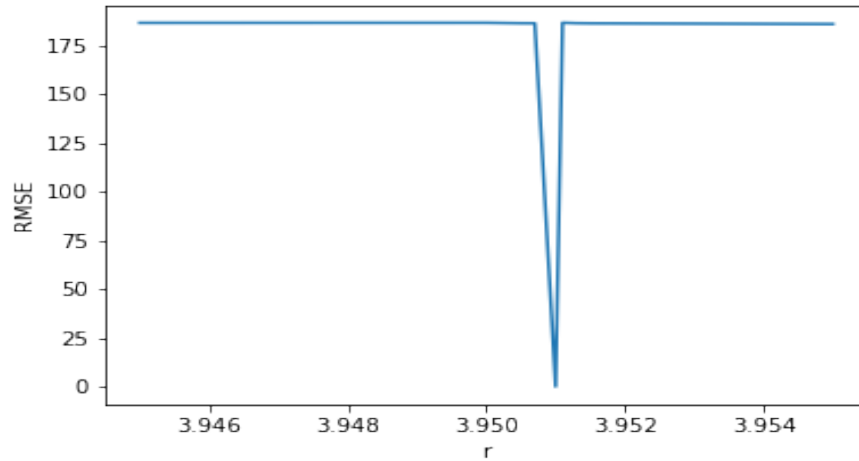


FIGURE 2.5 – RMSE de l'image originale et l'image cryptée en fonction de r

Apparemment, les images décryptées avec de mauvaises clés sont incapables de fournir des informations significatives sur les images originales. Les détails des images originales deviennent flous lorsque la clé de décryptage s'écarte de la clé de cryptage et la variation des images décryptées est invisible visuellement si une légère déviation se produit au niveau des clés principales. Par conséquent, l'algorithme de cryptage d'images proposé présente une sécurité élevée.

2.5 Conclusion

Le cryptage chaotique par logistic map a montré son efficacité en terme de clé ce qui nous motive à le déployer dans notre projet.

OCT diagnostic

3.1 Introduction

On a vu dans les chapitres précédentes les deux aspects de notre projet : "DeepLearning pour la classification" et cryptage des images. Dans ce chapitre on va voir les résultats dans notre projets et les implémentées dans une application web.

3.2 Présentation du projet

3.2.1 Contexte du projet

Notre projet porte sur la classification des troubles réiniens à partir d'images OCT. Parce que l'algorithme que nous proposons permet de bénéficier automatiquement et rapidement du diagnostic et de l'analyse de multiples anomalies réiniennes sans perdre beaucoup de temps tout en s'appuyant sur les techniques d'intelligence artificielle, notamment le deep learning, il représente un support très important pour les ophtalmologistes.

L'application est destiné dans la détection de la néovascularisation chorôidienne (NVC), de l'œdème maculaire diabétique (OMD) et des DRUSEN multiples présents dans la DMLA précoce.

Il bien de noter que ces maladies présentent un grand risque sur l'acuité visuelle chez les patients, en conduisant à une perte de la vision et à une déficience visuelle chez les adultes, une enquête récente a déclaré qu'environ 7,7 millions d'Américains sont atteints de rétinopathie diabétique et que, parmi eux, environ 750 000 sont également atteints de DME de type "2".

3.2.2 Problématique

La rétine est un organe sensible et situé au fond de l'œil, elle est responsable de capturer les rayons lumineux issues de la cornée et du cristallin, elle permet de transmettre la lumière au cerveau via le nerf optique afin de construire l'image et l'analyser. Les maladies réiniennes présentent la principale cause de basse vision.

Certes maintenant on a des appareils qui permettent de prendre les images en différentes coupes pour visualiser la forme de la rétine, mais le diagnostic de ces pathologies peut prendre beaucoup du temps. Devant ce problème, on a fait un système de classification et de détection de ces maladies d'une façon rapide et automatique, ce qui réduira le temps et l'effort de l'ophtalmologue et améliorera la précision.

3.2.3 Objectif

L'objectif principal de ce projet est de mettre en place un algorithme qui est capable de détecter et classer les 3 trois maladies rétinienues en se basant sur l'apprentissage profond et l'apprentissage transféré, et implémenter ce programme dans un interface graphique pour faciliter l'utilisation toute en respectant la confidentialité des information du patient.

3.2.4 Pathologies maculaires : classes traitées

Nos yeux peuvent être touchés par de nombreuses maladies dont la plupart apparaissent naturellement avec l'âge comme la cataracte, le glaucome ou la dégénérescence maculaire. Les maladies oculaires par définition sont des maladies affectant une ou plusieurs parties de l'œil : rétine, cornée, cristallin, conjonctive, etc.). Le glaucome, la cataracte et la DMLA font partie des pathologies ophtalmiques les plus connues.

Dans cette partie, et au tout long de ce projet, on s'intéresse seulement aux anomalies rétinienues qui feront l'objet de notre travail.

L'œdème maculaire diabétique(OMD)

Il est connu comme une rupture de la barrière hémato-rétinienne, qui est la principale altération structurelle responsable de la maladie. L'environnement diabétique endommage de nombreuses protéines de la jonction serrée, entraînant une hyperperméabilité et une fuite vasculaire. Il s'agit de la maladie oculaire diabétique la plus courante, qui entraîne une perte de la vision et une déficience visuelle chez les adultes. Une étude récente a révélé qu'environ 7,7 millions d'Américains souffrent de rétinopathie diabétique et que, parmi eux, environ 750 000 sont également atteints d'OMD. 750 000 d'entre eux sont également atteints d'OMD de type "2".

La néovascularisation choroïdale (NVC)

NVC est caractérisée par le développement de nouveaux vaisseaux sanguins, qui sont anormaux et permettent aux fluides et aux globules rouges de pénétrer dans la rétine, ce qui déforme la vision en formant une "vésicule", qui est normalement plate. Selon les résultats, la cause la plus fréquente de cécité chez les personnes âgées dans le monde occidental est la néovascularisation choroïdienne liée à la dégénérescence maculaire liée à l'âge (DMLA).

La DRUSEN du nerf optique

DRUSEN est l'accumulation anormale d'un matériau protéique situé entre la lamelle basale de l'épithélium pigmentaire rétinien (EPR) et la couche collagénique interne de la membrane de Bruch. Elle représente le stade précoce de la dégénérescence maculaire liée à l'âge de la partie antérieure du nerf optique.

Les ophtalmologues utilisent la technique de la tomographie de cohérence optique, pour détecter et diagnostiquer ces maladies.

3.3 Matériel et méthodes

3.3.1 Base de données

Dans ce travail, nous avons utilisé une base de données publique contenant des milliers d'images OCT validées provenant du travail de Kermany et al.2018[13]. Il y avait 207 130 images OCT collectées pour la première fois, mais seulement 84483 images provenant de 4686 patients ont été utilisées pour entraîner le modèle, grâce à leur qualité. Les images sont en 4 classes : rétine saine, œdème maculaire diabétique (OMD), néovascularisation choroïdienne (NVC), et DRUSEN multiple présent dans la DMLA précoce. L'ensemble de données est collecté auprès de patients présentant des caractéristiques différentes. Le tableau 2 montre la distribution des patients qui ont été inclus dans cette étude.

Ce tableau représente la base de données et ses caractéristiques

TABLE 3.1 – La base de données OCT				
Diagnosis	OMD	CNV	DRUSEN	NORMAL
Nombre des patients	709	791	713	3548
moyenne d age	57 (Range : 20-90)	83 (Range : 58-97)	82 (Range : 40-95)	60(Range : 21-86)
genre				
masculin	38.30%	54.20%	44.40%	59.20%
feminin	61.70%	45.80%	55.60%	40.80%
éthincité				
Caucasian	42.60%	83.30%	85.20%	59.90%
Asian	23.40%	6.30%	8.60%	21.10%
ispanian	23.40%	8.30%	4.20%	10.20%
African Amrican	4.30%	2.10%	1.90%	1.40%
Mixe ou autres	10.60%	0%	0%	7.50%

3.3.2 Méthode proposée

Afin de classifier les trois maladies courantes de la rétine, sur la base d'images de tomodographie par cohérence optique, nous proposons deux architectures VGG16 et InceptionV3, avec une technique d'apprentissage par transfert, ces modèles sont pré-entraînés sur plus d'un million d'images de la base de données ImageNet, ils ont donc appris suffisamment de caractéristiques pour qu'ils puissent détecter les objets préliminaires.

3.3.3 Matériel pour l'implémentation du modèle

- Framework

Après avoir sélectionné le modèle, nous présentons dans cette section les différentes façons d'implémenter celui-ci et surtout les outils afin de lancer l'étape d'entraînement et de prédiction. En effet, un panorama des frameworks et technologie est essentielle pour choisir le système le plus adéquat. Il existe beaucoup de framework. Certains sont opensource, d'autre pas. Dans ce projet, on a choisi d'utiliser TensorFlow comme une framework. Puisqu'il présente divers avantages dans la détection des objets.



- **Langage de programmation**

On a utilisé Python version 3.6, on était basé sur Google Colab Gratuite comme un éditeur de texte ou on avait l'opportunité de compiler notre programme avec des configurations suffisante RAM : 12 Go et un processus GPU.



3.3.4 Méthode de classification

- **Répartition de la base de données :**

Avant d'entamer l'entraînement des deux modèles VGG16 et Inception, on a pris 1000 images pour chaque classe, ensuite on a fixé 80% des images pour l'entraînement et 20% pour la phase de validation. Le tableau au-dessous indique le nombre des images utilisées.

TABLE 3.2 – Séparation de la base de données

Classes	Entrainement	Validation
Normale	800	200
Œdème maculaire diabétique (OMD)	800	200
Néovascularisation choroïdale (NVC)	800	200
DRUSEN multiples multiple présent dans la DMLA précoce	800	200

On a fait aussi dans le cadre de pré-entraînement, le redimensionnement de l'image, on a choisi $(224 \times 224 \times 3)$ car les deux modèles requièrent cette dimension.

- **L'entraînement de VGG16 et InceptionV3 :**

La figure 3.1 montre une architecture simplifiée du réseau neurone que nous proposons, deux architectures différentes, VGG16 et InceptionV3, sont utilisées pour extraire des caractéristiques profondes pour la classification ; elles sont basées sur des paramètres initiaux obtenus à partir du pré-entraînement sur la base de données Imagenet.

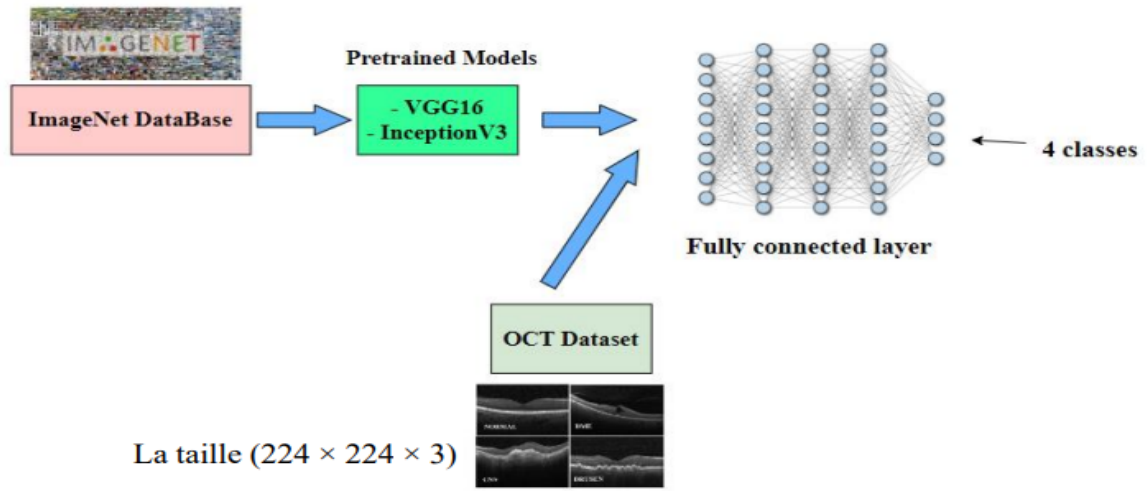


FIGURE 3.1 – Schéma d'entraînement du modèle

3.4 Résultats et Discussion

3.4.1 Critères d'évaluation

L'exactitude, le rappel et la précision sont des indicateurs importants qui doivent être pris en compte dans l'évaluation de chaque modèle, puisque les erreurs de diagnostic ou les diagnostics manqués peuvent avoir une influence majeure sur les patients. Ces taux ont été utilisés comme critères d'évaluation pour obtenir la matrice de confusion et la courbe ROC [16]. Et ils sont définis comme suit :

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \quad (3.1)$$

$$Precision = \frac{VP}{VP + FP} \quad (3.2)$$

$$Recall = \frac{VP}{VP + FN} \quad (3.3)$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3.4)$$

Avec VP sont les vrais positifs, c'est-à-dire les observations qui ont été classées comme positives et qui le sont. FP sont les faux positifs, c'est-à-dire les individus classés comme positifs et qui sont en fait des négatifs. De la même manière, FN sont les faux négatifs et VN sont les vrais négatifs. L'exactitude est le rapport entre le nombre d'échantillons classés sans erreur et le nombre total d'échantillons dans l'ensemble de données. La précision est définie comme le nombre d'échantillons positifs correctement classés divisé par le nombre total d'échantillons positifs classés (corrects ou incorrects). Le ratio de rappel indique la proportion entre le total des échantillons positifs qui ont été classés avec succès comme échantillons positifs et l'ensemble des échantillons positifs. Pour réaliser notre modèle, qui est considéré comme un problème de multi-classification, nous pouvons le transformer en une classification binaire pour chaque classe.

3.4.2 Résultats de la méthode proposée

La méthode proposée pour la classification des anomalies rétinienues est basée sur l'utilisation de deux modèles pré-entraînés, "InceptionV3" et "VGG16", et sur la technique d'apprentissage par transfert, en utilisant 1000 images OCT dans chaque classe durant la phase 45 d'entraînement.

La figure 3.2 montre l'exactitude de l'apprentissage et de la validation en fonction des itérations du modèle VGG16 finement ajusté ; on remarque que la courbe se converge après 15 epochs, de même que la variation de l'erreur a une valeur de convergence après 40 epochs. Nous avons obtenu une exactitude de 83 %. La figure 3.4 montre les paramètres de performance du modèle VGG16, nous pouvons facilement dénoter que la meilleure précision est obtenue pour DME avec 89 %.

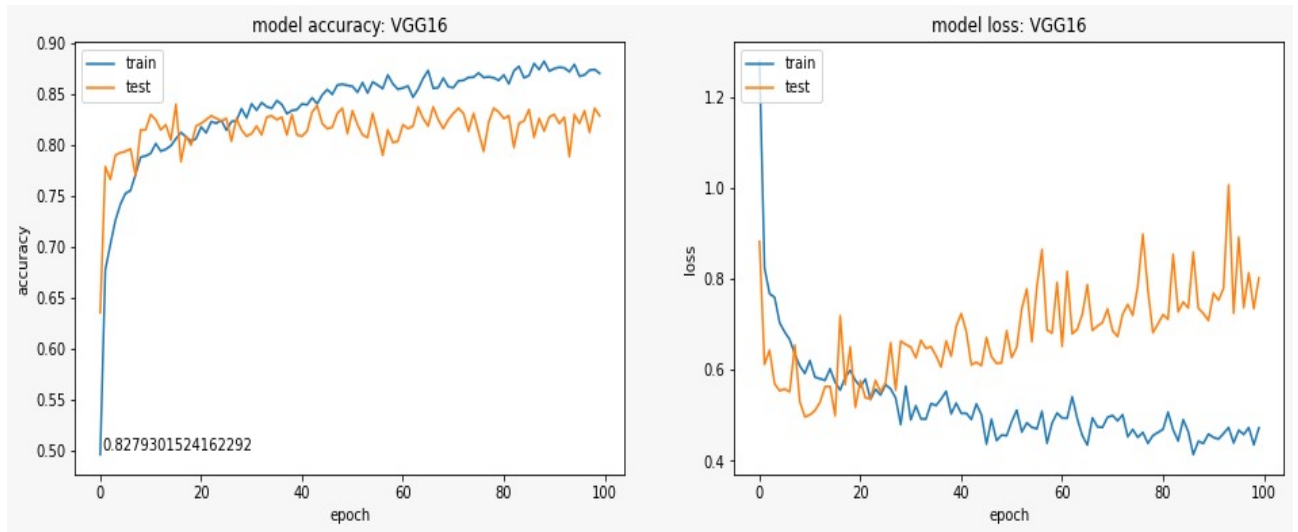


FIGURE 3.2 – Courbes de l'exactitude et de l'erreur pendant l'entraînement et la validation : modèle VGG16

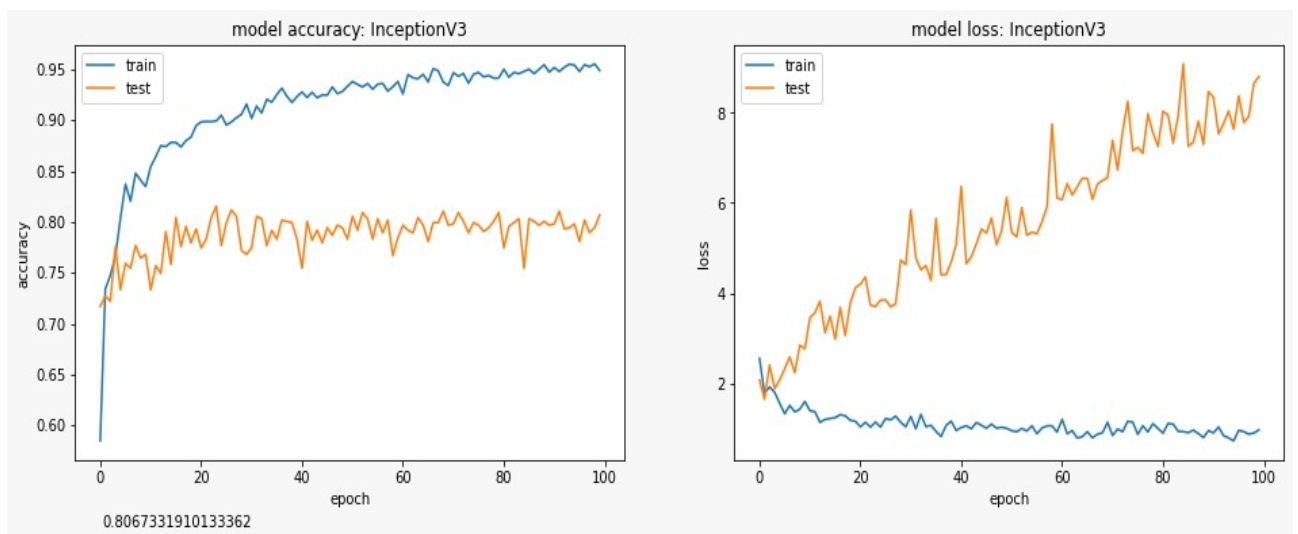


FIGURE 3.3 – Courbes de l'exactitude et de l'erreur pendant l'entraînement et la validation : modèle InceptionV3

```
Keras CNN - accuracy: 0.8279301524162292
```

	precision	recall	f1-score	support
Normal	0.81	0.90	0.85	182
CNV	0.76	0.90	0.83	186
DME	0.89	0.80	0.84	232
DRUSEN	0.86	0.73	0.79	202
accuracy			0.83	802
macro avg	0.83	0.83	0.83	802
weighted avg	0.83	0.83	0.83	802

FIGURE 3.4 – Paramètres de l'évaluation : VGG16

```
Keras CNN - accuracy: 0.8067331910133362
```

	precision	recall	f1-score	support
Normal	0.71	0.91	0.80	184
CNV	0.89	0.81	0.85	201
DME	0.86	0.80	0.83	198
DRUSEN	0.80	0.72	0.76	219
accuracy			0.81	802
macro avg	0.81	0.81	0.81	802
weighted avg	0.82	0.81	0.81	802

FIGURE 3.5 – Paramètres de l'évaluation : InceptionV3

Les mêmes paramètres et réglages ont été mis en œuvre dans InceptionV3, on peut noter sur la Figure 3.5, que nous avons obtenu une précision de 80.67%, et nous avons obtenu la meilleure précision pour les classes CNV (environ 89%). Mais nous pouvons voir sur la Figure 3.3 que les courbes de validation et l'apprentissage ont une valeur de convergence après 80 itérations, et la variance entre les deux courbes est remarquable, ce qui prouve l'existence de phénomène d'overfitting contrairement au modèle VGG16.

La Figure 3.6 et la Figure 3.7 montrent la matrice de confusion des deux modèles VGG16 et InceptionV3.

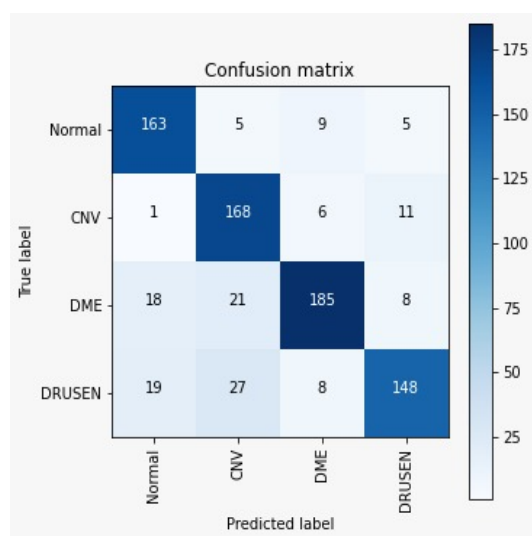


FIGURE 3.6 – Matrice de Confusion VGG16

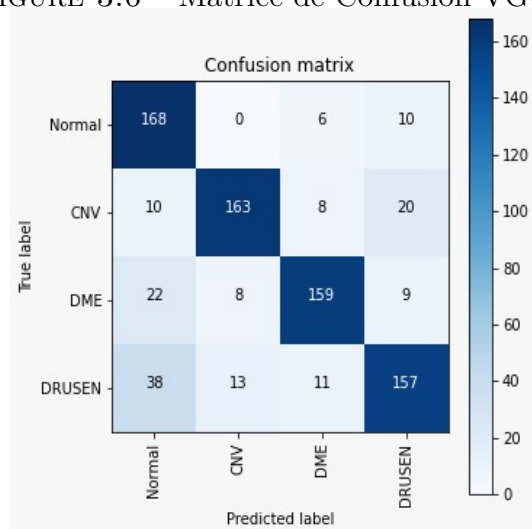


FIGURE 3.7 – Matrice de Confusion InceptionV3

3.5 Implémentation de la solution

3.5.1 Présentation de l'application

Après la discussion de différentes méthodes et modèles pour la classification des maladies et de cryptage d'images on s'intéresse maintenant à implémenter les meilleurs résultats dans notre application web.

La figure suivante explique la démarche utilisée :

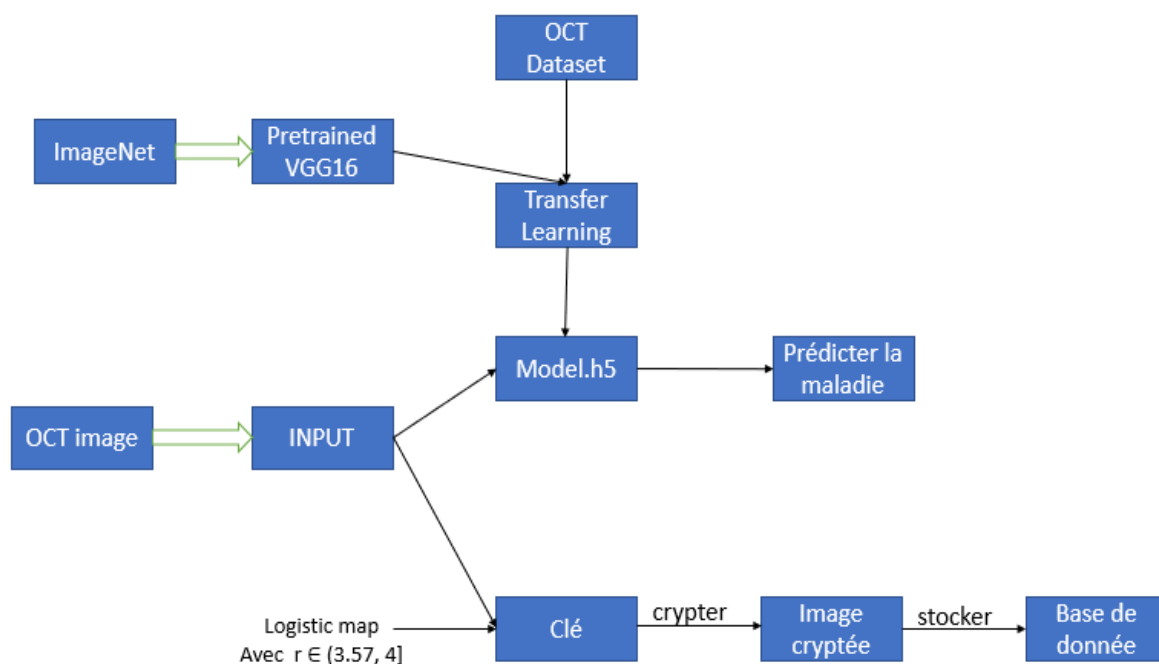


FIGURE 3.8 – Schéma d'implémentation d'application

3.5.2 Technologies Utilisées

— Django



Django est un framework web Python de haut niveau qui permet de développer rapidement des sites web sécurisés et faciles à maintenir. Conçu par des développeurs expérimentés, Django se charge d'une grande partie des tâches fastidieuses liées au développement Web, ce qui vous permet de vous concentrer sur l'écriture de votre application sans avoir à réinventer la roue. Il est gratuit et open source, possède une communauté active et florissante, une excellente documentation et de nombreuses options d'assistance gratuite et payante.

Django s'inspire du modèle MVC, c'est-à-dire que la structure du framework sépare les données (models) qui sont séparées des traitements (controller) qui sont eux-mêmes séparés de la vue (view / template). On vous oblige à bien coder, une structure doit être respectée et cela ne peut être que profitable au travail collaboratif ou simplement la cohérence / communication entre différents projets.

— SQLite



SQLite est une bibliothèque en langage C qui implémente un moteur de base de données SQL petit, rapide, autonome, très fiable et complet. SQLite est le moteur de base de données le plus utilisé au monde. SQLite est intégré à tous les téléphones mobiles et à la plupart des ordinateurs, ainsi qu'à d'innombrables autres applications que les gens utilisent tous les jours.

Le format de fichier SQLite est stable, multiplateforme et rétrocompatible. Les fichiers de base de données SQLite sont couramment utilisés comme conteneurs pour transférer des contenus riches entre systèmes et comme format d'archivage à long terme des données. Il y a plus d'un trillion de bases de données SQLite en utilisation active.

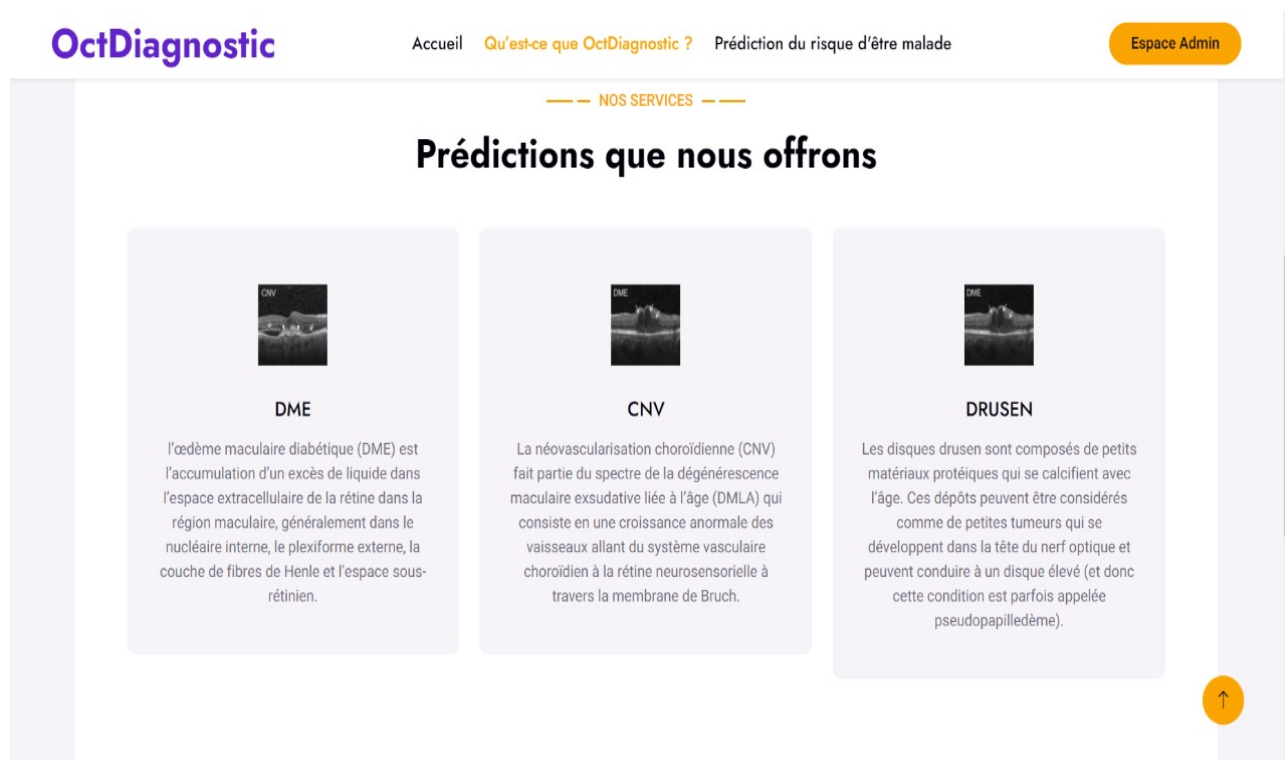
Le code source de SQLite est dans le domaine public et est libre d'utilisation pour tous.

3.5.3 Guide d'utilisation de l'application web

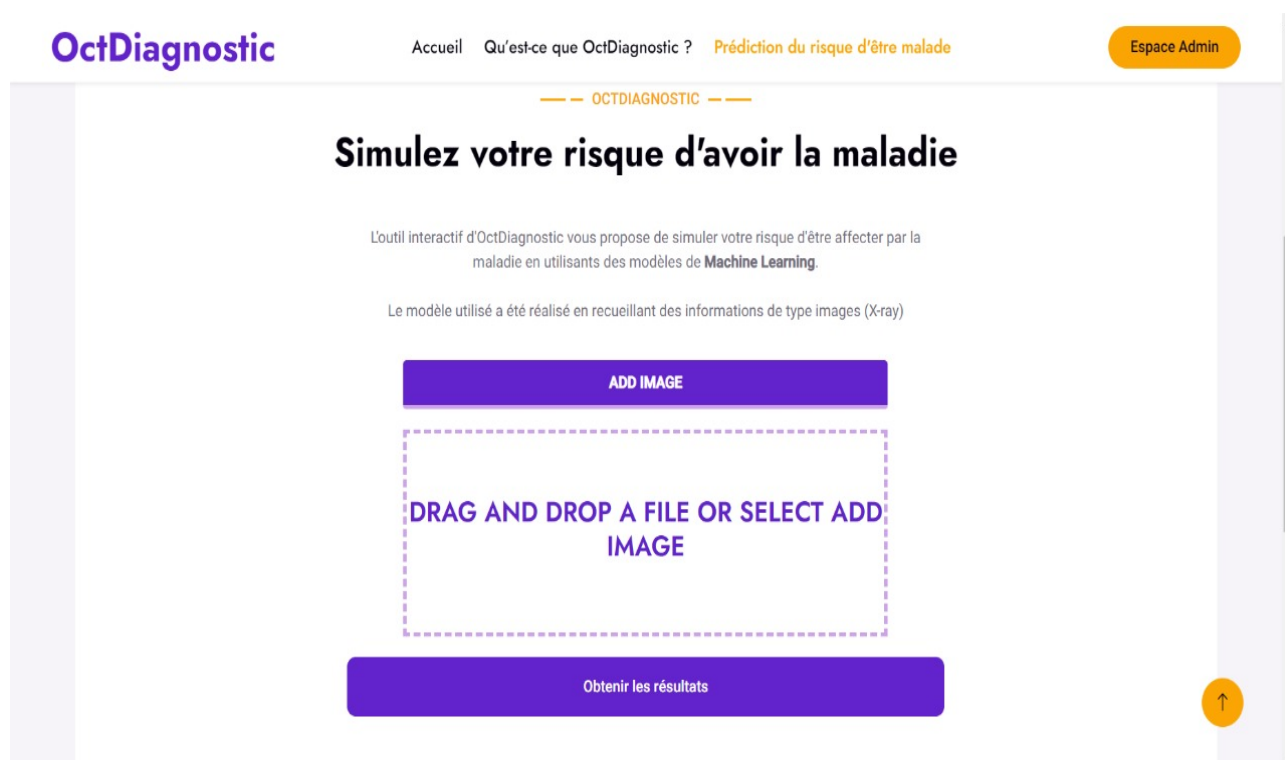
Page d'accueil :



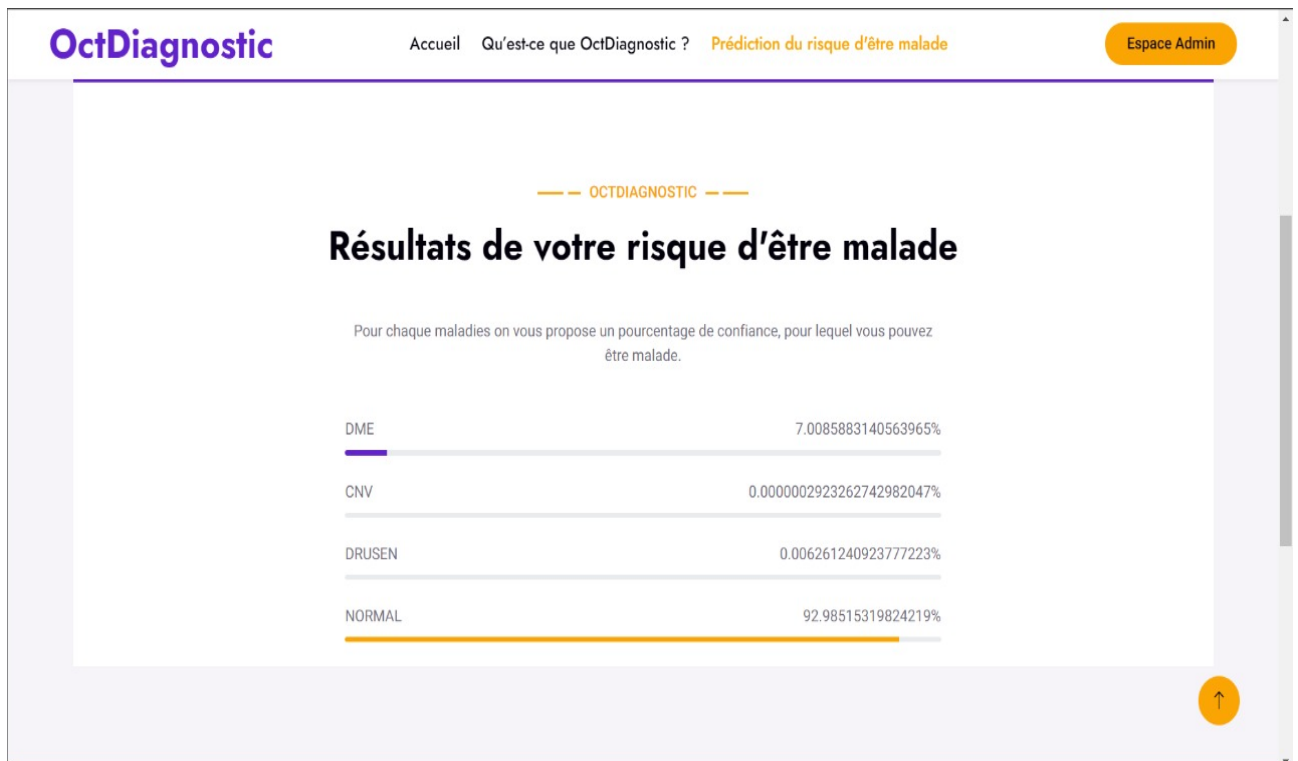
Page About : Qu'est-ce que OCTDiagnostic ? : contenant des informations sur les maladies de la rétine



Page de prédiction de la maladie : Saisi l'image OCT

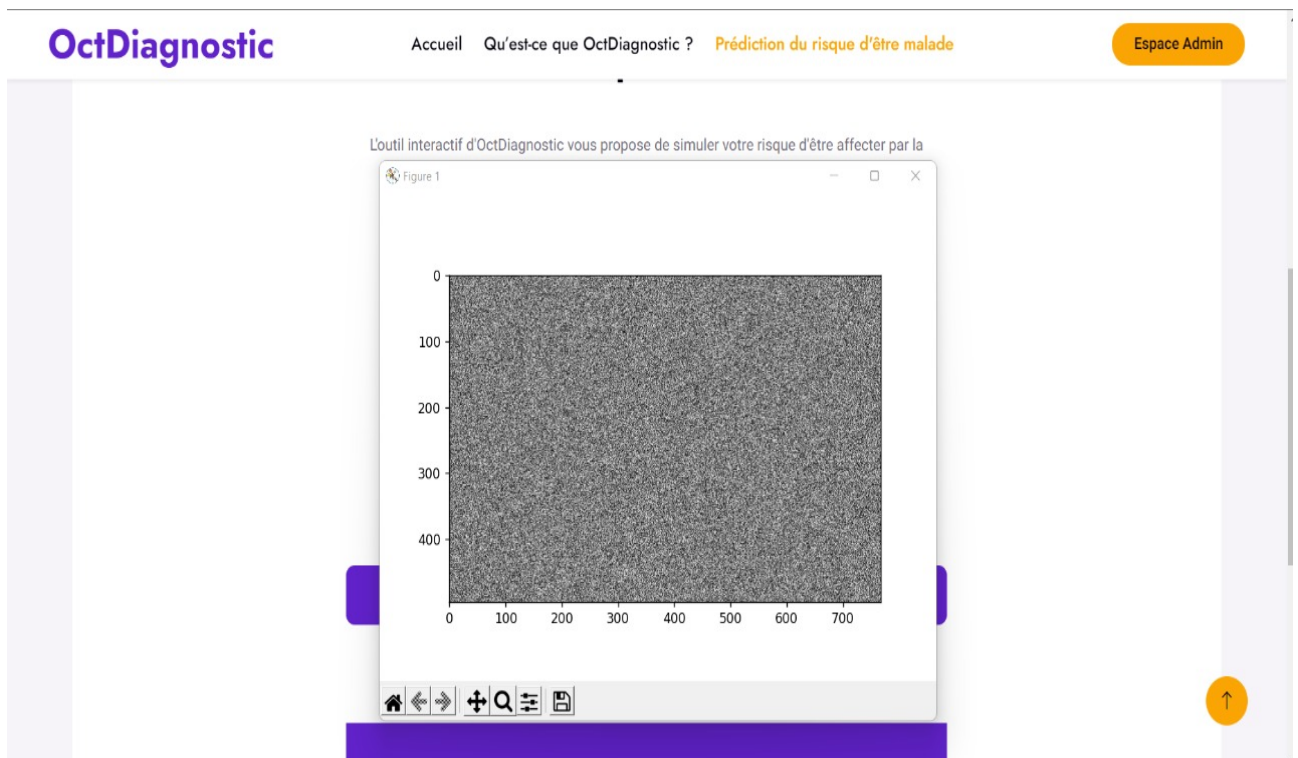


Page des résultats de prédiction de maladie de la rétine



Confidentialité des patient : Affichage de l'image OCT cryptée

Notre site web a respecté la confidentialité du patient puisqu'il utilise l'algorithme de cryptage par logistic map dans le chapitre 2. il crypte les entrées avant le stockage dans la base de données.



3.6 Conclusion

L'application web OCTDiagnostic est simple à utiliser. Vous pouvez prédire la maladie de la rétine en une clique avec confidentialité à cause du cryptage chaotique.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Pour diverses raisons, les troubles réiniens sont la principale cause de déficience visuelle. Nous avons tenté de proposer une technique qui permettrait aux ophtalmologistes de bénéficier d'une identification automatisée de ces troubles, notamment la néovascularisation choroïdienne, l'œdème maculaire diabétique (OMD) et les nombreux drusens prévalant dans la DMLA précoce. Cet outil est basé sur un algorithme d'apprentissage profond qui utilise des réseaux neuronaux convolutifs pour classer différentes images OCT en fonction de leur véritable diagnostic. La mise en œuvre de la technique d'apprentissage par transfert a également joué un rôle important dans l'amélioration des performances du modèle proposé et dans la réduction du temps d'apprentissage du programme malgré la petite taille de la base de données utilisée. Cette stratégie a également permis de réaliser une étude comparative entre deux modèles, VGG16 et InceptionV3, qui ont été préalablement pré-entraînés sur une très grande base de données appelée "ImageNet". Cette application est renforcée par le cryptage des images à l'aide des systèmes chaotiques. Même si le système de catégorisation des maladies réiniennes a donné d'excellents résultats, de nombreuses améliorations peuvent encore être apportées, telles que :

- Ajout de nouveaux troubles réiniens
- Utilisation d'un matériel performant pour améliorer la précision
- Amélioration de l'application et la déployer sur le cloud.
- Ajouter ses caractéristiques au logiciel de la machine OCT.

Bibliographie

- [1] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. (2012).
- [2] Géron, A. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow.
- [3] Karen Simonyan, Andrew Zisserman. VERY DEEP CONVOLUTIONAL NETWORKS.(2015).
- [4] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. (2014).
- [5] Min Lin, Qiang Chen, Shuicheng Yan. Network In Network. (2014).
- [6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going deeper with convolutions. (2014).
- [7] Depeng Wang, Liejun Wang.On OCT Image Classification via Deep Learning. Published in : IEEE Photonics Journal (Volume : 11, Issue : 5, October 2019)
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. (2015).
- [9] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. MobileNetV2 : Inverted Residuals and Linear Bottlenecks.(2019).
- [10] Srikanth Tammina. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images(2019)
- [11] N.K. Pareek a,b, Vinod Patidar a, K.K. Sud a,b, Image encryption using chaotic logistic map(2006)
- [12] JING YU, YUAN LI, XINWEN XIE, NANRUN ZHOU³, ZHIHONG ZHOU. Image encryption algorithm by using the logistic map and discrete fractional angular transform(2017)
- [13] Flament J., Storck D. Oeil et pathologie générale Masson(1997)
- [14] Django documentation version 4.0, [[https ://docs.djangoproject.com/en/4.0/](https://docs.djangoproject.com/en/4.0/)]. (2022).
- [15] SQLite documentation version 2.2.0 , [[https ://www.sqlite.org/docs.html](https://www.sqlite.org/docs.html)]. (2022).
- [16] Cyril Goutte, Eric Gaussier. A Probabilistic Interpretation of Precision,Recall and F-score, with Implication forEvaluation. (2005).