REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



Université Yahia Fares De Médéa

Faculté des Sciences

Département de Mathématique et Informatique

Année Universitaire : 2019/2020

MÉMOIRE

Présenté pour obtenir le diplôme de

MASTER

Spécialité : Ingénierie des Systèmes et Technologie du Web

Thème

Conception et Réalisation d'un Modèle Deep Learning pour la Classification des Images Spams

Promoteur :

Mr. HADJIR ISMAIL

Préparé par :

BOUTELDJA MOHAMED AMINE

ZIOUCHE ACHOUAK

Remerciement

Au nom d'Allah, le Tout-Puissant, le Très Miséricordieux, le Plus Miséricordieux, je remercie Allah, qui m'a toujours béni et m'a fourni un personnel formidable. Je remercie tous ceux qui ont rendu ce travail possible. Monsieur HADJIR Ismail, mon superviseur, à qui j'ai beaucoup appris et sans son soutien, je ne pourrais pas écrire ce mémoire

Tous mes professeurs de l'université Dr. Yahia Fares de Médéa.

Achouak

Je tiens à exprimer mes vifs remerciements à Mr HADJIR ISMAIL

A tous mes professeurs de l'université Dr. Yahia Fares de Médéa

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

Amine

Dédicace

À Mon père, ma mère et toute ma famille qui continuaient à m'encourager et à croire en moi tout le temps.

A Tous mes amis, à ma chère Hiba DJENIDI qui n'a jamais cessé de m'encourager.

A mon ami Oussama qui était toujours là pour nous aider.

Achouak

A mon père, pour m'avoir insufflé l'abnégation

A ma chère mère, pour sa précieuse attention à mon égard ;
A tous ceux qui croient aux vertus du sacrifice, de l'effort et du travail.

Amine

Table des matières

Tables des matières

Liste des figures

Liste des tableaux

Liste des acronymes

Résumé

Introduction Générale.....1

Chapitre I. Techniques de traitement des images spams

I.1 Introduction..... 4

I.2 Image Spam..... 4

I.2.1 Définition.....5

I.2.2 Types des images spam.....5

I.2.3 Techniques de filtrage de spam.....5

I.3 Architecture de filtrage des spams..... 6

I.4 Processus de filtrage des courriers électroniques indésirables.....6

I.5 Mesures d'évaluation du rendement.....8

I.6 Matériaux et méthodes.....12

I.6.1 Technique de regroupement.....12

I.6.2 Classificateur Naïve Bayes.....13

I.6.3 Les Réseaux de Neurones.....15

I.6.4 Classificateurs d'ensembles approximatifs (Rough Set).....16

I.6.5 Classificateurs SVM.....17

I.6.6 Algorithmes d'apprentissage profond.....17

I.7 Caractéristiques de l'image.....18

I.8 Extraction de caractéristiques.....21

I.9 Conclusion.....23

Chapitre II. Notion utiles sur l'apprentissage profond

II.1 Introduction.....26

II.2 L'apprentissage profond.....26

II.3 Reseaux de Neuronnes.....26

II.4 Reseaux de Neuronnes Convolutifs……..……30
II.5 Apprentissage du réseau de neurones convolutif……..…..39
II.6 Reseaux Residuel……....43
II.7 L'Apprentissage par transfert ……...45
II.7.1 Définition……..45
II.7.2 Transfert d'apprentissage avec des données d'image……....46
II.7.3 Avantages de l'apprentissage par transfert……...47
II.7.4 Groupe de géométrie visuel……..48
II.7.5 AlexNet……....48
II.8 Conclusion……....48
Chapitre III. L'approche proposée
III.1 Introduction…….……50
III.2 Etapes de prétraitement…….….50
III.3 Réseaux de neurones convolutifs…….…....50
III.3.1 L'architecture CNN utilisée……...……..51
III.3 Réseau Résiduel…….....….51
III.4.1 L'architecture ResNet utilisée ……..…51
III.5 L'apprentissage par transfert…… ……..52
III.5.1 Groupe de géométrie visuelle…….. ….….. ..52
III.5.2 AlexNet……. … 53
III.5.3 Réseau résiduel 18……..…… 54
III.6 Outils et environnement…….... ..55
III.7 Conclusion ……...… …..57
Chapitre IV. Implémentation et évaluation
IV.1 Introduction……...……59
IV.2 Ensemble de données……...…… 59
IV.3 Rappel et précision……...……..59
IV.4 Mise en oeuvre et évaluation……...…….…..60
IV.5 Discussion ……...……..67
IV.6 Conclusion ……...……..70
Conclusion générale……...……71
Références……...……72

Liste des figures

Figure I.1 Échantillons pour (i) finance, (ii) produits, Internet et (iv) les images spams de loisirs.…… ……..**.....4**

Figure I.2 Architecture de filtrage anti-spam du serveur de messagerie…….**7**

Figure I.3 Canaux RGB d'histogramme couleur. ……..**20**

Figure I.4 Canaux HSV de l'histogramme HSV…….**21**

Figure I.5 Les bords des images spam et ham…… …….. **22**

Figure II.6 Les fonctions d'activation…….. **28**

Figure II.7 Réseau de neurones multicouche à action directe……**29**

Figure II.8 Fonction d'activation unipolaire sigmoïde……...**30**

Figure II.9 Réseau de Neurone Convolutif……**31**

Figure II.10 Flux graphique des couches montrant la connexion entre les couches…..**31**

Figure II.11 Flux graphique des couches montrant le partage des poids……**...32**

Figure II.12 Architecture des couches du réseau de neurones convolutionnels…….**32**

Figure II.13 Fonctionnement de la couche convolutionnelle. ……...**35**

Figure II.14 Opération de corrélation croisée……**36**

Figure II.15 Les opérations max pooling et average pooling……**37**

Figure II.16 Aplatir une matrice en vecteur……...**38**

Figure II.17 Passage des gradients de max pooling. ……**41**

Figure II.18 Passage des gradients d'average pooling……...**42**

Figure II.19 Les poids sont partagés dans de nombreux neurones d'entrée pendant l'opération de convolution ……**42**

Figure II.20 Mappage d'identité dans ResNet……..……..**44**

Figure II.21 Bloc d'identité dans ResNet……**.....45**

Figure II.22 Bloc convolutif dans ResNet…… ….**45**

Figure II.23 Trois façons dont le transfert pourrait améliorer l'apprentissage…….**.....47**

Figure III.24 Architecture du modèle CNN…… …..**50**

Figure III.25 Architecture ResNet utilisée……**....51**

Figure III.26 Architecture du modèle VGG-11 utilisée……**52**

Figure III.27 Architecture du modèle AlexNet –Couche 1- ……**53**

Figure III.28 Architecture du modèle AlexNet –Couche 2-……..**53**

Figure III.29 Architecture du modèle AlexNet –Couche 6-……..**54**

Figure III.30 Architecture du modèle ResNet-18……..**54**

Figure IV.31 Graphe de la précision/Perte de CNN……...**61**

Figure IV.32 Graphe de la précision/perte de ResNet…….... **63**

Figure IV.33 Graphe de la précision/perte de ResNet18……**64**

Figure IV.34 Graphe de précision/perte d'AlexNet……...**65**

Figure IV.35 Graphe de précision/perte de VGG-11……**66**

Liste des Tableaux

Tableau I.1 Ensemble de caractéristiques……...23

Tableau II.2 Notation d'architecture de CNN……..…….34

Tableau IV.3 Comparaison entre le résultat des modèles…….67

Tableau IV.4 Performances du CNN par rapport aux autres filtres anti-spam d'image. A : Accuracy, P : précision, R : rappel, F : F-mesure……...68

Liste des acronymes

AI	Artificial Intelligence
CNN	Convolutional Neural Network
DL	Deep Learning
FCL	Fully Connected Layer
ILSVRC	Image Net Large Scale Visual Recognition Challenge
LSTM	Long-Short Term Memory
ML	Machine Learning
OCR	Optical Character Recognition
RELU	Rectifier Linear Activation Function
ResNet	Residual Network
VGG	Visual Geometry Groupe

Résumé

Le spam est un contenu en vrac indésirable, tandis que le spam image est un contenu indésirable intégré à l'intérieur des images. Le spam image constitue une menace pour les systèmes de communication par courrier électronique. De nos jours, beaucoup de contenus non sollicités circulent sur Internet.

Bien que de nombreuses techniques d'apprentissage automatique réussissent à détecter les spams textuels, ces techniques ne sont pas applicables sur les images spams qui peuvent facilement échapper à ces systèmes de détection de spams textuels. Dans ce projet, nous explorons et évaluons les techniques d'apprentissage profond qui détectent les spams images. Tout d'abord, nous étudions les réseaux de neurones profonds, que nous formons sur diverses fonctionnalités de l'image. Nous explorons leur robustesse sur un ensemble de données téléchargé de Image Spam Hunter. Enfin, nous concevons des architectures de réseaux de neurones à convolution, et nous fournissons des résultats expérimentaux pour celles-ci parallèlement au modèle d'apprentissage par transfert existant pour détecter les images spams.

Mots clés : Courrier électronique, Classification des images, images spams, apprentissage profond. CNN, Transfert Learning.

Abstract

Spam is unwanted bulk content and image spam is unwanted content embedded within images. Image spam creates a threat to email communication systems. Nowadays, a lot of unsolicited content circlates on the Internet.

While many machine learning techniques can successfully detect text spam, these techniques are not applicable on spam images which can easily escape these text spam detection systems. In this project, we explore and evaluate deep learning techniques that detect image spam. First, we study deep neural networks, which we train on various functionalities of the image. We explore their robustness on a dataset downloaded from Image Spam Hunter. Finally, we design convolutional neural network architectures and provide experimental results for them in parallel with the existing transfer learning model to detect spam images.

Keywords : E-mail, Image classification, spam images, Deep Learning. CNN, Transfer Learning.

ملخص

البريد الإلكتروني المزعج فيه هو محتوى مجمع غير مرغوب فيه، أما البريد الإلكتروني المزعج للصور فهو محتوى غير مرغوب فيه مضمن داخل الصور. يؤدي البريد غير المرغوب فيه للصور إلى تهديد أنظمة اتصالات البريد الإلكتروني. في أيامنا هذه، يتم تداول الكثير من المحتويات غير المرغوب فيها على الإنترنت على الرغم من نجاح العديد من تقنيات التعلم الالي في الكشف عن الرسائل النصية غير المرغوب فيها إلا أن هذه التقنيات لا تسري على الصور غير المرغوب فيها، والتي يمكنها بسهولة الإفلات من أنظمة الكشف عن الرسائل النصية غير المرغوب فيها. في هذا المشروع سوف نستكشف ونقيم أساليب التعلم المتعمقة التي تكشف عن الرسائل غير المرغوب فيها للصور. أولا، ندرس شبكات الخلايا العصبية العميقة التي ندربها على العديد من خصائص الصورة. ونستكشف بعدها قوة مجموعة البيانات التي تم تحميلها من (Image Spam Hunter). وأخيرا، نقوم بتصميم شبكات عصبية معمقة ونوفر لها نتائج تجريبية بالتوازي مع نموذج التعلم القائم للنقل (Transfer Learning)، لاكتشاف صور البريد الإلكتروني غير المرغوب فيه.

الكلمات الرئيسية : البريد الإلكتروني، تصنيف الصور، الصور المزعجة، التعلم العميق، نقل التعلم، شبكة الخلايا العصبية التلافيفية.

Introduction Générale

Le courrier électronique est aujourd'hui l'une des formes de communication numérique les plus courantes [1], selon [2] le nombre d'utilisateurs d'email est 3.9 milliards en 2019, le nombre de compte é-mail est 4.4 milliards quant au nombre de mail envoyé chaque jour est 293 milliards (hors spam).

Le spam peut être défini comme un courrier électronique en masse non sollicité. L'utilisation répandue du courrier électronique en fait une cible attrayante pour les spammeurs. Le courrier indésirable qui peut inclure des publicités, des logiciels malveillants, du contenu pour adultes, etc., représente une menace importante pour l'utilité du courrier électronique en tant que moyen de communication. [1]

Les spammeurs injectent des informations indésirables dans un e-mail et les combinent pour réduire le taux de reconnaissance des systèmes de filtrage de spam, mettant ainsi en œuvre l'objectif d'éluder la détection. [3]

À ses débuts, le spam a été vu sous forme d'e-mails texte. De nombreux classificateurs puissants ont été développés pour filtrer les courriers indésirables, en fonction du contenu, du sujet, de l'en-tête, etc. Des algorithmes d'apprentissage automatique, y compris les K Nearest Neighbors (KNN), Support Vector Machines (SVM), Neural Networks (NN), etc. ont été utilisés pour la détection du spam.

Avec de solides classificateurs développés basés sur le texte, les spammeurs ont réagi en développant de nouvelles techniques, le spam d'image et le spam de rétrodiffusion pour échapper à la détection basée sur le texte. Le spam image est un spam envoyé par e-mail sous forme d'images. Le texte de spam intégré dans une image peut être une méthode efficace pour échapper à la détection basée sur le texte. Selon un récent rapport de Symantec, le spam représente désormais 90,4% de tous les e-mails. [1]

Le spam images était vu sous forme de texte simple converti en images. Pour détecter ce type de spam d'image, la reconnaissance optique de caractères (OCR) a été utilisée. La reconnaissance optique de caractères extrait le texte à l'intérieur de ces images, puis il est soumis à des techniques de détection basées sur le texte. En réaction à la détection basée sur l'OCR, les spammeurs ont introduit des techniques d'obscurcissement dans les images spam. L'obscurcissement (obfuscation) empêche l'OCR de lire le texte intégré dans les images.

Ce document est organisé comme suit : Le chapitre 1 donne un aperçu du spam images, les techniques et les méthodes de détection des spams et les propriétés des images spams, le deuxième chapitre décrit et explique le Deep Learning et ses différentes modèles de détection de spam et la chapitre 3 présente notre approche basée sur cette technique. En fin Le chapitre 4 donne les détails d'implémentation de nos techniques. Les résultats expérimentaux et l'analyse des performances, la discussion des résultats et une conclusion.

Problématique

La classification des images spams est un nouveau besoin dans le domaine de la sécurité. Vu l'incapacité des filtres anti spams textuels de détecter les images spam, et vu les limites des filtres anti spams images développés par des anciennes méthodes (Machine Learning et autres), par rapport aux nouvelles méthodes basés sur le Deep Learning, nous avons pensé à élaborer un filtre anti spam image basé sur un des récents Deep Learning modeles existants, pour faire face à ce genre de mail indésirables basé sur les images.

Objectif de la recherche

Les approches précédentes pour détecter le spam image consistent à extraire diverses propriétés d'image et à classer les images comme spam ou non-spam, en utilisant des techniques d'apprentissage automatique. Ces travaux antérieurs avaient une insuffisance de classification des images spams et obtiennent une faible précision. Dans cette recherche, nous développons et analysons une modèle Deep Learning plus récente et plus robuste afin d'obtenir de meilleurs résultats et précision des filtres que ceux qui sont obtenus dans les articles précédents jusqu'à présent.

Chapitre I

« Techniques de traitement des images spams»

I.1. Introduction

Dans ce chapitre, nous allons présenter les images spam, leurs types, leurs techniques de filtrage et leurs caractéristiques, puis nous expliquerons l'architecture et le processus de filtrage sur ces images, par la suite nous mentionnant les différentes techniques et les méthodes de classification des images (SVM, Naïve Bayes, les réseaux de neurones et l'apprentissage profond).

I.2. Image Spam

Cette section fournit une introduction au phénomène des images spams. Les spammeurs ont profité des images pour lutter contre les techniques de filtrage textuel. Le spam d'image peut facilement échapper aux filtres anti-spam traditionnels et devient de plus en plus difficile à détecter. Les spammeurs utilisent l'image dans un e-mail de spam pour intégrer leurs messages cibles. Figure.1 montre un échantillon des images spams dans différents domaines. [4]

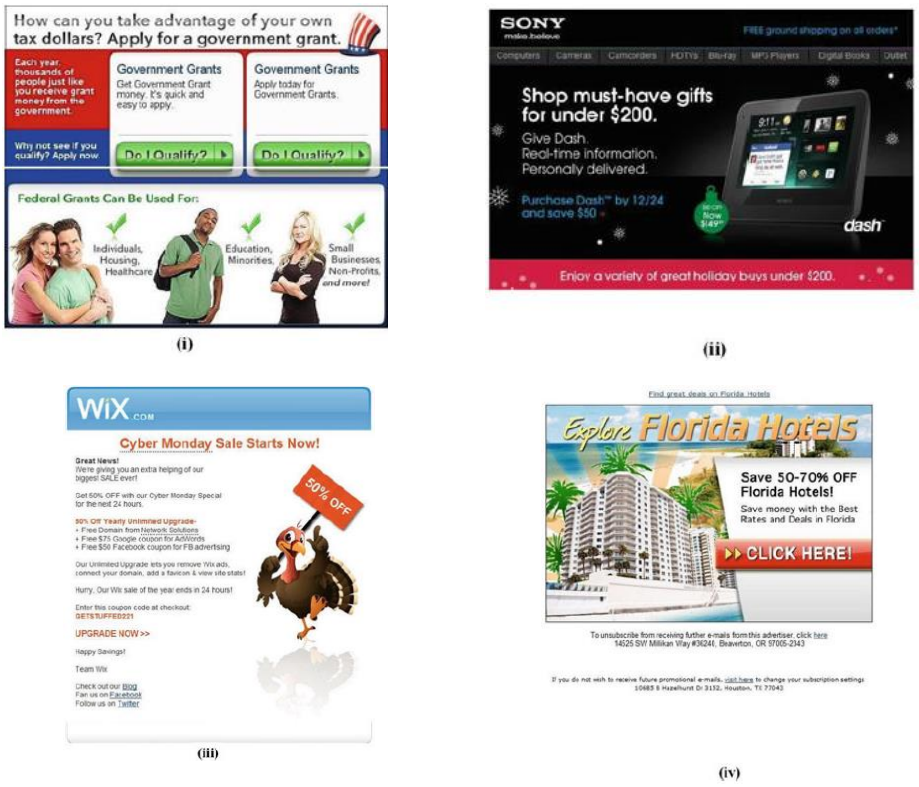


Figure I.1 Échantillons pour (i) finance, (ii) produits, Internet et (iv) les images spams de loisirs.

I.2.1. Définition

Il existe différentes définitions de ce que sont les e-mails des images spam et en quoi ils diffèrent des autres e-mails. L'image spam a une signification différente pour différentes personnes et il n'y a pas de définition uniforme en théorie jusqu'à présent., l'image spam est une sorte de spam par courrier électronique où le texte du message spam est présenté sous forme d'image dans un fichier image. [4]

I.2.2. Types des images spam

Image Spam est une technique de spam par courrier électronique développée pour échapper aux techniques de détection basées sur le contenu. Les techniques des images spam ont évolué aujourd'hui. On peut les classer en 3 générations.

- Image spam de première génération : le début des images spams a commencé avec du texte simple intégré dans les images. Il s'agissait d'un effort réussi pour échapper aux schémas de détection basés sur le contenu. La combinaison de la technique OCR avec le filtrage basé sur le contenu a servi de bon classificateur pour cette classe de spam d'image.
- Image spam de deuxième génération : dans la deuxième génération des images spam, des images d'arrière-plan et du bruit ont été introduits dans l'image. C'était une tentative de rendre le filtrage OCR difficile. Étant donné que l'OCR recherche du texte à l'intérieur des images, l'ajout de bruit de fond a rendu difficile la reconnaissance par OCR du texte.
- Image spam de troisième génération : cette classe des images spam présente des images pertinentes avec le texte. Par exemple, ajouter une image d'une montre avec le texte de l'annonce. Dans ce scénario, même si l'OCR détecte le texte, la présence d'une veille réelle dans l'image perturberait le schéma de détection.

I.2.3. Techniques de filtrage de spam

Les techniques de détection de spam peuvent être divisées en deux catégories en fonction du contenu de l'e-mail. Les classificateurs de chaînes sont construits à l'aide de mots clés extraits des courriers indésirables, des en-têtes, de la charge utile, etc.

- **Filtres basés sur le contenu**

  Des schémas de détection basés sur le contenu peuvent être utilisés pour filtrer les e-mails de spam basés sur du texte. Ils s'appuient sur le contenu / texte contenu dans les spams.

- **Filtres non basés sur le contenu**

  Des schémas de détection non basés sur le contenu sont utilisés pour détecter les spams de formulaires plus avancés comme le spam d'image. Ces schémas de détection reposent fortement sur d'autres propriétés des e-mails comme les propriétés de l'image. [1]

I.3. Architecture de filtrage des spams

Le filtrage anti-spam vise à réduire au minimum le volume des e-mails non sollicités. Le filtrage des e-mails est le traitement des e-mails pour les réorganiser conformément à certaines normes définies. Les filtres de messagerie sont généralement utilisés pour gérer les courriers entrants, filtrer les courriers indésirables, détecter et éliminer les courriers contenant des codes malveillants tels que virus, cheval de Troie ou malware. Le fonctionnement du courrier électronique est influencé par certains protocoles de base qui incluent le SMTP. Certains des agents d'utilisateur de messagerie (MUA) largement utilisés sont Mutt, Elm, Eudora, Microsoft Outlook, Pine, Mozilla Thunderbird, IBM notes, Kmail et Balsa. Ce sont des clients de messagerie qui aident l'utilisateur à lire et à composer des e-mails. Les filtres anti-spam peuvent être déployés à des endroits stratégiques des clients et des serveurs.

Les filtres anti-spam sont déployés par de nombreux fournisseurs d'accès Internet (FAI) à chaque couche du réseau, devant le serveur de messagerie ou au relais de messagerie où il y a présence d'un pare-feu. Le pare-feu est un système de sécurité réseau qui surveille et gère le trafic réseau entrant et sortant en fonction de règles de sécurité prédéterminées. Le serveur de messagerie sert de solution anti-spam et antivirus intégrée fournissant une mesure de sécurité complète pour le courrier électronique au périmètre du réseau. Les filtres peuvent être implémentés dans les clients, où ils peuvent être montés en tant que modules complémentaires dans les ordinateurs pour servir d'intermédiaire entre certains périphériques d'extrémité. Les filtres empêchent les e-mails non sollicités ou suspects qui menacent la sécurité du réseau d'accéder au système informatique. De plus, au niveau du courrier électronique, l'utilisateur peut disposer d'un filtre anti-spam personnalisé qui bloquera les courriers indésirables conformément à certaines conditions définies. [5]

I.4. Processus de filtrage des courriers électroniques indésirables

Un message électronique est composé de deux composants principaux qui sont l'en-tête et le corps. L'en-tête est la zone qui contient des informations générales sur le contenu de l'e-mail. Il comprend le sujet, l'expéditeur et le destinataire. Le corps est au cœur de l'e-mail.

Il peut inclure des informations qui n'ont pas de données prédéfinies. Les exemples incluent la page Web, l'audio, la vidéo, les données analogiques, les images, les fichiers et le balisage HTML. L'en-tête de l'e-mail comprend des champs tels que l'adresse de l'expéditeur, l'adresse du destinataire ou l'horodatage qui indiquent quand le message a été envoyé par des serveurs intermédiaires aux agents de transport de messages (MTA) qui fonctionnent comme un bureau pour organiser les e-mails. La ligne d'en-tête commence généralement par un «De» et elle subit quelques modifications chaque fois qu'elle passe d'un serveur à un autre via un serveur intermédiaire. Les en-têtes permettent à l'utilisateur de visualiser l'itinéraire emprunté par le courrier électronique et le temps pris par chaque serveur pour traiter le courrier. Les informations disponibles doivent passer par un certain traitement avant que le classificateur puisse les utiliser pour le filtrage. La figure 2 ci-dessous illustre une architecture de serveur de messagerie et comment le filtrage anti-spam est effectué. Les étapes nécessaires qui doivent être observées dans l'extraction des données d'un message électronique peuvent être classées comme suit : [5]

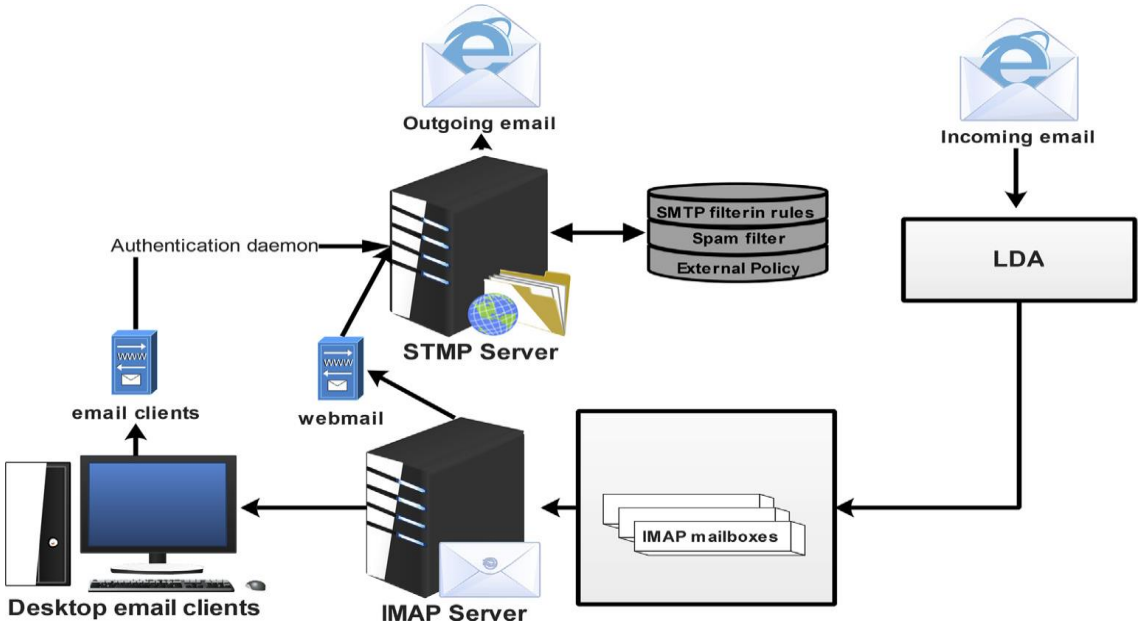


Figure I.2 Architecture de filtrage anti-spam du serveur de messagerie.

- **Prétraitement** : il s'agit de la première étape qui est exécutée chaque fois qu'un courrier entrant est reçu. Cette étape consiste en une tokenisation.

- **Tokenisation** : La tokenisation est le processus de démarcation et éventuellement de classification des chaînes de caractères d'entrées d'un e-mail. Il transforme également un message en ses parties significatives. Il prend l'e-mail et le divise en une séquence de symboles représentatifs appelés jetons. Subramaniam, Jalab et Taqa ont souligné que ces symboles représentatifs sont extraits du corps de l'e-mail, de l'en-tête et du sujet. Guzella et Caminhas ont affirmé que le processus de remplacement des informations par des symboles d'identification distinctifs extraira toutes les caractéristiques et tous les mots du courrier électronique à l'exclusion de la prise en compte de la signification.
- **Sélection des fonctionnalités** : la suite de l'étape de prétraitement est la phase de sélection des fonctionnalités. Sélection d'entités une sorte de réduction de la mesure de la couverture spatiale qui illustre efficacement des fragments fascinants de message électronique en tant que vecteur d'entités compressé. La technique est avantageuse lorsque la taille du message est grande et qu'une représentation condensée des caractéristiques est nécessaire pour rendre la tâche de correspondance de texte ou d'image rapide. Fraude à l'avance sur les frais, y compris les escroqueries en matière d'héritage, de loterie, de visa et de dédouanement, les escroqueries militaires, les publicités pour divers sites externes, gagner de l'argent grâce à Emplois «travail à domicile», achats en ligne, demandes de plaidoyer et de cadeaux, propositions commerciales et autres. Certaines des fonctionnalités les plus importantes pour le filtrage du courrier indésirable comprennent : le corps et l'objet du message, le volume du message, le nombre d'occurrences de mots, les modèles circadiens du message (les messages indésirables présentent généralement de nombreuses différences sémantiques), l'âge, le sexe et le pays du destinataire, a répondu le destinataire (indique si le destinataire a répondu au message), Contenu adulte et Sac de mots du contenu du message. Les fonctionnalités du compte d'expéditeur utilisées pour le filtrage du courrier indésirable comprennent : le pays de l'expéditeur (la répartition des pays indiquée par les utilisateurs sur leur profil et révélée par leur adresse IP), l'adresse IP de l'expéditeur, l'adresse électronique de l'expéditeur, l'âge de l'expéditeur et du destinataire, la réputation de l'expéditeur. Les caractéristiques les moins importantes sont : la distance géographique entre l'expéditeur et le destinataire, la date de naissance de l'expéditeur, le nom d'utilisateur et le mot de passe de l'expéditeur, la durée de vie du compte, le sexe de l'expéditeur

et l'âge du destinataire. La reconnaissance des courriers indésirables avec un nombre minimum de fonctionnalités est importante compte tenu de la complexité et du temps de calcul. La sélection des fonctionnalités implique des processus tels que la racine, la suppression du bruit et les étapes de suppression des mots vides. [5]

I.5. Mesures d'évaluation du rendement

Les filtres anti-spam sont généralement évalués sur de grandes bases de données contenant des messages Ham et des messages Spam qui sont accessibles au public. Un exemple des mesures de performance utilisées est la précision de classification (*Acc*). C'est le nombre comparatif de messages correctement classés, le pourcentage de messages correctement classés est utilisé comme mesure supplémentaire pour évaluer les performances du filtre. Il a cependant été souligné que l'utilisation de la précision comme seul indice de performance n'est pas suffisante. D'autres mesures de performance telles que le rappel, la précision et les mesures dérivées utilisées dans le domaine de la recherche d'informations doivent être prises en compte, de même que les faux positifs et les faux négatifs utilisés dans la théorie de la décision. Ceci est très important en raison des coûts liés à une mauvaise classification. Lorsqu'un message indésirable est classé à tort comme Ham, cela pose un problème quelque peu insignifiant, car la seule chose que l'utilisateur doit faire est de supprimer ce message. En revanche, lorsqu'un message non spam est incorrectement étiqueté comme spam, cela est désagréable, car il indique la possibilité de perdre des informations précieuses en raison de l'erreur de classification du filtre. Ceci est très impératif, en particulier dans les paramètres où les messages indésirables sont automatiquement supprimés. Par conséquent, il est inadéquat d'évaluer les performances de tout algorithme d'apprentissage automatique utilisé dans le filtre anti-spam en utilisant exclusivement la précision de la classification. De plus, dans un cadre déséquilibré ou biaisé où le nombre de messages de spam utilisés pour tester les performances du filtre est très supérieur à celui des messages Ham, le classificateur peut enregistrer une très grande précision en se concentrant sur la détection des messages spam uniquement. Dans un environnement réel où il n'y a rien de tel qu'une probabilité nulle de catégoriser à tort un message ham, il est nécessaire qu'un compromis soit trouvé entre les deux types d'erreurs, en fonction de la prédisposition de l'utilisateur et des indicateurs de performance utilisés. Les formules de calcul de la précision de la classification et de l'erreur de classification sont décrites dans les équations. (1) et (2) ci-dessous :

Étant donné que

|S→H| = Le nombre de messages de spam classés comme non-spam

|H→S| = Le nombre de messages non spam classés comme spam

Supposant

NH = Nombre de messages non spam à classer

NS = Nombre de spams à classer

Précision de classification $(Acc) = \frac{|H \to H| + |S \to S|}{N_H + N_S}$ (**1**)

Erreur de classification $(Err) = 1 - Acc = \frac{|H \to S| + |S \to H|}{N_H + N_S}$ (**2**)

Selon [6], la précision et l'erreur de classification prennent mutuellement en compte les occurrences faux Positifs $|H \to S|$ et faux Negatifs $|S \to H|$ pour supporter un coût égal. Il est nécessaire de souligner que des coûts d'erreur disproportionnés sont impliqués dans le filtrage du spam. La classification erronée d'un message Ham comme spam (également connu sous le nom d'événement faux positif) est une erreur coûteuse par rapport au message de spam qui échappe au filtre. Un tel incident est appelé faux négatif. Lorsqu'un e-mail légitime est correctement classé comme Ham, il est appelé un véritable événement positif $|H \to H|$. Cependant, lorsqu'un e-mail de spam est correctement classé comme spam, un véritable événement négatif $|S \to S|$ s'est produit. Sur la base des explications ci-dessus, le taux de faux positifs (*FPR*) peut être défini comme le ratio de ham ou d'e-mails valides classés comme spam. Il est indiqué à l'aide de la formule de l'équation. (3) ci-dessous :

$$FPR = \frac{Nb\ de\ Faux\ Positifs}{Nb\ de\ Faux\ Positifs + Nb\ de\ Vrais\ Négatifs} \quad (\mathbf{3})$$

En outre, autoriser les spams qui ont été infectés par des malwares, des spywares, des adwares, des chevaux de Troie, des botnet, des virus, des vers ou des appâts de phishing tels que les messages prétendant provenir de sites Web sociaux, de sites de rencontres, de sites d'enchères, de banques, de processeurs de paiement en ligne ou Les administrateurs informatiques sont généralement utilisés pour piéger les victimes. Cela peut entraîner des pertes monumentales pour les utilisateurs. Le ratio de messages indésirables qui ont été classés à tort comme du jambon est appelé taux de faux négatifs (FNR). Il s'agit d'une mesure de plus appropriée pour évaluer les performances d'un filtre. La formule de calcul du FNR est en équation. (4) ci-dessous :

$$FNR = \frac{Nb\ de\ Faux\ Négatifs}{Nb\ de\ Vrais\ Positives + Nb\ de\ Faux\ Négatives} \quad (4)$$

Les filtres anti-spam avec un *FPR* et un *FNR* considérablement réduits auraient de meilleures performances. Ces caractéristiques standard (*FNR* et *FPR*) représentent l'efficacité des filtres qui visent directement la limite de décision de classification sans générer d'estimation de probabilité.

Les deux métriques importées du domaine de la «rappel» et de la «précision» de la recherche d'informations sont respectivement utilisées pour obtenir l'efficacité et les caractéristiques des filtres anti-spam. [5]

Étant donné que

|S→H| = Le nombre de messages de spam classés comme non-spam

|H→S| = Le nombre de messages non spam classés comme spam

Respectivement, et

Où |H→H| et |S→S|

L'équation. (5) ci-dessous représente le rappel du spam (Rs) et la précision du spam (Ps) :

$$R_S = \frac{|S \to S|}{|S \to S| + |S \to H|} \quad \text{Et} \quad P_S = \frac{|S \to S|}{|S \to S| + |H \to S|} \quad (5)$$

Le rappel (*Rs*) également connu sous le nom d'efficacité peut être défini comme le nombre comparatif de messages spam que le filtre a réussi à empêcher d'entrer dans la boîte de réception des e-mails. La précision (*Ps*), également décrite comme la valeur ou la fiabilité du filtre, est calculée en divisant le nombre de messages classés par le filtre comme spam, mais ils sont vraiment limités par le nombre total d'e-mails. L'évaluation des performances de différents filtres anti-spam à l'aide de (*Rs*) et (*Ps*) est délicate compte tenu des différentes valeurs impliquées dans les calculs qui ont produit (*Rs*) et (*Ps*). Selon [7], le coût des faux positifs est beaucoup plus grand (λ fois) que celui des faux négatifs, où lambda (λ) est un facteur numérique qui stipule à quel point il est «risqué ou nuisible» de classer à tort un e-mail comme spam. Il indique également à quel point il peut être difficile pour l'utilisateur de se remettre d'une telle performance abyssale et inacceptable du filtre anti-spam. La sensibilité aux coûts doit être considérée comme suggérée. Pour ce faire, chaque message valide doit être égal à α. Clark [8] dans son article a donné la formule de calcul des mesures sensibles aux coûts telles que la précision pondérée (*WAcc*), le taux d'erreur pondéré (*WErr*) et le ratio de coût total (*TCR*). [5]

Le coût total est utilisé pour mesurer la précision des filtres, il a été proposé par [7]. Un TCR plus élevé implique de meilleures performances. Lorsque la valeur de TCR <1, il est préférable de ne pas utiliser le filtre. La comparaison ne peut être établie entre les valeurs de TCR que lorsque tous les TCR évalués ont été calculés en utilisant un λ similaire.
Les formules de calcul de la précision pondérée (*WAcc*), du taux d'erreur pondéré (*WErr*) et du ratio de coût total (*TCR*) sont représentées en équations. (6), (7) et (8) ci-dessous :

$$W_{Acc} = \frac{\lambda \mid H \to H \mid + \mid S \to S \mid}{N_H + N_S} \quad (6)$$

$$W_{Err} = \frac{\lambda \mid H \to S \mid + \mid S \to H \mid}{N_H + N_S} = 1 - W_{Acc} \quad (7)$$

$$TCR = \frac{N_S}{\lambda \mid H \to S \mid + \mid S \to H \mid} \quad (8)$$

Une autre mesure pour mesurer les performances d'un filtre est la mesure F (F1-score ou F-score). Il s'agit d'une mesure de la précision d'un test et est décrit comme la moyenne harmonique pondérée de la précision (*Ps*) et du rappel (*Rs*) du test dans une seule équation. La mesure F utilise un paramètre qui permet de parvenir à un compromis concernant le rappel et la précision. F1 est la mesure F traditionnelle qui est couramment utilisée et elle présente un poids uniforme pour rappel et précision comme indiqué dans les équations. (9).

$$F1 = \frac{2 * Rappel * Précision}{Rappel + Précision} \quad (9)$$

I.6. Matériaux et méthodes

Récemment, la classification du courrier indésirable est normalement gérée par des algorithmes d'apprentissage automatique (ML) destinés à différencier les messages indésirables et non indésirables. Les algorithmes d'apprentissage automatique y parviennent en utilisant une technique automatique et adaptative. Plutôt que de dépendre de règles codées à la main qui sont sensibles aux caractéristiques perpétuellement variables des messages de spam, les méthodes ML ont la capacité d'obtenir des informations à partir d'un ensemble de messages fournis, puis d'utiliser les informations acquises pour classer les nouveaux messages qu'ils viennent de recevoir. Dans cette section, nous passerons en revue certaines des méthodes d'apprentissage automatique les plus populaires qui ont été appliquées à la détection du spam. [5]

I.6.1. Technique de Regroupement

Le clustering consiste à classer un groupe de modèles en classes apparentées. Le clustering est un type d'approche utilisé pour diviser des objets de cas en collections relativement similaires appelées clusters. Les techniques de regroupement ont attiré l'attention de nombreux chercheurs et académiciens récents et ont été appliquées dans différents domaines d'application.

Les algorithmes de clustering qui sont des outils d'apprentissage non supervisés sont utilisés sur les jeux de données de spam par e-mail qui ont généralement de véritables étiquettes. Pourvu qu'il y ait des représentations appropriées, un bon nombre d'algorithmes de clustering ont la capacité de classer les jeux de données spam par e-mail en ham ou en clusters spam. Whissell et Clarke [9] l'ont prouvé dans leurs travaux de recherche sur le regroupement des spams par courrier électronique. Il classe des objets ou des opinions de telle sorte que les objets d'un même groupe se ressemblent davantage que ceux d'un autre groupe. Deux types de méthodes de clustering qui ont été utilisées pour la classification des spams sont le clustering basé sur la densité et les K-voisins les plus proches (*KNN*). KNN est une méthode sans distribution, qui ne repose pas sur des hypothèses selon lesquelles les données sont tirées d'une distribution de probabilité donnée. [5]

Dans le classificateur KNN, le modèle de classification n'est pas construit à partir de données, la classification est plutôt effectuée en faisant correspondre l'instance de test avec K exemples d'apprentissage et la décision est prise de savoir à quel groupe il appartient en fonction de la ressemblance avec K voisins les plus proches. Le KNN est appelé un apprenant paresseux car les points de données d'entrainement ne sont pas utilisés par lui pour effectuer la généralisation. Autrement dit, il n'y a pas de stade d'entraînement évident et s'il existe, il est extrêmement petit. L'implication est que l'algorithme a une phase d'apprentissage modérément rapide. L'absence d'universalité oblige KNN à stocker toutes les données d'entraînement. Pour être précis, l'intégralité des données d'entrainement est requise tout au long de la phase de test, car les décisions sont prises sur la base de l'ensemble complet de données de l'entrainement. La contradiction est tout à fait claire ici qu'il n'y a pas d'étape d'entrainement importante, mais plutôt une étape de test approfondie.

I.6.2. Classificateur Naïve Bayes

La classification bayésienne illustre une technique d'apprentissage supervisé et en même temps une technique statistique de classification. Il agit comme un modèle probabiliste fondamental et saisissons l'ambiguïté sur le modèle de manière éthique en influençant les probabilités des résultats. Il est utilisé pour apporter une solution aux problèmes analytiques

et prédictifs. La classification bayésienne tire son nom de Thomas Bayes (1702–1761), qui a proposé l'algorithme. Il calcule les probabilités exactes de postulation et il est robuste au bruit dans les données d'entrée. Un classificateur Naive Bayes est un classificateur probabiliste simple qui est fondé sur le théorème de Bayes avec des hypothèses solides qui sont de nature indépendante.

Une meilleure expression pour le modèle de probabilité devrait être un modèle caractéristique autonome comme indiqué dans l'équation. (10) :

Théorème de Bayes : Prob *(B* given *A)*= Prob *(A* and *B)*=Prob *(A)* (**10**)

Néanmoins, l'algorithme est efficace et très robuste. Il fonctionne comme les autres algorithmes d'apprentissage supervisé. Il y a eu une recrudescence de l'acceptation de NB en tant qu'algorithme simple et efficace sur le plan informatique avec des performances satisfaisantes pour résoudre des problèmes du monde réel. En raison de ses qualités exceptionnelles, les classificateurs NB ont trouvé une application comme algorithme de classification dans le texte, le courrier indésirable, l'analyse des sentiments, les systèmes de recommandation, les avis de spam et d'autres applications en ligne. Les classificateurs Naive Bayes sont particulièrement utilisés dans la classification de texte (car ils produisent un résultat supérieur dans les problèmes de classes multiples et les règles d'indépendance) et ont un taux de réussite supérieur par rapport à certains autres algorithmes d'apprentissage automatique. En raison de cet avantage évident, il est largement appliqué dans le domaine du filtrage du spam (détecter les courriers indésirables) et de l'analyse des sentiments (dans l'analyse des médias sociaux, pour reconnaître les opinions positives et négatives des clients). C'est une méthode générale pour différencier les e-mails non autorisés, c'est-à-dire le spam, des e-mails légaux, souvent appelés ham. Pratiquement toutes les techniques de filtrage de spam basées sur des statistiques utilisent le classificateur de Naïve Bayes pour regrouper les statistiques de chaque jeton en un score total, et le score est utilisé pour prendre une résolution sur le filtrage. Le jeton T qui dénote le spam (indice de spam) est calculé comme illustré dans l'équation. (11) :

$$S[T] = \frac{C_{Spam}(T)}{C_{Spam}(T)+C_{Ham}(T)} \quad (\mathbf{11})$$

Où :

C*Spam* (T) = Le nombre de messages spam contenant le token T

C*Ham* (T) = Le nombre de messages ham contenant le token T

Il sera nécessaire de fusionner le spam du différent jeton pour calculer le message spam global afin de calculer la probabilité pour un message M avec des jetons {T_1, ..., T_N}. Calculer le produit du jeton spam spécifique et le comparer avec le produit du jeton ham spécifique est un moyen simple de faire des classifications. Il est représenté dans l'équation. (12) ci-dessous :

$$(H[M] = \prod_{i=1}^{N}(1 - S[T_i]) \quad (\mathbf{12})$$

Le message est classé comme spam si le produit de spam total S [M] est supérieur au produit de ham H [M].

I.6.3 Les Réseaux de Neurones

Les réseaux de neurones artificiels sont des groupes d'unités de traitement simples qui sont interconnectés et communiquent entre eux au moyen d'un nombre important de connexions pondérées. Chacune des unités accepte l'entrée des unités voisines et des sources externes et calcule la sortie qui est transmise aux autres voisins. Le support pour affiner les poids des connexions est également disponible. Les réseaux de neurones sont un algorithme puissant pour résoudre tout problème d'apprentissage automatique qui nécessite une classification. En raison de leur ingéniosité, ils évoluent en tant qu'outil majeur dans l'ensemble d'outils du chercheur en apprentissage automatique. Néanmoins, les réseaux de neurones ne sont pas couramment utilisés dans la détection des spams comme on peut le penser. Comme alternative, presque tous les filtres anti-spam de pointe utilisent des classificateurs Naïve Bayes. Cela est principalement dû au travail bien connu de Paul Graham intitulé «Un plan pour le spam». Naïve Bayes est une excellente méthode de classification du spam avec une grande précision (99,99%) et un faible taux de faux positifs. Ce qui améliore sa grande précision est le grand nombre de composants de traitement (neurones) bien interconnectés qui fonctionnent en harmonie pour apporter une solution à certains problèmes. Par exemple, Google a raconté l'augmentation de la précision des filtres anti-spam Gmail de 99,5% à 99,9% après y avoir incorporé des réseaux de neurones. Cela rappelle que les réseaux de neurones pourraient être utiles pour améliorer les performances des filtres anti-spam, en particulier lorsqu'ils sont hybrides avec la classification bayésienne et d'autres techniques. D'un autre côté, de nombreuses recherches doivent être effectuées sur l'application des réseaux de neurones pour la détection du spam, et presque toutes les recherches actuelles prennent la configuration du réseau, l'élan et le taux d'apprentissage à fixer. Davantage d'efforts de recherche doivent être concentrés sur l'efficacité du réseau à travers les ensembles de données plutôt que sur la pertinence des diverses conceptions de réseau pour le travail. Il existe généralement trois types d'unités.

- **Unité d'entrée** : cette unité accepte le signal provenant d'une source extérieure.
- **Unité de sortie** : cette unité transmet des données en dehors du réseau.
- **Unité cachée** : cette unité accepte et transmet des signaux dans le réseau.

Le fonctionnement du système est synchronisé afin qu'un grand nombre d'unités puissent fonctionner en parallèle. L'ANN peut être personnalisé pour accepter un ensemble d'entrées et générer l'ensemble de sorties nécessaire. Ce processus est appelé apprentissage ou entrainement. Il existe deux types d'apprentissage en réseau neuronal :

- **Supervisé** : Ici, le réseau reçoit un ensemble d'entrées et des modèles de sortie correspondants, appelés ensemble de données d'apprentissage, pour former le réseau.
- **Non-supervisé** : dans ce cas, le réseau se forme en produisant des groupes de modèles. Il n'y a aucun ensemble antérieur de données d'entrainement fournies au système.

I.6.4. Classificateurs d'ensembles approximatifs (Rough Set)

La théorie de l'ensemble approximatif a été proposée en 1982 par [11] dans le but de présenter un cadre approprié pour la conversion automatisée des données en connaissances. La technique est axée sur la ventilation de la catégorisation des informations inexactes, ambiguës ou partielles exprimées en termes de données obtenues par l'expérience. La théorie des ensembles approximatifs peut être décrite comme une méthode mathématique récente du flou. L'idée de Rough Set repose sur l'hypothèse qu'une certaine connaissance est associée à chaque objet de l'univers. RS est un outil mathématique qui se concentre sur l'incertitude [10]. Il est conforme à la notion que tout modèle inexact peut être estimé de dessous et de dessus en utilisant une association qui est de nature imperceptible. L'une des principales caractéristiques de la philosophie RS est la nécessité de découvrir la redondance et les dépendances entre les fonctionnalités [12]. La théorie de Rough Set a été appliquée au filtrage du spam car elle fournit des algorithmes efficaces et moins longs à extraire des modèles cachés dans les données. Il a également la capacité d'identifier facilement les relations que d'autres techniques statistiques conventionnelles ont du mal à trouver. De plus, il accepte l'utilisation de données à la fois quantitatives et qualitatives. Il a la capacité d'estimer les ensembles minimaux de données nécessaires pour regrouper les emplois.

Découvrir l'importance des données et créer un groupe de règles de décision à partir de l'ensemble de données donné fait partie de la force des classificateurs RS. Il est important de noter que la théorie des ensembles approximatifs exprime l'imprécision en utilisant une section limite d'un ensemble plutôt que par voie d'appartenance. Le fait d'avoir la section limite d'un

ensemble vide implique que l'ensemble a été clairement défini (exact), sinon l'ensemble est dit grossier (inexact). Pour une section borderline qui contient au moins un élément dans l'ensemble signifie que ce que nous savons sur l'ensemble n'est pas suffisant pour décrire exactement l'ensemble. Selon [13], on peut observer que les techniques Rough Set permettent aux utilisateurs d'évaluer la signification des données. Il permet à l'utilisateur de générer automatiquement les ensembles de règles de décision à partir des données. C'est facile à comprendre. Il offre une interprétation simple des résultats obtenus. Il convient au traitement simultané (parallèle / distribué).

I.6.5. Classificateurs SVM

Les machines à vecteurs de support (SVM) sont des algorithmes d'apprentissage supervisé qui se sont avérés plus performants que certains autres algorithmes d'apprentissage associés [14]. SVM est un groupe d'algorithmes proposé par [15] pour résoudre les problèmes de classification et de régression. SVM a trouvé une application pour fournir une solution aux problèmes de programmation quadratiques qui ont des contraintes d'inégalité et une égalité linéaire en différenciant différents groupes au moyen d'un hyperplan. Il profite pleinement de la frontière [16]. Bien que le SVM ne soit pas aussi rapide que d'autres méthodes de classification, l'algorithme tire sa force de sa grande précision en raison de sa capacité à modéliser des limites multidimensionnelles qui ne sont pas séquentielles ou simples. SVM n'est pas facilement sensible à une situation où un modèle est disproportionnellement complexe, comme avoir de nombreux paramètres comparés au nombre d'observations. Ces qualités font de SVM l'algorithme idéal pour une application dans les domaines de la reconnaissance numérique de l'écriture manuscrite, de la catégorisation de texte, de la reconnaissance des locuteurs, etc.

I.6.6. Algorithmes d'apprentissage profond

L'apprentissage profond est un nouveau domaine émergent qui exploite l'intelligence artificielle et l'apprentissage automatique pour apprendre des fonctionnalités directement à partir des données, en utilisant plusieurs couches de traitement non linéaire. Les modèles d'apprentissage profond peuvent atteindre une très grande précision dans la classification des spams par e-mail. Un réseau neuronal (NN) comprend divers neurones unifiés. Le type d'application utilisé déterminera le nombre de neurones et les connexions entre eux. Selon [18], les méthodes d'apprentissage approfondi peuvent être classées comme suit :

1. **Réseaux profonds non supervisés (également appelés apprentissage génératif).**

Les exemples de réseaux profonds incluent Autoencoder, Sparse Autoencoder (SAE), Stacked Sparse Autoencoder (SSAE), Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs), Deep Boltzmann Machines (DBMs) et generalized débruiting autoencoders. Ils peuvent être utilisés pour produire des échantillons de manière expressive par échantillonnage à partir des réseaux, et ils sont donc appelés modèles génératifs. Dans l'apprentissage en profondeur non supervisé, il n'y a aucune disposition pour les étiquettes pendant l'entrainement et le but général est de désigner une fonction qui est utilisée pour représenter une structure invisible à partir de données non étiquetées.

Les modèles d'apprentissage en profondeur non supervisés remplissent différentes fonctions telles que l'estimation de la densité, le regroupement, l'apprentissage des fonctionnalités, la réduction des dimensions et autres. Cela en fait un meilleur algorithme pour le filtrage du courrier indésirable. Le principal avantage de l'apprentissage non supervisé réside dans les situations où il y a un grand volume de données non étiquetées. L'utilisation de méthodes d'apprentissage en profondeur non supervisées pour des données aussi volumineuses permet un apprentissage facile de fonctionnalités qui sont mieux comparées à des fonctionnalités artisanales.

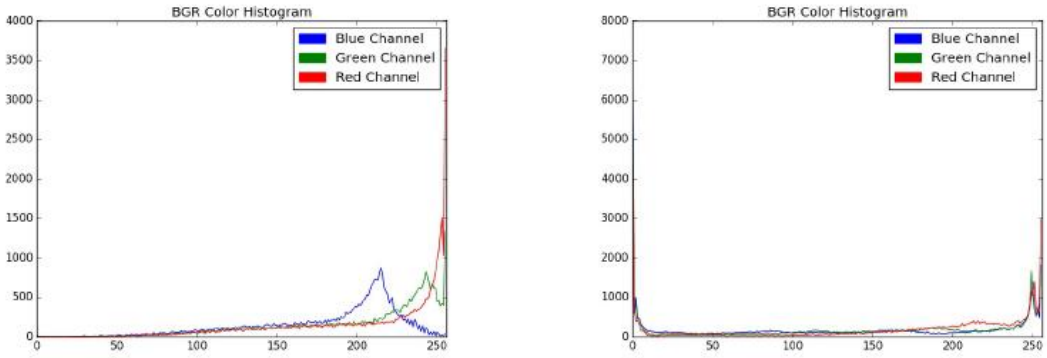
2. **Réseaux profonds supervisés**

Ce sont des réseaux qui sont destinés à fournir une puissance de classification aux fins de la classification des modèles. Ils sont généralement caractérisés par les allocations ultérieures de classes conditionnées par les données visibles. Le réseau somme-produit (SPN) et le réseau neuronal convolutif (CNN) sont des exemples de réseau profond supervisé.

I.7. Caractéristiques des images

Les caractéristiques de l'image sont analogues aux propriétés de l'image. Les images spam sont générées par ordinateur. Ils n'ont pas les propriétés de couleur et la composition de base d'une image normale / ham. Par conséquent, les propriétés d'image des images spam varient beaucoup de naturel / images ham. Par exemple, le changement de luminosité des images naturelles est très élevé par rapport à celui des images spam. Nous avons utilisé des techniques avancées de traitement d'image pour extraire de nombreuses propriétés de ce type à partir d'images. Un total de 41 caractéristiques ont été collectées, dont 21 sont basées sur des recherches antérieures [19]. Le tableau 1 donne un bref aperçu de toutes les fonctionnalités. Ces fonctionnalités peuvent être classées de manière approximative dans 5 domaines.

- Caractéristiques des métadonnées : des propriétés telles que la taille, la hauteur, la largeur, le rapport hauteur / largeur, le taux de compression, le nom de l'image, etc. Une certaine anomalie peut être observée dans les images générées par ordinateur par rapport aux images naturelles.
- Caractéristiques des couleurs : divers histogrammes contiennent des informations sur la constitution de l'image.
  - Histogramme des couleurs : un histogramme des couleurs contient des informations sur l'utilisation des couleurs rouge, vert et bleu. Normalement, dans une image spam, très peu de couleurs sont utilisées par rapport aux images naturelles.
    Nous avons également quantifié les histogrammes RGB et les avons utilisés pour la classification. La figure 3 compare les canaux RGB des images ham et spam.
  - Histogrammes rouge, vert, bleu : La moyenne, la variance, l'inclinaison et le kurtosis de chacun de ces 3 histogrammes sont calculés.



Ham **Spam**

Figure I.3 Canaux RGB d'histogramme couleur. [1]

- Histogramme de teinte, de saturation et de valeur (HSV) : l'histogramme HSV capture les 3 aspects suivants des couleurs d'une image.
  - ❖ Teinte : Elle définit la proximité de la couleur avec le rouge. La teinte est mesurée entre 0 et 1 ; 0 étant rouges.
  - ❖ Saturation : il définit la pureté de la couleur. Des valeurs de saturation plus élevées correspondent à des couleurs plus profondes / plus riches. Le blanc correspond à 0 saturation.
  - ❖ Intensité / Valeur : L'intensité définit la luminosité. Des valeurs d'intensité plus élevées correspondent au blanc.

- Teintes, saturation, histogrammes d'intensité : la moyenne, l'inclinaison, la variance et la kurtose de chacun de ces histogrammes sont capturées. Cela ajoute jusqu'à 12 fonctionnalités extraites des 3 fonctionnalités. La figure 4 compare les canaux HSV d'images ham et de spam

Ham Spam

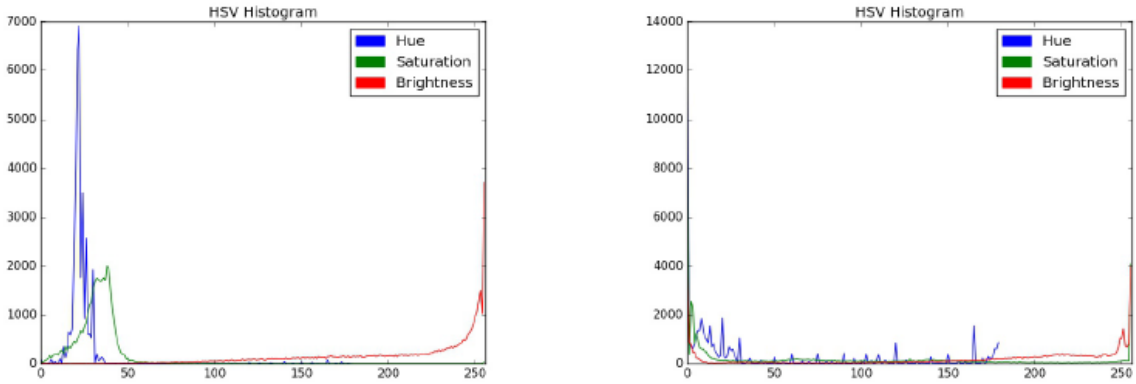


Figure I.4 Canaux HSV de l'histogramme HSV. [1]

- Caractéristiques de texture :
  - Histogramme de modèle binaire local (LBP) : cet histogramme capture des informations sur la texture de l'image. Pour chaque pixel, LBP permet de quantifier la similitude ou la différence de chaque pixel avec ses pixels voisins. Étant donné que les images spam n'ont pas de véritable arrière-plan, LBP capture relativement moins d'informations.
- Caractéristiques de forme :
  - Histogramme des dégradés orientés : cet histogramme est couramment utilisé pour la détection d'objets. Il décrit comment l'intensité des gradients change dans l'image.
  - Bords : Les bords marquent le changement de contraste. Les bords mettent en évidence les limites des caractéristiques d'une image. La figure 5 montre la sortie du filtre de bord sur une image spam et une image ham. Les images spam contiennent en général beaucoup de texte, ce qui entraîne une augmentation du nombre de bords par rapport aux images ham. Une autre observation que nous pouvons faire en regardant les images est que les bords des images spam sont plus

petits que ceux des images ham. Le nombre de bords et la longueur de bord moyenne ont été considérés comme 2 éléments.

- Caractéristiques du bruit :
  - Entropie du bruit : la quantité de bruit dans une image spam est inférieure à une image normale. L'entropie de l'histogramme de bruit est mesurée comme une caractéristique.
  - Rapport signal / bruit (SNR) : le SNR est le rapport de la moyenne et de l'écart type dans l'histogramme de l'image en niveaux de gris.

Image ham



Image spam

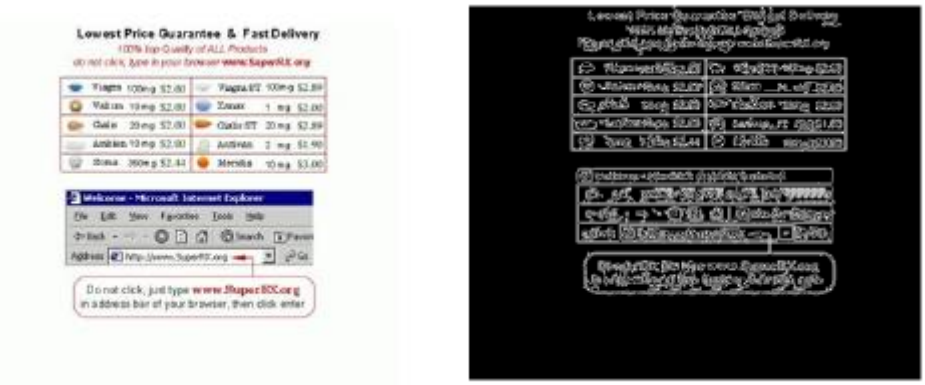


Figure I.5 Les bords des images spam et ham.

I.8. Extraction de caractéristiques

Une fois les propriétés de l'image extraites, ces caractéristiques doivent être quantifiées. Ce processus peut être appelé préparation des données. Les algorithmes d'apprentissage automatique nécessitent une entrée sous forme de vecteurs de caractéristiques, chaque caractéristique étant un nombre. Par conséquent, les caractéristiques de l'image telles que les bords délicats, les histogrammes, etc. doivent être converties en nombres. Diverses techniques statistiques comme l'entropie des histogrammes, la moyenne, la variance, le kurtosis, l'extraction du nombre de bords à partir d'une image de bord rusé, etc. sont utilisées.

Une fois les vecteurs de caractéristiques construits, plusieurs expériences peuvent être menées pour sélectionner un sous-ensemble de caractéristiques afin d'obtenir une plus grande précision. [1]

Tableau I.1 Ensemble de caractéristiques

Domaine de caractéristique	**caractéristique**	**Description**
caractéristiques de métadonnées	la taille	Hauteur de l'image
	Largeur	Largeur de l'image
	Ratio d'aspect	Rapport hauteur / largeur
	Ratio de compression	Quelle est la compression de l'image
	Taille du fichier	Taille sur le disque
	Zone d'image	Zone d'image
Caractéristiques de couleur	entr-color	Entropie de l'histogramme des couleurs
	r-mean	Moyenne de l'histogramme des canaux rouges
	g-mean	Moyenne de l'histogramme des canaux verts
	b-mean	Moyenne de l'histogramme des canaux bleus
	r-skew	Inclinaison de l'histogramme des canaux rouges
	g-skew	Inclinaison de l'histogramme du canal vert
	b-skew	Inclinaison de l'histogramme des canaux bleus
	r-var	Variance de l'histogramme des canaux rouges
	g-var	Variance de l'histogramme des canaux verts
	b-var	Variance de l'histogramme des canaux bleus
	r-kurt	Kurtosis de l'histogramme du canal rouge
	g-kurt	Kurtosis de l'histogramme des canaux verts
	b-kurt	Kurtosis de l'histogramme du canal bleu

	entr-hsv	Entropie de l'histogramme HSV
	h-mean	Moyenne du canal de teinte de l'histogramme HSV
	s-mean	Moyenne du canal de saturation de l'histogramme hsv
	v-mean	Moyenne du canal de luminosité de l'histogramme hsv
	h-var	Variance du canal de teinte de l'histogramme HSV
	s-var	Variance du canal de saturation de l'histogramme hsv
	v-var	Variance du canal de luminosité de l'histogramme hsv
	h-skew	Inclinaison du canal de teinte de l'histogramme HSV
	s-skew	Inclinaison du canal de saturation de l'histogramme hsv
	v-skew	Inclinaison du canal de luminosité de l'histogramme hsv
	h-kurt	Kurtosis du canal de teinte des histogrammes hsv
	s-kurt	Kurtosis du canal de saturation de l'histogramme HSV
	v-kurt	Kurtosis du canal de luminosité de l'histogramme HSV
Caractéristiques de texture	lbp	Entropie de l'histogramme des motifs binaires locaux
Caractéristiques de forme	entr-hog	Entropie de l'histogramme des gradients
	edges	Nombre total de bords dans une image
	avg-edge-length	Longueur moyenne des bords
Caractéristiques du bruit	snr	Rapport signal sur bruit
	entr-noise	Entropie du bruit

I.9. Conclusion

Nous avons présenté dans ce chapitre les différents types d'images, leurs caractéristiques, l'architecture et le processus du filtrage et les méthodes qui permettent d'effectuer cette opération. Dans le chapitre suivant nous allons détailler la méthode suivie dans nos recherches qui est le Deep Learning.

Chapitre II

« Notions utiles sur l’apprentissage profond »

II.1. Introduction

La technologie d'apprentissage profond apprend à représenter le monde. C'est-à-dire «comment la machine représentera la parole ou l'image par exemple», cette phrase de Yann Lecun[1] résume parfaitement ce qu'est l'apprentissage profond. Dans ce chapitre, nous allons présenter le deep Learning et ses diverses applications.

II.2. Apprentissage Profond

II.2.1.Définition

Le Deep Learning est un ensemble de méthodes d'apprentissage qui tentent de modéliser des données avec des architectures complexes combinant différentes transformations non linéaires. Les briques élémentaires de l'apprentissage en profondeur sont les réseaux de neurones, qui sont combinés pour former les réseaux de neurones profonds.

Ces techniques ont permis des progrès significatifs dans les domaines du traitement du son et de l'image, notamment la reconnaissance faciale, la reconnaissance vocale, la vision par ordinateur, le traitement automatique du langage, la classification des textes et des images (par exemple, la reconnaissance du spam). Les applications potentielles sont très nombreuses. [20] Il existe plusieurs types d'architectures pour les réseaux de neurones:

- Les perceptrons multicouches, les plus anciens et les plus simples.
- Les Réseaux de Neurones Convolutionnels (CNN), particulièrement adaptés au traitement d'images.
- Les réseaux de neurones récurrents, utilisés pour les données séquentielles telles que le texte ou séries chronologiques.

II.3. Réseaux de Neurones

Un réseau neuronal artificiel est une application, non linéaire par rapport à ses paramètres θ qui associe à une entrée x une sortie $y = f(x, \theta)$. Par souci de simplicité, nous supposons que y est unidimensionnel, mais il pourrait également être multidimensionnel. Cette application f a une forme particulière que nous préciserons. Les réseaux de neurones peuvent être utilisés pour la régression ou la classification. Comme d'habitude dans l'apprentissage statistique, les paramètres θ sont estimés à partir d'un échantillon d'apprentissage.

[1] **Yann Le Cun**, né le 8 juillet 1960 à Soisy-sous-Montmorency, est un chercheur en intelligence artificielle et vision artificielle. Il est considéré comme l'un des inventeurs de l'apprentissage profond. Il reçoit le Prix Turing le 27 mars 2019, partagé avec Yoshua Bengio et Geoffrey Hinton.

La fonction de minimisation n'est pas convexe, conduisant à des minimiseurs locaux. Le succès de la méthode est venu d'un théorème d'approximation universel dû à Cybenko (1989) et Hornik (1991). De plus, Le Cun (1986) a proposé un moyen efficace de calculer le gradient d'un réseau neuronal, appelé rétropropagation du gradient, qui permet d'obtenir facilement un minimiseur local du critère quadratique. [20]

II.3.1. Neurone artificiel

Un neurone artificiel est une fonction fj de l'entrée $x = (x_1,..., x_d)$ pondérée par un vecteur de poids de connexion $w_j = (w_{j,1},..., w_{j,d})$, complétée par un biais de neurone b_j , et associée à une fonction d'activation Φ, à savoir

$$\boldsymbol{y_j = f_j(x) = \varphi(\langle w_j, x\rangle + b_j)}$$

Plusieurs fonctions d'activation peuvent être envisagées.

- La fonction identitaire

$$\boldsymbol{\varphi(x) = x}$$

- La fonction sigmoïde (ou logistique)

$$\boldsymbol{\varphi(x) = \frac{1}{1 + e^{-x}}}$$

- La fonction tangente hyperbolique ("tanh")

$$\boldsymbol{\varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{(2x)} - 1}{e^{(2x)} + 1}}$$

- La fonction seuil

$$\boldsymbol{\varphi_\beta(x) = 1_x \geq \beta}$$

- La fonction d'activation de l'unité linéaire rectifiée (ReLU)

$$\boldsymbol{\varphi(x) = \max(0, x)}$$

La figure 6 représente la fonction d'activation décrite ci-dessus. [20]

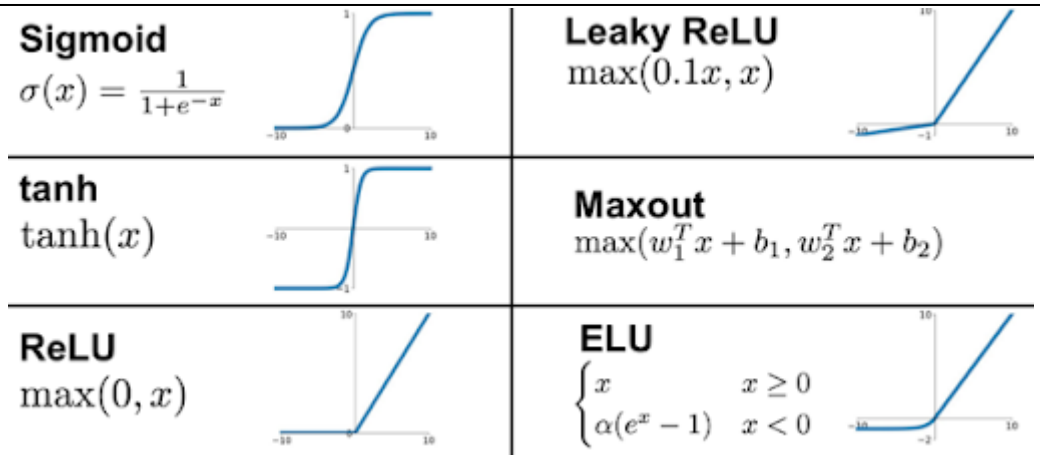


Figure II.6 Les fonctions d'activation.

Historiquement, le sigmoïde était la fonction d'activation la plus utilisée car il est différenciable et permet de conserver les valeurs dans l'intervalle [0 ; 1]. Néanmoins, elle est problématique car son gradient est très proche de 0 lorsque |x| n'est pas proche de 0.

Avec les réseaux de neurones avec un nombre élevé de couches (ce qui est le cas pour l'apprentissage en profondeur), cela cause des problèmes à l'algorithme de rétropropagation pour estimer le paramètre. C'est pourquoi la fonction sigmoïde a été supplantée par la fonction linéaire rectifiée. Cette fonction n'est pas différenciable en 0 mais en pratique ce n'est pas vraiment un problème puisque la probabilité d'avoir une entrée égale à 0 est généralement nulle. La fonction *ReLU* a également un effet de sparsification. La fonction *ReLU* et sa dérivée sont égales à 0 pour les valeurs négatives, et aucune information ne peut être obtenue dans ce cas pour une telle unité, c'est pourquoi il est conseillé d'ajouter un petit biais positif pour s'assurer que chaque unité est active. Plusieurs variations de la fonction *ReLU* sont considérées pour s'assurer que toutes les unités ont un gradient non évanouissant et que pour x <0 la dérivée n'est pas égale à 0. À savoir

$$\boldsymbol{\varphi(x) = \max(x, 0) + \alpha\, min(x, 0)}$$

Où $\boldsymbol{\alpha}$ est soit un paramètre fixe réglé sur une petite valeur positive, soit un paramètre à estimer. [20]

II.3.2. Le perceptron multicouche

Le perceptron multicouche est le type de réseau neuronal le plus connu et le plus utilisé. Dans la plupart des cas, les signaux sont transmis au sein du réseau dans une seule direction : de l'entrée à la sortie. Il n'y a pas de boucle, la sortie de chaque neurone n'affecte pas le neurone lui-même. Cette architecture est appelée feedforward (Fig.7). [21]

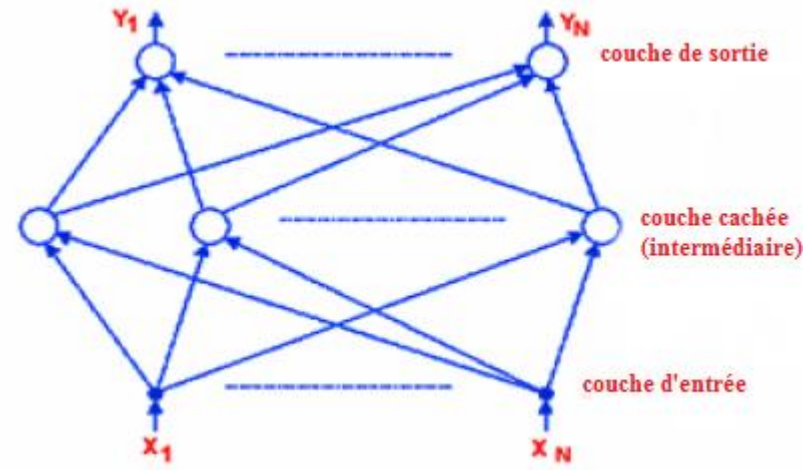


Figure II.7 Réseau de neurones multicouche à action directe.

Les couches qui ne sont pas directement connectés à l'environnement sont appelées cachées. Dans le matériau de référence, il existe une controverse concernant la première couche (la couche d'entrée) considérée comme une couche autonome dans le réseau, car sa seule fonction est de transmettre les signaux d'entrée aux couches supérieures, sans aucun traitement. Sur les entrées. Dans ce qui suit, nous ne compterons que les couches constituées de neurones autonomes, mais nous mentionnerons que les entrées sont regroupées dans la couche d'entrée.

Il existe également des réseaux de rétroaction, qui peuvent transmettre des impulsions dans les deux sens, en raison des connexions de réaction dans le réseau. Ces types de réseaux sont très puissants et peuvent être extrêmement compliqués. Ils sont dynamiques, changeant leur état tout le temps, jusqu'à ce que le réseau atteigne un état d'équilibre, et la recherche d'un nouvel équilibre se produit à chaque changement d'entrée. L'introduction de plusieurs couches a été déterminée par la nécessité d'augmenter la complexité des régions de décision.

Comme indiqué dans le paragraphe précédent, un perceptron avec une seule couche et une entrée génère des régions de décision sous la forme de semi-plans.

En ajoutant une autre couche, chaque neurone agit comme un perceptron standard pour les sorties des neurones dans la couche antérieure, ainsi la sortie du réseau peut estimer les régions de décision convexes, résultant de l'intersection des demi-plans générés par les neurones. À son tour, un perceptron à trois couches peut générer des zones de décision arbitraires. Concernant la fonction d'activation des neurones, il a été constaté que les réseaux multicouches n'augmentent pas la puissance de calcul par rapport aux réseaux à une seule couche, si les fonctions d'activation sont linéaires, car une fonction linéaire des fonctions linéaires est aussi une fonction linéaire. [21]

La puissance du perceptron multicouche provient précisément des fonctions d'activation non linéaires. Presque toutes les fonctions non linéaires peuvent être utilisées à cette fin, à l'exception des fonctions polynomiales. Actuellement, les fonctions les plus couramment utilisées aujourd'hui sont le sigmoïde unipolaire (ou logistique), illustré à la figure 8 :

$$f(s) = \frac{1}{1 + e^{-s}}$$

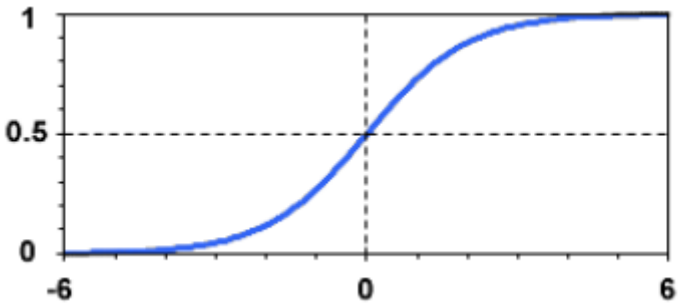


Figure II.8 Fonction d'activation unipolaire sigmoïde.

II.4. Réseaux de Neurones Convolutifs

Pour certains types de données, notamment pour les images, les perceptrons multicouches ne sont pas bien adaptés. En effet, ils sont définis pour les vecteurs comme des données d'entrée, donc, pour les appliquer aux images, il faut transformer les images en vecteurs, en perdant par la façon dont les informations spatiales contenues dans les images, comme les formes. Avant le développement de l'apprentissage en profondeur pour la vision par ordinateur, l'apprentissage était basé sur l'extraction de variables d'intérêt, appelées fonctionnalités, mais ces méthodes nécessitent beaucoup d'expérience pour le traitement d'images. Les réseaux de neurones convolutifs (CNN) introduits par LeCun [22] ont révolutionné le traitement d'image et supprimé l'extraction manuelle des fonctionnalités. CNN agit directement sur les matrices, voire sur les tenseurs pour les images avec trois canaux de

couleurs RGB. Les CNN sont désormais largement utilisés pour la classification d'images, la segmentation d'images, la reconnaissance d'objets, la reconnaissance faciale.

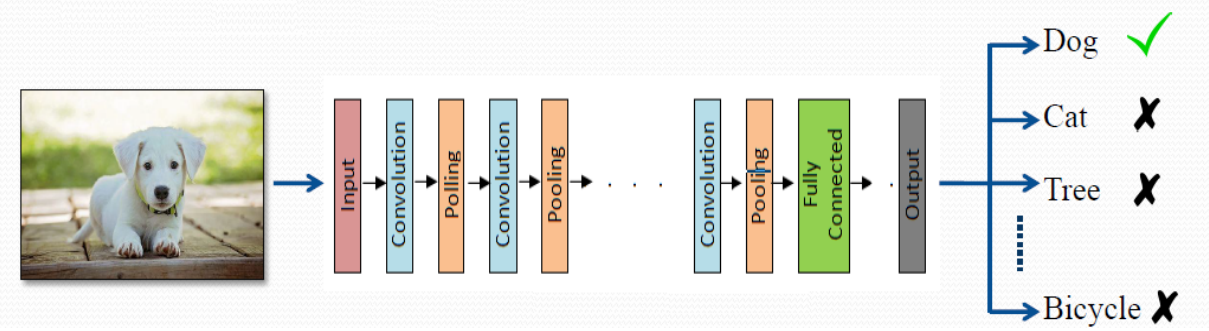


Figure II.9 Réseau de Neurone Convolutif.

II.4.1. Fonctionnement de l'algorithme

Cette section explique brièvement le fonctionnement de l'algorithme CNN. L'entrée sur le réseau est une image 2D. Le réseau a une couche d'entrée qui prend l'image comme entrée, une couche de sortie d'où nous obtenons la sortie entraînée et les couches intermédiaires appelées « couches cachées ». Comme indiqué précédemment, le réseau comporte une série de couches convolutives et de sous-échantillonnage. Ensemble, les couches produisent une approximation des données d'image d'entrée. Les CNN exploitent la corrélation spatialement locale en imposant un modèle de connectivité locale entre les neurones des couches adjacentes [24]. Les neurones de la couche disent, «*l*» sont connectés à un sous-ensemble local de neurones de la couche précédente de (*l*-1), où les neurones de la couche (*l*-1) ont des champs récepteurs contigus, comme le montre la figure (10). [25]

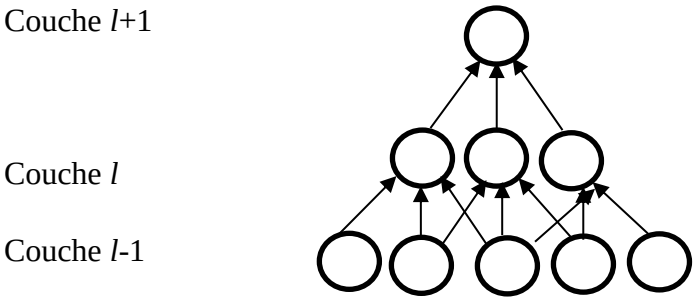


Figure II.10 Flux graphique des couches montrant la connexion entre les couches.

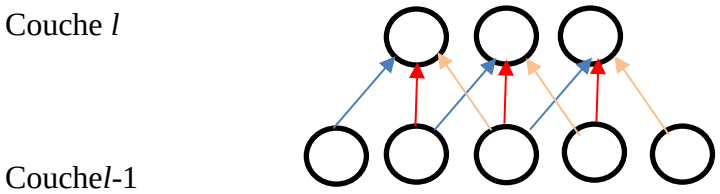


Figure II.11 Flux graphique des couches montrant le partage des poids.

Dans l'algorithme CNN, chaque filtre clairsemé est répliqué sur l'ensemble du champ visuel. Ces unités forment ensuite des cartes d'entités, qui partagent le vecteur de poids et le biais. La figure 11 représente trois unités cachées de la même carte d'entités. Les poids de même couleur sont partagés, donc contraints d'être identiques.

Le gradient des poids partagés est la somme des gradients des paramètres partagés. Une telle réplication permet en quelque sorte de détecter des caractéristiques quelle que soit leur position dans le champ visuel. En plus de cela, le partage de poids permet également de réduire le nombre de paramètres d'apprentissage gratuits. Grâce à ce contrôle, CNN tend à obtenir une meilleure généralisation des problèmes de vision.

CNN utilise également le concept de max-pooling, qui est une forme de sous-échantillonnage non linéaire. Dans cette méthode, l'image d'entrée est partitionnée en rectangles qui ne se chevauchent pas. La sortie pour chaque sous-région est la valeur maximale.

Le réseau de neurones convolutif montre des performances de pointe dans la classification des images. Deep Neural Network - trouve une structure complexe dans un grand ensemble de données en utilisant l'algorithme de rétro propagation. [23]

II.4.2. Architecture du réseau de neurone convolutionnel

L'architecture d'un ConvNet est analogue à celle du modèle de connectivité des neurones dans le cerveau humain et s'inspire de l'organisation du cortex visuel. Les neurones individuels ne répondent aux stimuli que dans une région restreinte du champ visuel connue sous le nom de champ récepteur, une collection de ces champs se chevauchent pour couvrir toute la zone visuelle.

Le CNN se compose de trois types principaux de couches: couches convolutives; des couches de sous-échantillonnage et une couche de sortie, les couches de réseau étant disposées dans une structure prospective: une couche de sous-échantillonnage suit chaque couche convolutionnelle et la couche de sortie suit la dernière couche convolutionnelle.

Les couches de convolution et de sous-échantillonnage sont considérées comme des couches 2D tandis que la couche de sortie est considérée comme une couche 1D, Dans CNN chaque 2D a plusieurs plans, Un plan se compose de neurones qui sont disposés en réseau 2D, La sortie d'un plan est appelée une carte des caractéristiques.[40]

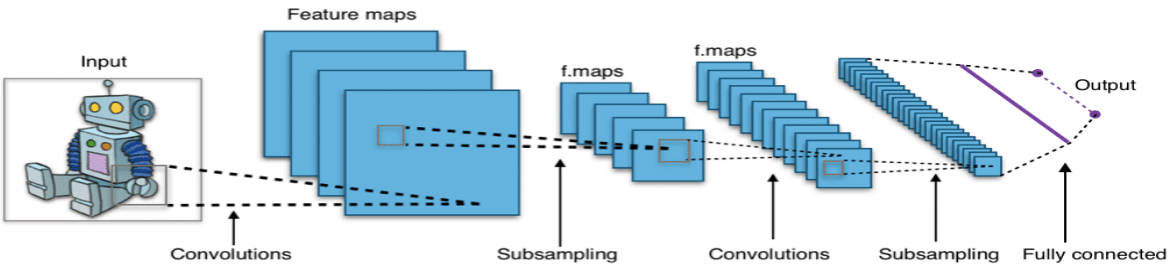


Figure II.12 Architecture des couches du réseau de neurones convolutionnels.

II.4.3. Notation mathématiques

Le tableau 2 résume les notations utilisées pour décrire les aspects fonctionnels de CNN, le symbole l indique l'indice d'une couche réseau, N^l le nombre de caractéristiques de la carte dans la couche, et f^l la fonction d'activation de la couche l, soit y_j^l le jème comporte la carte (sortie) de la couche l.

Tableau II.2. Notation d'architecture de CNN. [40]

Description	Symbole
La taille de l'image	$H * W$
La taille du filtre (Kernel)	$F * F$
Index de couche	L
Nombre de canaux dans l'image	C
Taille des entités tenseur de carte dans la couche l	$H^l * W^l * D^l$
Carte des caractéristiques d'entrée	X
Carte des caractéristiques de sortie	Y
Nombre de la carte d'entités dans la couche l ou le nombre de filtre dans l	N^l
Biais de la carte d'entités de la couche l-1 à la carte d'entités j^l	b^l_j
Un noyau appliqué sur la carte d'entités dans la couche l-1 pour produire une sortie en j^l	$W^l_{i,j}$
Padding	P
Stride	S
Fonction d'activation dans la couche l	F^l

II.4.4.Réseau Neuronal multicouche

Composé de :

- Blocs simples ou multiples de couches convolutionnelles et de mise en commun
- Une ou plusieurs couches entièrement connectées
- Une couche de sortie [23]

II.4.4.1. Couche Convolutionnelle

L'opération de convolution extrait différentes fonctionnalités à l'aide de filtres appris. [23]

La couche de convolution est la première couche du réseau CNN. La structure de cette couche est représentée sur la figure 13. Elle se compose d'un masque de convolution, de termes de biais et d'une expression d'expression de fonction. Ensemble, ceux-ci génèrent la sortie de la couche. La figure ci-dessous montre un masque 5x5 qui effectue une convolution

sur une carte d'entités d'entrée 32x32. La sortie résultante est une matrice 28x28. Ensuite, un biais est ajouté et la fonction sigmoïde est appliquée sur la matrice. [25]

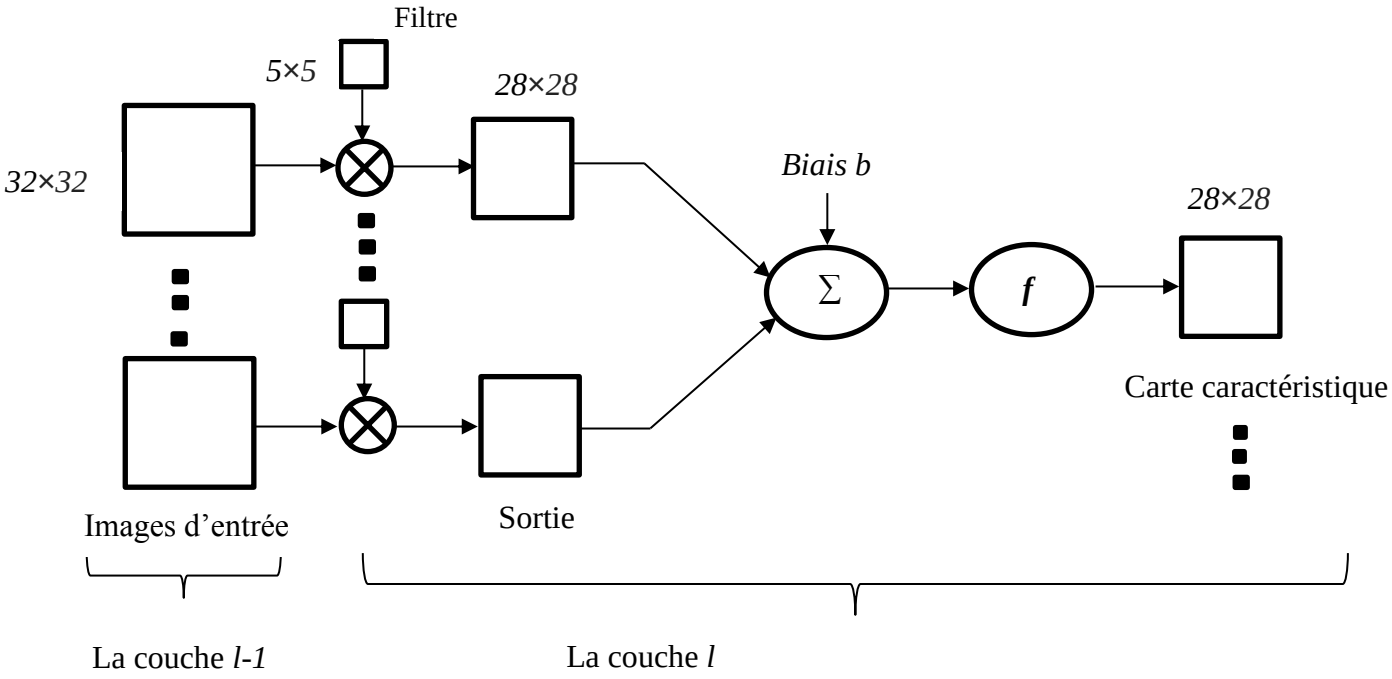


Figure II.13 Fonctionnement de la couche convolutionelle. [25]

Nous pouvons résumer ce qui se passe avec les équations :

$$y_j^l = f^l(\sum_{i\epsilon Mj} x_i^{l-1} \otimes w_{i,j}^l + b_j^l)$$

$\boldsymbol{x_i^{l-1}}$Est la carte des caractéristiques de la couche *l-1*.

$\boldsymbol{y_j^l}$ Est la carte d'entités dans la couche *l*.

$\boldsymbol{w_{i,j}^l}$ Est le noyau d'entrée dans la couche *l-1* pour la sortie dans la couche *l*. *i*, *j* le nombre de noyaux (ou filtres) est l'hyper paramètre qui représente la profondeur de la sortie dans la couche *l*.

$\boldsymbol{b_j^l}$ Est le biais associé à la couche *l*.

$\otimes$ Désigne l'opération de corrélation croisée. L'opération est expliquée dans la figure 14.

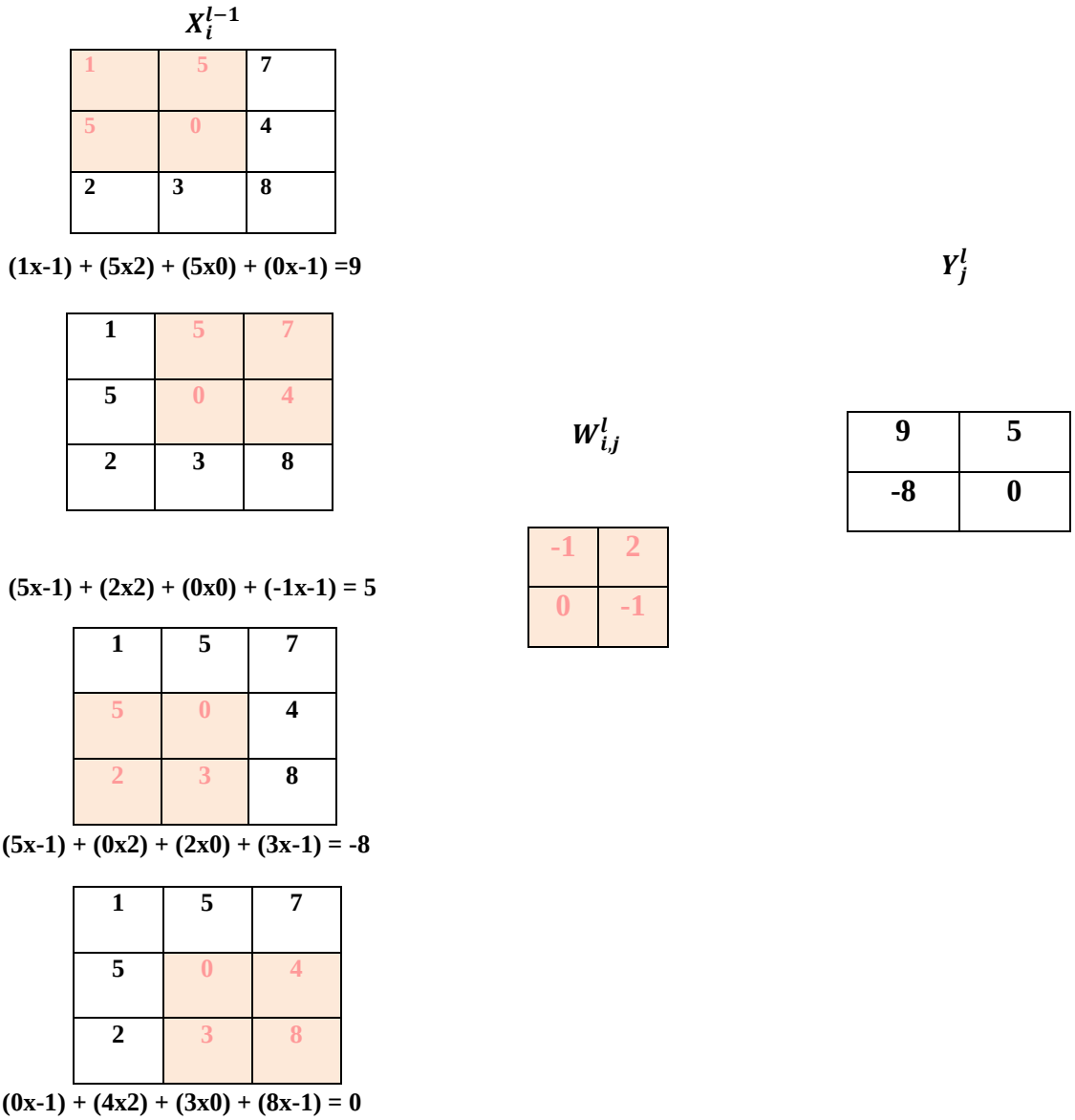


Figure II.14 Opération de corrélation croisée.

f^l Est la fonction d'activation dans la couche *l*. dans un réseau convolutionnel, *ReLu* est souvent choisie comme fonction d'activation.

$$f(x) = Max(0, x)$$

M_j représente une sélection de la carte d'entrée.

La taille du filtre affecte la sortie, comme le montre la figure 14. La taille dépend également des valeurs de Padding (P) et de Stride (S) et du filtre F. nous pouvons calculer la hauteur et la largeur de sortie avec l'équation :

$$H^l = \frac{H^{l-1} + 2P - F}{S} + 1 \qquad W^l = \frac{H^{l-1} + 2P - F}{S} + 1$$

Alors que la profondeur de la profondeur de sortie est le nombre de noyaux N_k.

Donc nous avons :

$$(H^l, W^l, N_C) \otimes (F, F, N_C) = \frac{H^{l-1} + 2P - F}{S} + 1\ , \frac{H^{l-1} + 2P - F}{S} + 1, N_K$$

Où N_C est le nombre de canaux et *F* est la hauteur et la largeur du filtre.

II.4.4.2. Couche de sous-échantillonnage

Le but des couches de sous-échantillonnage (ou des couches de mise en commun) est de réduire la taille spatiale de l'entrée, ce qui réduit le nombre de paramètres et facilite les calculs, ce qui en retour aide à lutter contre le sur ajustement. Il existe deux opérations de mise en commun communes, la mise en commun maximale et la mise en commun moyenne, toutes deux expliquées dans la figure 15.

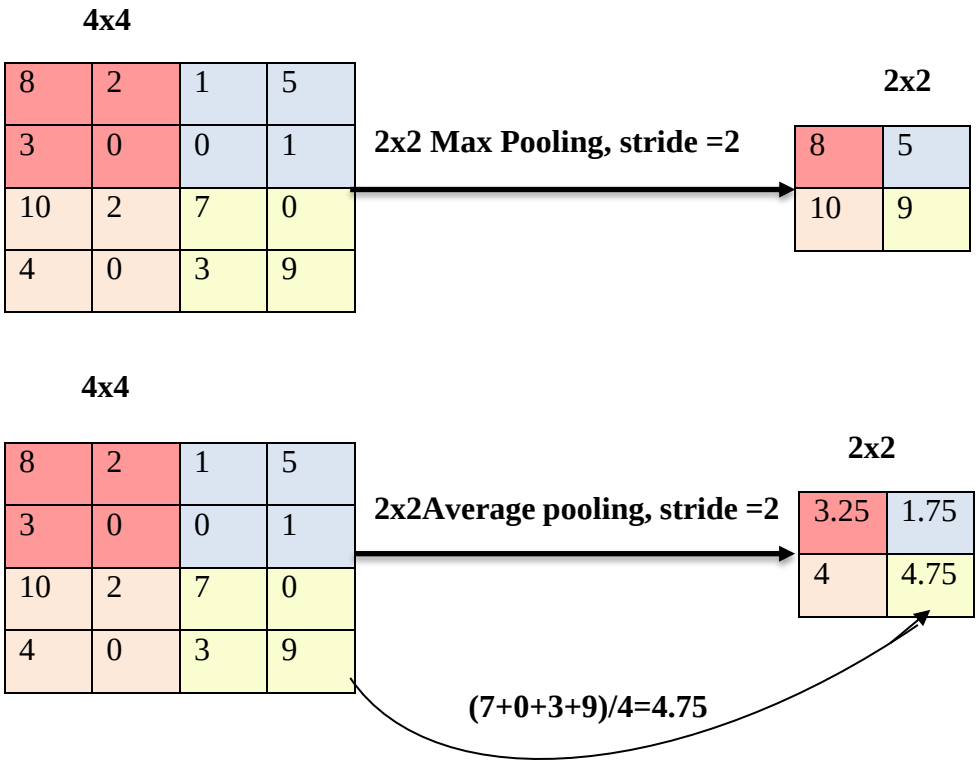


Figure II.15 Les opérations max pooling et average pooling.

La taille de regroupement la plus utilisée est 2x2 avec une stride S = 2. Il n'y a pas de règle spécifique sur la meilleure forme de regroupement, le choix dépend généralement de ce qui donne les meilleurs résultats. [40]

La taille de sortie des couches de mise en commun dépend de la taille spatiale F et de le stride S comme suit :

$$W^{l+1} = {(W^l - f)}/{S+1} \qquad H^{l+1} = {(H^l - f)}/{S+1}$$

Alors que la profondeur D reste la même.

II.4.4.3. Couche entièrement connectée

Dans la dernière couche de convolution, nous transformons la carte des entités, qui est maintenant un tenseur de dimensions (W, H, D), en un vecteur de taille W * H * D (figure 16). Ce vecteur est la première couche du FCL (Fully Connected Layer), qui n'est qu'un autre nom du perceptron multicouche (MLP). Après avoir extrait les entités à l'aide de couches convolutives, nous utilisons FCL pour classer nos images en fonction de ces nouvelles fonctionnalités. [40]

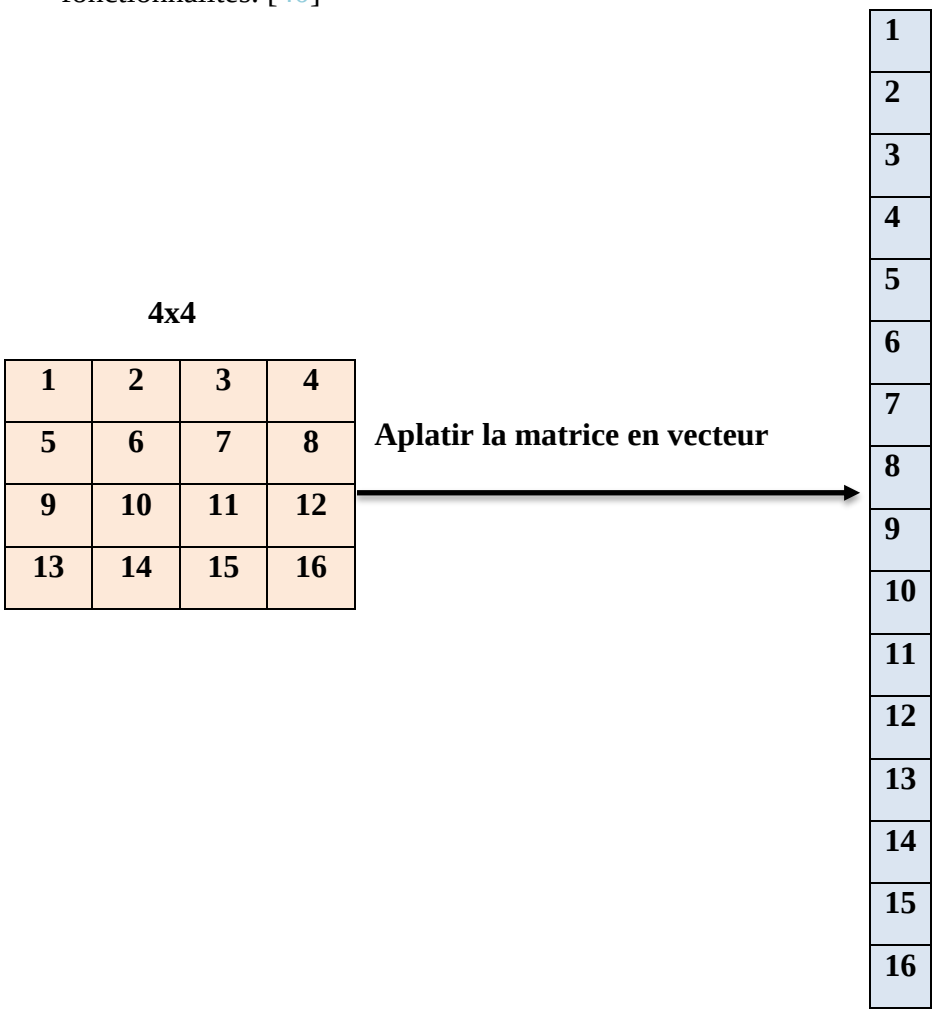


Figure II.16 Aplatir une matrice en vecteur.

Dans FCL, pour calculer la sortie de la nouvelle couche y_n^l, nous utilisons l'opération de produit scalaire entre le vecteur d'entrée x_m^{l-1}et la matrice de pondération correspondante $w_{m,n}^l$, puis en ajoutant le biais b_n^lcomme suit :

$$y_n^l = f\left(\sum_{m=1}^{N^{l-1}} x_m^{l-1} * W_{m,n}^l + b_n^l\right)$$

Le choix de la fonction d'activation (ReLu, Sigmoïde, tanh) dépend totalement des résultats, cependant, dans la dernière couche, et dans le cas d'une classification multi-étiquettes comme dans la classification des images, Relu est généralement utilisé.

$$\mathrm{f(x)} = \max(0, x)$$

II.5. Apprentissage du réseau de neurones convolutif

Pour que les CNN effectuent différentes tâches de reconnaissance visuelle, un algorithme d'apprentissage doit être conçu. Dans une application donnée, l'objectif d'apprentissage du réseau est de minimiser la fonction d'erreur, qui est définie en termes de sorties réelles du réseau et de sorties souhaitées.

Supposons que l'ensemble d'apprentissage a n images d'entrée et k sorties, Soit E l'erreur pour toutes les images d'entrée qui est définie comme :

$$E = \sum_{n=1}^{N^l} E^{(n)}$$

Et l'erreur la moins carrée $E\,(n)$ est définie comme :

$$E^{(n)} = \frac{1}{2}\sum_{K=1}^{K}\left(d_K - y_k^l\right)2$$

Où d_K est la sortie réelle du réseau et y_k^lest la prédiction du réseau.

II.5.1 Ajustement des poids et des biais :

Nous mettons à jour chaque poids en trouvant la dérivée de la fonction d'erreur par rapport à ce poids (ou biais) spécifique :

$$W_{ij}^l := W_{i,j}^l - \alpha \frac{dE}{dW_{ij}^l}$$

$$b_j^l := b_j^l - \alpha \frac{dE}{db_j^l}$$

Où α est le taux d'apprentissage

Pour faciliter les calculs, nous utilisons l'informatique vectorisée ; c'est-à-dire, plutôt que d'opérer sur une seule valeur à la fois ; nous opérons sur des vecteurs et des matrices.

- **Dans la couche entièrement connectée :**

Supposons $u(l) = W^l x^{l-1} + b^l$ et $x^l = f(u^l)$

Où W^l est les matrices de poids dans la couche *l*. b^lest un vecteur de biais, x^{l-1} est un vecteur d'entrée dans la couche *l*-1 et *f* est une fonction d'activation.

En utilisant la règle de chaîne, nous pouvons simplifier

$$\frac{dE}{dW^l} = \frac{dE}{dx^l}\frac{dx^l}{du^l}\frac{du^l}{dW^l} \quad , \delta^l = \frac{dE}{dx^l}\frac{dx^l}{du^l}$$

Dans la dernière couche (la couche de sortie), nous pouvons calculer δ^l:

$$\delta^l = (d_K - y_k^l) \circ f'(u^l)$$

Où ° indique une multiplication par élément.

Comme pour les autres couches en FCL on prend le produit scalaire entre la matrice de poids et le vecteur des gradients locaux δ^{l+1} et on le multiplie par élément avec le vecteur de dérivée de la fonction d'activation de la couche courante :

$$\delta^l = (W^{l+1})^T \delta^{l+1} \circ f'(u^l)$$

De plus, nous avons :

$$\frac{du^l}{dW^l} = x^{l-1}$$

Au final, on peut avoir la matrice des gradients des poids dans la couche *l* $\frac{dE}{dW^l}$ en calculant le produit extérieur du vecteur d'entrées x^{l-1} et du vecteur des «sensibilités» δ^l:

$$\frac{dE}{dW^l} = x^{l-1} (\delta^l)^T$$

Quant aux biais, et parce que $\frac{du}{db}$ =1 :

$$\frac{dE}{db^l} = \frac{dE}{dx^l}\frac{dx^l}{du^l}\frac{du^l}{db^l} = \frac{dE}{du^l} = \delta^l$$

– **Dans les couches de sous-échantillonnage :**

Dans le pool max, les valeurs qui ne sont pas égales au maximum, c'est-à-dire si :

$$x_i^l \neq \max(x_k^l)$$

Où x_i^l est le pixel i dans la carte des entités X^let x_k^lest le bloc de pixels dans la carte des entités X^lqui était couvert par la fonction max pooling.

Ces neurones recevront un gradient de valeur = zéro (figure 17). Parce que les modifier légèrement n'affecte pas du tout la sortie.

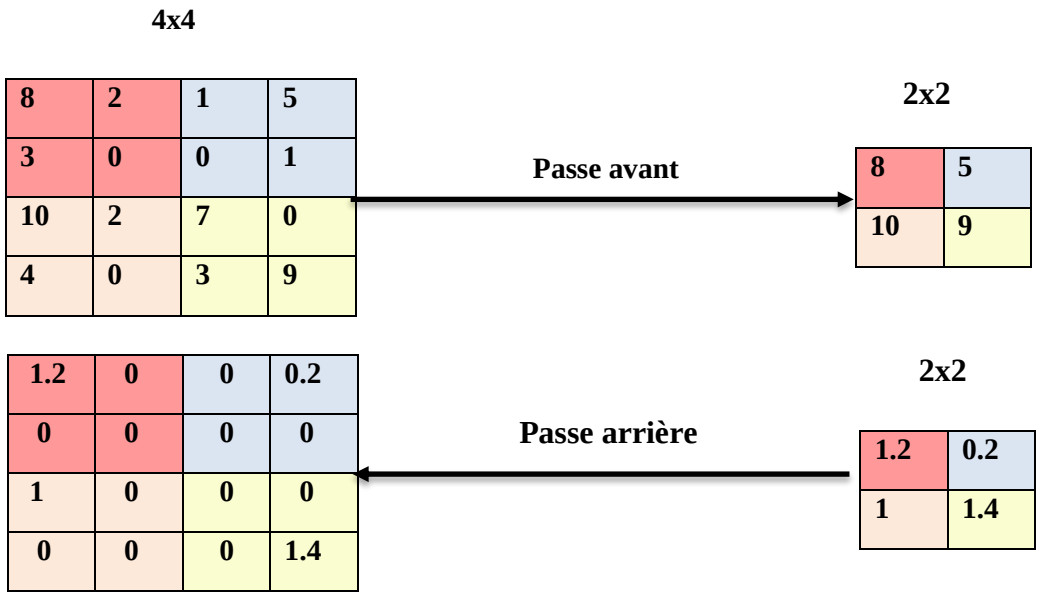


Figure II.17 Passage des gradients de max pooling.

En average pooling, tous les neurones reçoivent le gradient ; car ils affectent tous la sortie. Le gradient donné comme :$\delta_i^l = \frac{1}{n} * \delta_i^{l+1}$ où $\delta^{l+1} = \frac{dE}{du^{l+1}}$ et n est la taille du bloc.

La figure 18 explique le processus de passage du gradient en cas d'average pooling, nous voyons que l'erreur est divisée par n = 4 qui est le nombre total couvert dans un bloc de l'opération pooling. [26]

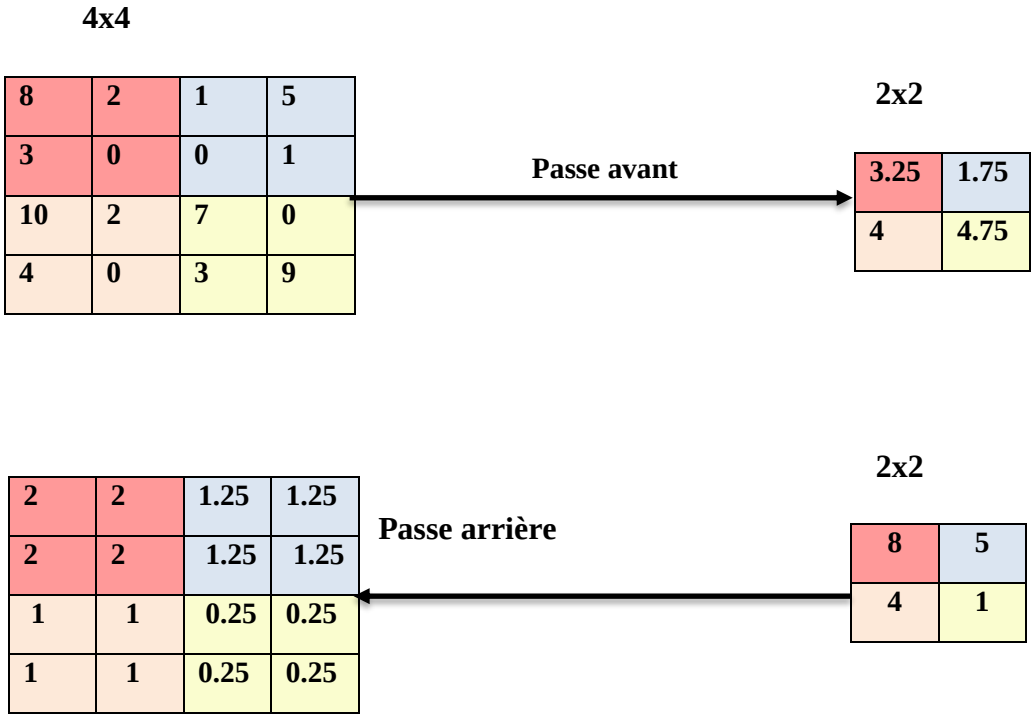


Figure II.18 Passage des gradients d'average pooling.

– **Couche convolutionelle**

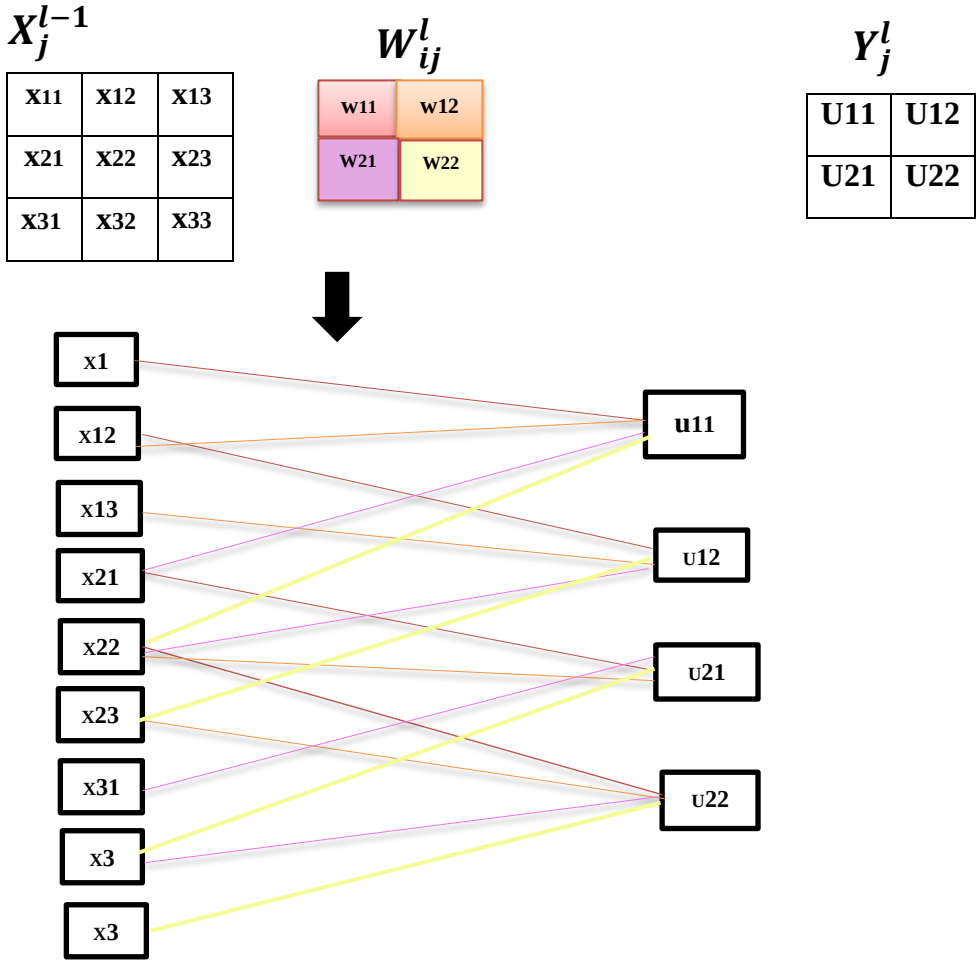


Figure II.19 Les poids sont partagés dans de nombreux neurones d'entrée pendant l'opération de convolution.

U11 = x11w11 + x12w12 + x21w21+ x22w22+ b

U12 = x12w11 + x13w12 + x22w21 + x23w22+ b

U21 = x21w11 + x22w12 + x31w21 + x32w22 + b

U22 = x22w11 + x23w12 + x32w21 + x33w22 + b

Le calcul des gradients dans les couches convolutives suit les mêmes règles que les couches entièrement connectées (FCL). La différence est cette fois les poids partagés entre plusieurs unités comme le montre la figure 19. Nous faisons la somme de tous ces gradients qui partagent le même poids.

$$\frac{dE}{dW_{ij}^{l}} = \sum_{u,v} (\delta_j^l)_{uv} \left(A_i^{l-1}\right)uv$$

Où δ_j^lest le gradient local : $\delta_j^l = \frac{dE}{dx^l}\frac{dx^l}{du^l}Et\left(A_i^{l-1}\right)uv$ est l'aire de X_i^{l-1}qui a été multipliée par W_{ij}^{l}.

La même chose vaut pour le biais, on additionne tous les gradients affectés par ce biais très spécifique que l'on veut mettre à jour :

$$\frac{dE}{db_j^l} = \sum_{u,v}(\delta_j^l)_{uv}$$

Parce que $\frac{du^l}{db^l} = 1$

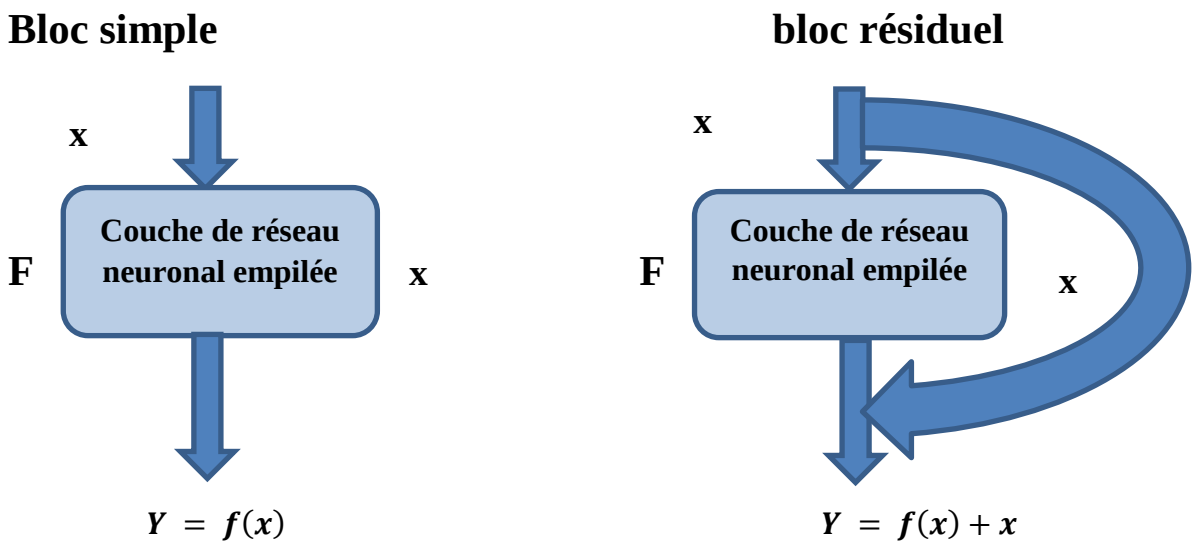
II.6. Réseau Résiduel

II.6.1. Définition

Le réseau neuronal résiduel (ResNet) est un réseau neuronal qui utilise des connexions ou des raccourcis pour sauter par-dessus certaines couches [27].Le but de ces connexions est de résoudre les problèmes de gradient et de neurones morts disparaissant, qui sont assez répandus dans les réseaux convolutionnels profonds [28].Ces raccourcis sont toujours actifs car leur gradient n'est pas un petit nombre (nous entendons par petit nombre un nombre proche de zéro) qui aident le réseau à s'entraîner plus rapidement.

L'intuition derrière ResNet est plutôt que d'apprendre un mappage direct $H\,(x)$, il est préférable d'apprendre un mappage résiduel $F\,(x) = H\,(x) - x$ qui peut être écrit comme $H\,(x) = F\,(x) + x$ où $F\,(x)$ est la sortie des couches empilées et *x* est la sortie de la

fonction d'identité.



FigureII.20 Mappage d'identité dans RESNET.

- **Propagation arrière dans ResNet :**

La rétro propagation dans ResNet est similaire à celle de CNN, la seule différence est le passage des gradients dans les blocs résiduels :

Supposons que *l* soit un indice de la couche bloc et que nous ayons :

$$x^{l+1} = x^{l} + f(x^{l} + W^{l})$$

Le calcul des gradients de x^{l} donne :

$$\frac{dE}{dx^{l}} = \frac{dE}{dx^{l+1}}\frac{dx^{l+1}}{dx^{l}} = \frac{dE}{dx^{l+1}}\left(1 + \frac{df(x^{l},W^{l})}{dx^{l}}\right)$$

Cela garantit que même si $\frac{df(x^{l},W^{l})}{dx^{l}}$ disparaît, c'est-à-dire même si $\frac{df(x^{l},W^{l})}{dx^{l}} \cong 0$, le gradient de $\frac{dE}{dx^{l}}$ ne sera jamais proche de zéro.

II.5.2. Bloc résiduel

Il existe deux types de blocs dans les réseaux résiduels, la principale différence dans le changement des dimensions de l'entrée ou de la sortie. [29]

- **Bloc d'identité**

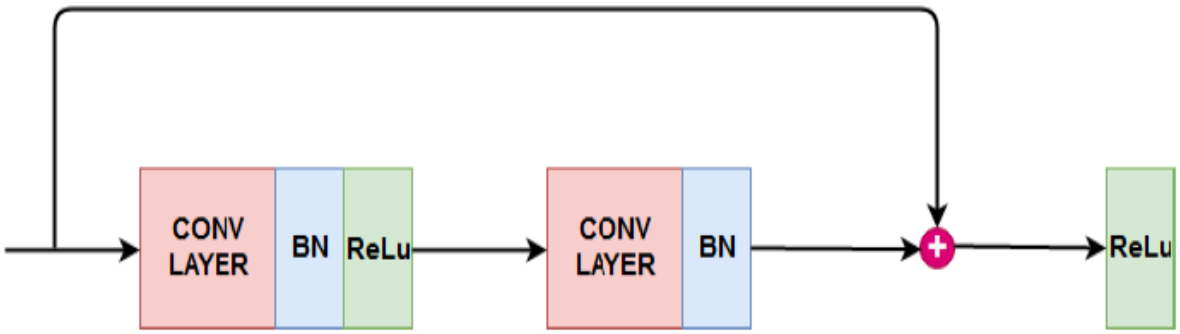


Figure II.21 Bloc d'identité dans ResNet.

Le bloc d'identité est utilisé dans le cas où les dimensions de la couche d'entrée *l* sont les mêmes que celles de la couche de sortie *l* + 2 comme le montre la figure 21.

- **Bloc convolutionnel**

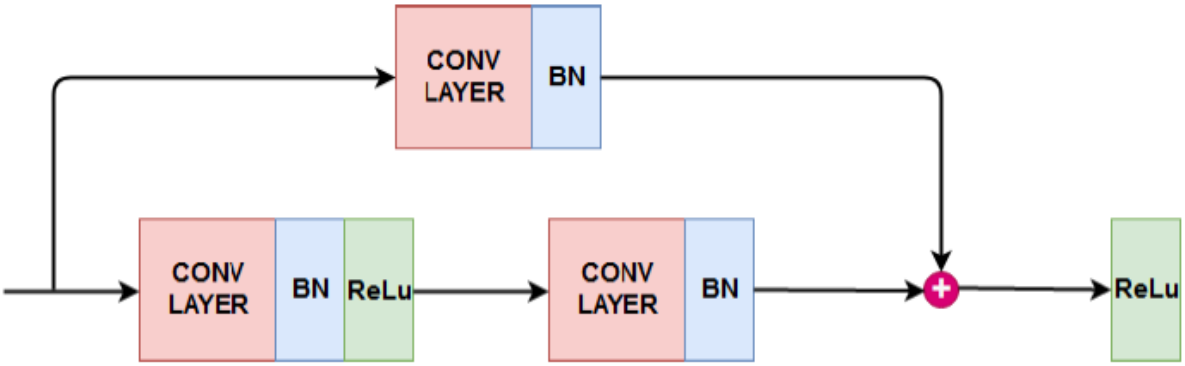


Figure II.22 Bloc convolutif dans ResNet.

Dans le cas où les dimensions de l'entrée changent, nous sous-échantillonnons l'entrée dans le raccourci en ajoutant une couche convolutionnelle (figure 22) de sorte que la dimension de l'entrée et de la sortie devient la même.

II.7. l'Apprentissage par transfert

II.7.1. Définition

L'apprentissage par transfert est une méthode d'apprentissage automatique dans laquelle un modèle développé pour une tâche est réutilisé comme point de départ pour un modèle sur une deuxième tâche. Il s'agit d'une approche populaire en apprentissage profond où des modèles préformés sont utilisés comme point de départ pour les tâches de vision par ordinateur et de

traitement du langage naturel, étant donné les vastes ressources de calcul et de temps nécessaires pour développer des modèles de réseau neuronal sur ces problèmes et des énormes sauts de compétences qu'ils fournissent sur les problèmes connexes. Le transfert d'apprentissage est l'amélioration de l'apprentissage dans une nouvelle tâche par le transfert de connaissances d'une tâche connexe qui a déjà été apprise.

Dans l'apprentissage par transfert, nous formons d'abord un réseau de base sur un ensemble de données et une tâche de base, puis nous réaffectons les fonctionnalités apprises, ou les transférons, à un deuxième réseau cible pour être formés sur un ensemble de données et une tâche cibles. Ce processus aura tendance à fonctionner si les fonctionnalités sont générales, c'est-à-dire adaptées aux tâches de base et cibles, au lieu d'être spécifiques à la tâche de base. [30]

II.7.2. Transfert d'apprentissage avec des données d'image

Il est courant d'effectuer un apprentissage par transfert avec des problèmes de modélisation prédictive qui utilisent des données d'image en entrée. Cela peut être une tâche de prédiction qui prend des photos ou des données vidéo en entrée.

Pour ces types de problèmes, il est courant d'utiliser un modèle d'apprentissage approfondi préformé pour une tâche de classification d'images volumineuse et difficile, comme le concours de classification de photographies de classe ImageNet.

Les organismes de recherche qui développent des modèles pour ce concours et publient bien souvent leur modèle final sous une licence permissive de réutilisation. Ces modèles peuvent prendre des jours ou des semaines pour s'entraîner sur du matériel moderne.

Exemple de modèles de ce type :

- Modèle VGG
- Modèle de création de Google
- Modèle ResNet
- Modèle AlexNet.

Cette approche est efficaces car les images ont été formées sur un grand corpus de photographies et nécessitent que le modèle fasse des prédictions sur un nombre relativement important de classes, ce qui nécessite que le modèle apprenne efficacement à extraire des caractéristiques des photographies afin de bien fonctionner sur le problème.[30]

II.7.3. Avantages de l'apprentissage par transfert

L'apprentissage par transfert est une optimisation, un raccourci pour gagner du temps ou obtenir de meilleures performances. En général, il n'est pas évident qu'il y aura un avantage à utiliser l'apprentissage par transfert dans le domaine tant que le modèle n'aura pas été développé et évalué.

Lisa Torrey et Jude Shavlik [47] dans leur chapitre sur l'apprentissage par transfert décrivent trois avantages possibles à rechercher lors de l'utilisation de l'apprentissage par transfert :

- **Début plus élevé**. La compétence initiale (avant d'affiner le modèle) sur le modèle source est plus élevée qu'elle ne le serait autrement.
- **Pente plus élevée**. Le taux d'amélioration des compétences lors de l'entrainement du modèle source est plus rapide qu'il ne le serait autrement.
- **Asymptote supérieure**. La compétence convergente du modèle entrainé est meilleure qu'elle ne le serait autrement.

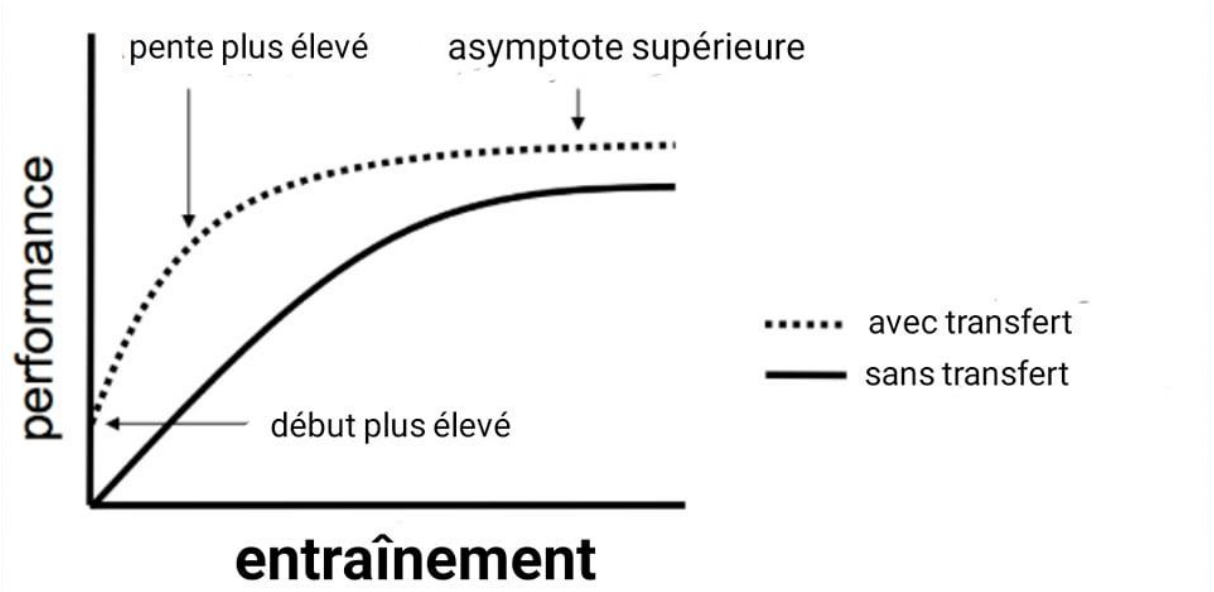


Figure II.23 Trois façons dont le transfert pourrait améliorer l'apprentissage. [30]

II.7.4. Groupe de géométrie visuel

II.7.4.1. Définition

La forme complète de VGG [31] est *VisualGeometryGroup*. Le réseau VGG contient normalement VGG16 et VGG19. Dans ce réseau, les noyaux de grande taille sont remplacés

par le nombre multiple de filtres 3x3, pour cette raison, nous extrayons des fonctionnalités complexes à faible coût. [32]

VGG était le deuxième dans ILSVRC 2014 du groupe VGG, Oxford [31]. Il fait l'amélioration par rapport à AlexNet et compte 19 couches au total. Sa principale contribution a été de montrer que la profondeur du réseau ou le nombre de couches est un élément critique pour de bonnes performances. Bien que VGGNet atteigne une précision phénoménale sur l'ensemble de données ImageNet, son déploiement sur les unités de traitement graphique (GPU) même les plus modestes est un problème en raison des exigences de calcul énormes, à la fois en termes de mémoire et de temps.

II.7.5. AlexNet

II.7.5.1. Définition

AlexNet [34] a été le premier réseau neuronal convolutionnel profond à démontrer avec succès des performances supérieures aux procédures classiques de reconnaissance d'objets d'image. Le modèle se compose de cinq couches convolutionnelles connectées séquentiellement de taille de filtre décroissante, suivies de trois couches entièrement connectées. L'une des principales caractéristiques d'AlexNet est le sous-échantillonnage très rapide des représentations intermédiaires par le biais de convolutions stridées et de couches max pooling. La dernière carte convolutionnelle est remodelée en un vecteur et traitée comme une entrée pour une séquence de deux couches entièrement connectées de 4096 unité. La sortie de cette couche représente le descripteur d'image produit par AlexNet.

II.8. Conclusion

Nous avons présenté dans ce chapitre le Deep Learning. Il s'agit d'obtenir les différents types de Deep Learning, leur fonctionnement et leurs applications. Nous avons évoqué le réseau de neurones convolutif, ses différentes couches et ses différents modèles (ResNet, VGG, AlexNet).

Chapitre III

« L’approche proposée»

III.1. Introduction

Pour atteindre la classification des images spams, il faut nécessairement vérifier le travail par des procédés avérés. Dans ce chapitre nous expliquons notre approche et les architectures que nous avons proposées, commençant avec le « *CNN* » le plus ancien des procédés qui n'a pas fait l'unanimité de tous. Il était donc nécessaire de proposé une nouvelle architecture avec un modèle différent, donc nous avons choisi d'utiliser le « *ResNet* », un procédé plus perfectionné qui nous a donné un bon résultat et pour confirmer ces résultats, il fallait utiliser le « Transfer Learning ». Ce sont des modèles préformés sur un grand Dataset que nous avons adapté selon notre objectif qui est la classification des images comme spam ou non-spam. La couche responsable de la classification est la couche entièrement connectée (*Fully Connected Layer*). Le Transfer Learning est plus efficace et communément utilisé car ses résultats sont probables et dans notre recherche il a donné les meilleurs résultats.

III.2. Réseaux de neurones convolutifs

III.2.1. l'architecture CNN utilisée

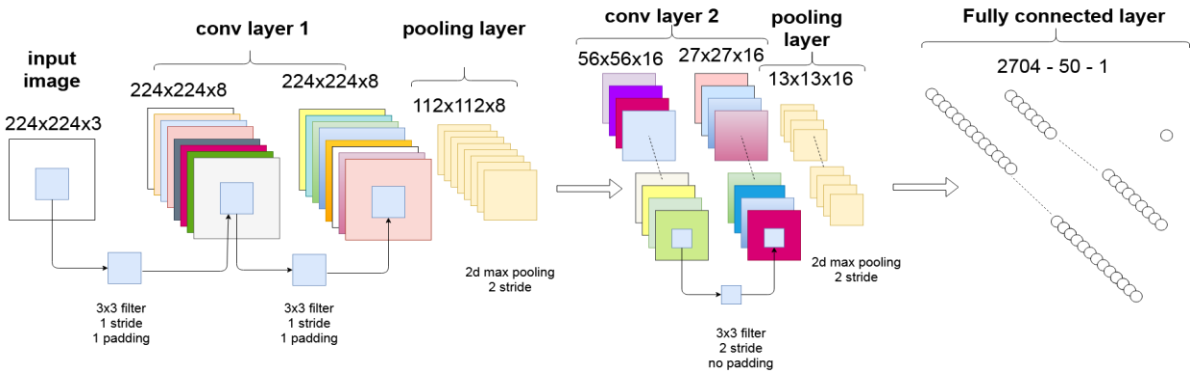


Figure III.24. Architecture CNN utilisée.

Avec CNN, nous avons introduit l'image pour la classifier, nous avons normalisé l'entrée au format (224x224). Le modèle se compose de deux couches convolutionnelles et deux couches de sous échantillonnage, suivies d'une couche entièrement connectée (FCL).

Nous avons huit filtres de taille (3x3) avec un stride =1 et padding = 1, ces huit filtres sont appliqués successivement sur l'image, chaque filtre nous donne une particularité différente des autre et ainsi de suite jusqu'au huitième, le résultat est appelé features map. Cette opération était rendue nécessaire pour obtenir plus de précision dans la classification. Ceci étant appelé la première couche convolutionnelle. Les résultats du second filtre nous donnent des features map différents, suivie de la fonction d'opérateur linéaire (RELU).

Les résultats obtenus ont été soumis à la couche de sous-échantillonnage (max pooling) qui a réduit les dimensions des huit images au format (112x112x8) car le stride =2.

Ces résultats obtenus ont été passés à la deuxième couche convolutionnelle, nous avons obtenu une nouvelle dimension des images (56x56) puis (27x27). La taille spatiale de ces outputs a été réduite par la deuxième couche de sous-échantillonnage au format (13x13) ce qui réduit le nombre de paramètres.

La dernière étape est la couche entièrement connectée (FCL), le résultat obtenu de la couche précédente sous forme de matrice de taille (13x13x16) est transformé par aplatissement en un vecteur d'entrée de taille 2704, une couche caché de longueur 50 et une seule sortie, 0 non-spam et 1 spam. La fonction d'activation utilisée au niveau de cette couche est la fonction sigmoïde car elle permet de conserver les valeurs dans l'intervalle [0 ; 1].

III.3. Réseau Résiduel

III.3.1 l'architecture ResNet utilisée

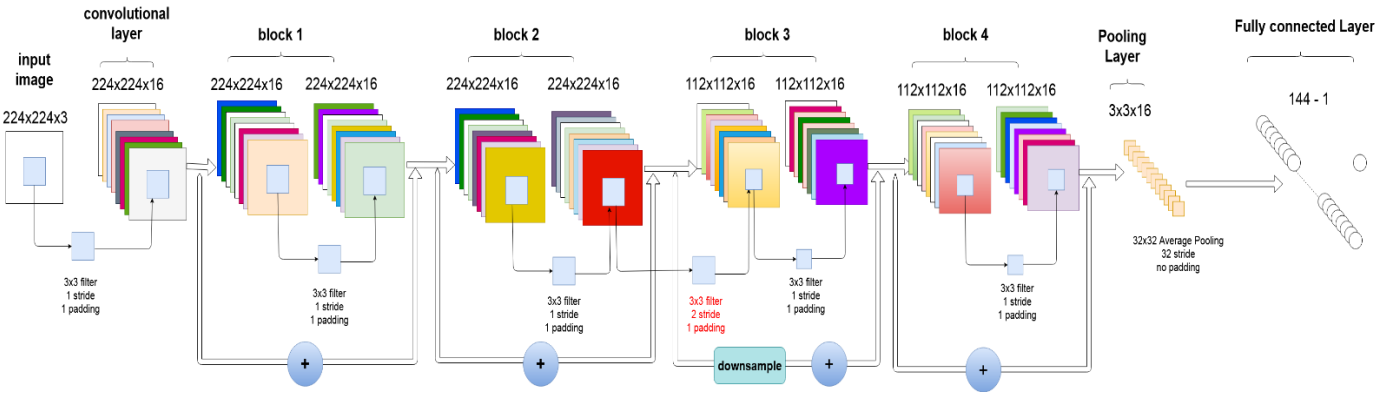


Figure III.25. Architecture ResNet utilisée.

Cette architecture ResNet contient une couche convolutionnelle, quatre blocs résiduels, une couche de sous-échantillonnage et la couche entièrement connectée.

Le format de l'entrée est (224x224), nous avons seize filtres de taille (3x3) avec un stride =1 et padding = 1. Le procédé se distingue par la particularité de transporté l'image d'entrée au bloc suivant après traitement. Le bloc 1 et 2 ne change pas les dimensions de l'image (224x224) quant au bloc 3 et 4 la taille change car le stride = 2 d'où réduction de dimension à (112x112). Au cours la couche de sous-échantillonnage nous utilisons la moyenne (average pooling) et pas le maximum commun (max pooling) car elle nous a donné un meilleur résultat, la taille est réduite au format (3x3x16).

La matrice obtenue est aplati en vecteur de dimension 144 dans la couche entièrement connecté.

III.4. l'apprentissage par transfert

III.4.1 Groupe de géométrie visuelle

III.4.1.1. Architecture du groupe de géométrie visuelle

Le réseau est composé de plusieurs couches de types et de dimensions différentes, comme indiqué ci-dessous

- Convolution utilisant 64 filtres + Max Pooling
- Convolution utilisant128 filtres + Max Pooling
- Convolution utilisant 256 filtres
- Convolution utilisant 256 filtres + Max Pooling
- Convolution utilisant 512 filtres
- Convolution à l'aide de 512 filtres + Max Pooling
- Convolution utilisant 512 filtres
- Convolution à l'aide de 512 filtres + Max Pooling
- Couche entièrement connecté avec 4096 nœuds
- Couche entièrement connecté avec 4096 nœuds
- Couche de sortie avec activation Softmax avec 1000 nœuds. [33]

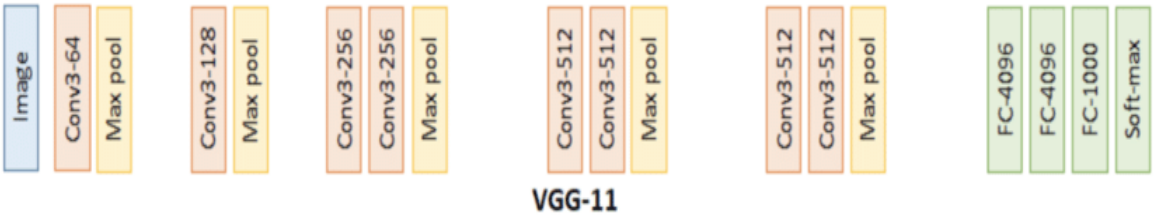


Figure III.26 Architecture du modèle VGG-11 utilisée.

III.4.2. AlexNet

III.4.2.2. Architecture du modèle AlexNet

Couche 1

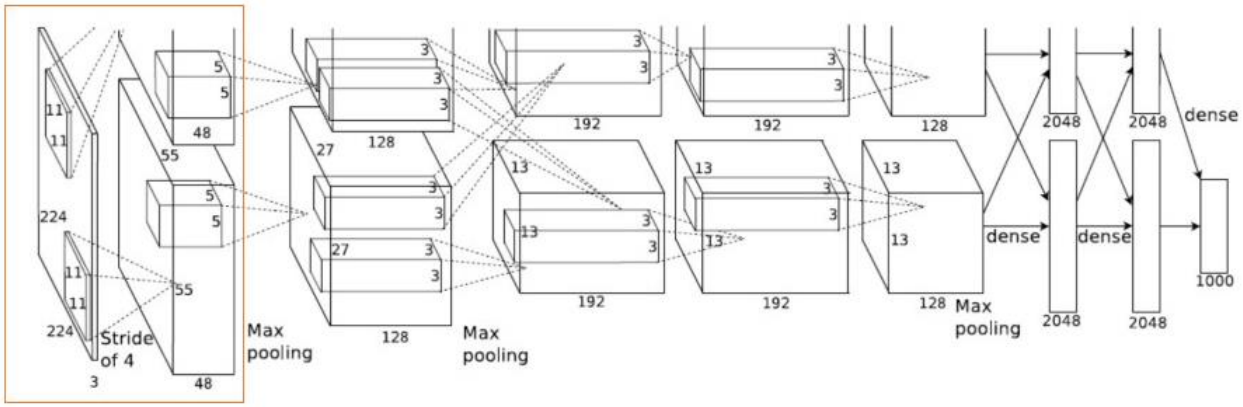


Figure III.27 Architecture du modèle AlexNet –Couche 1- [35]

- La couche 1 est une couche de convolution
- La taille de l'image d'entrée est de : 224 x 224 x 3
- Nombre de filtres : 96
- Taille du filtre : 11 x 11 x 3
- Stride : 4
- Sortie couche 1
- 224/4 x 224/4 x 96 = 55 x 55 x 96 (à cause destride : 4)

Couche 2

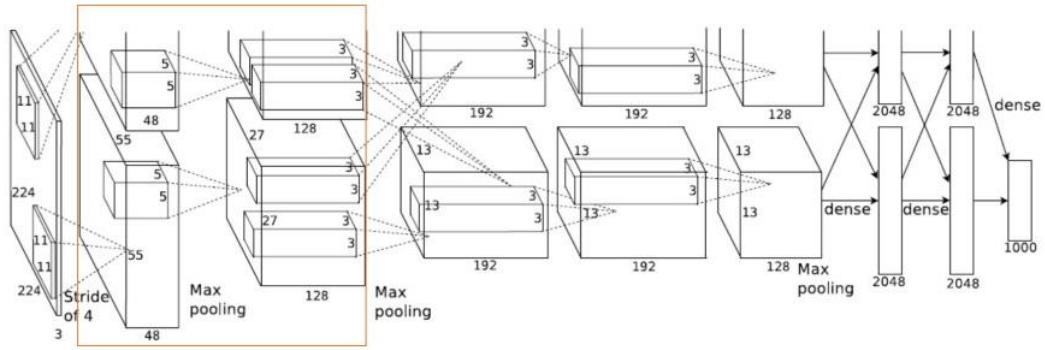


Figure III.28 Architecture du modèle AlexNet –Couche 2-

- La couche 2 est un max pooling suivi d'une convolution
- Entrée : 55 x 55 x 96
- Max pooling : 55/2 x 55/2 x 96 = 27 x 27 x 96
- Nombre de filtres : 256
- Taille du filtre : 5 x 5 x 48
- Sortie couche 2
  - 27 x 27 x 256

Couche 6

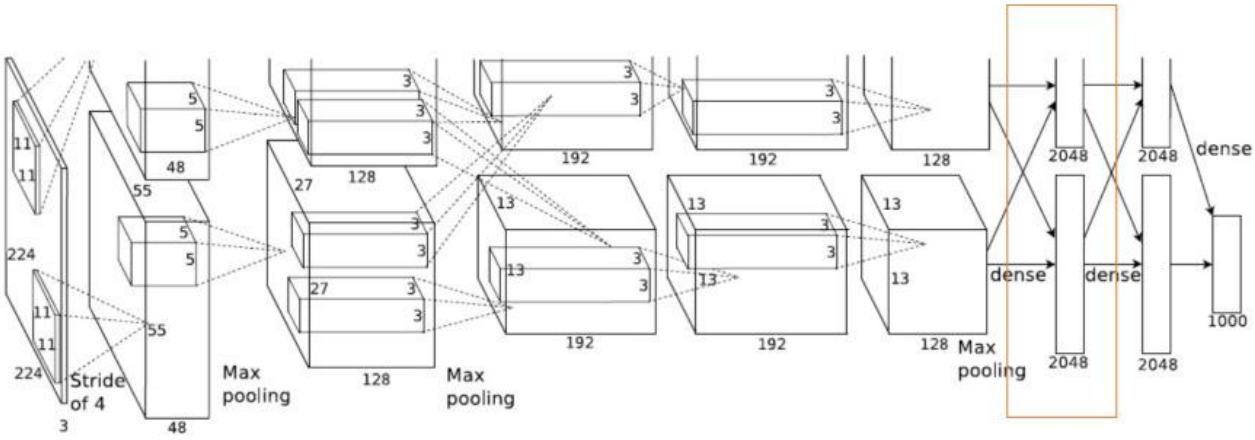


Figure III.29 Architecture du modèle AlexNet –Couche 6-

- La couche 6 est entièrement connectée
- Entrée - 13 x 13 x 128 -> est transformés en vecteur
- Et multiplié par une matrice de la dimension suivante : (13 x 13 x 128) x 2048
- GEMV (General Matrix Vector Multiply) est utilisé ici :
- Vecteur X = 1 x (13x13x128)
- Matrice A = (13x13x128) x 2048 : Il s'agit d'une entrée externe au réseau
- La sortie est : 1 x 2048 [35]

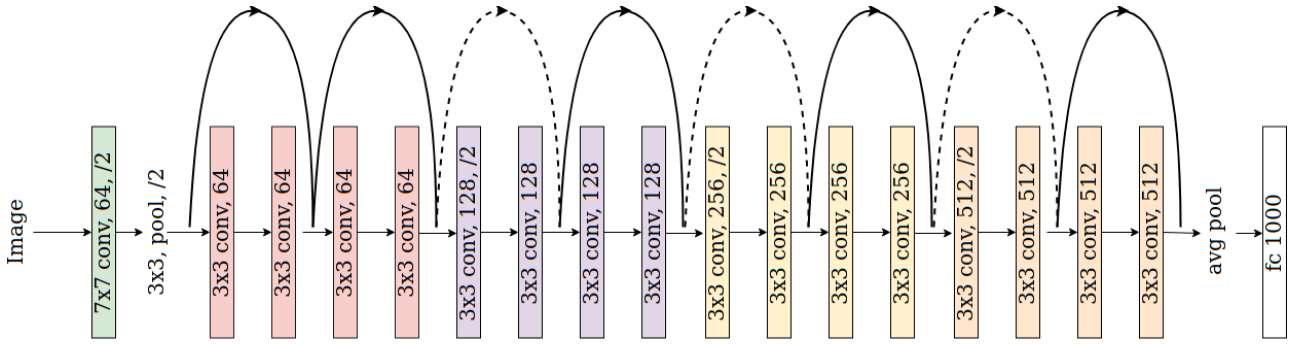


Figure III.30 Architecture du modèle ResNet-18 [36]

Tous les modèles vus au-dessus (AlexNet, VGG-11, ResNet18) sont des modèles déjà entrainés sur de grands Datasets (ImageNet) qui contiennent toutes les informations que nous pouvons extraire d'une image. Notre Dataset qui contient seulement les images spam et non-spam est petit par rapport au Dataset utilisé dans le Transfer Learning. Donc la partie de convolution dont le but est l'extraction des features est déjà faite, par conséquent les paramètres des modèles ne changent pas.

La partie responsable de la classification est la couche entièrement connectée (FCL), qui classifie les images en une voiture, animal ou arbre etc…, cette partie doit être entrainée de telle sorte qu'elle classifie les images selon ce que nous cherchons, c'est-à-dire déterminer les images spam et ham seulement. Au niveau de cette couche, nous avons une couche cachée de longueur 1000 dans le modèle VGG, dans le modèle ResNet-18 une couche cachée de longueur 100 et une couche cachée de longuer 500 dans AlexNet.

Nous avons choisi d'implémenter nos recherches par ces modèles (CNN, ResNet, VGG, AlexNet), car le ResNet résout le problème de vanishing gradient, comme nous l'avons expliqué dans la section II.4.1, en plus la notion de bloc permet de prendre l'input x tout au long de la phase de convolution ce qui permet d'extraire plus de caractéristiques de notre input (features map), par conséquent, notre modèle s'entraine d'une façon plus profonde et nous donne une bonne classification des images spam.

III.5. conclusion

Dans ce chapitre, nous avons introduit et expliqué notre approche, nous avons présenté l’architecture CNN et nous avons expliqué ses différentes couches, puis nous sommes passés de l’architecture ResNet proposée et nous avons terminé enfin par l’apprentissage par transfert.

Chapitre IV

« Implémentation et évaluation »

IV.1. Introduction

Dans ce chapitre, nous allons définir l'ensemble de données choisi, les étapes de prétraitement et l'environnement de travail et aussi expliquer les résultats des méthodes d'apprentissage proposées pour détecter les images spams et enfin nous allons discuter nos résultats et les comparer avec des résultats des recherches précédentes.

IV.2. L'ensemble de données

Dans ce projet, nous collectons 928, taille de la collection des images spam à partir de véritables e-mails en tant qu'un ensemble d'échantillons de spam. Ces images sont un sous-ensemble des images spam et ham, nous avons téléchargé cette base du site « Image Spam Hunter ». L'ensemble d'images non-spam comprend 810.

L'ensemble de données était divisé en deux ensembles :

- Ensemble d'apprentissage : 80% contient 743images spam+648 image non-spam.
- Ensemble de test : 20% contient 185 images spam+ 162 image non-spam.

IV.3. Etapes de prétraitement

Dans notre travail nous avons redimensionné toutes les entrées (images) à 224x224

- Normalisation des images, Cette étape permet d'accélérer le traitement.
- Divisé en lot notre Dataset où chaque lot a 32 images, cette étape permet le gain du temps dans la phase de l'entrainement des images.

IV.4. Outils et environnement

IV.4.1. Python

Le langage de programmation utilisé dans notre expérience est Python (version 3.7). Le langage de programmation Python a été créé en 1989 par Guido van Rossum, aux Pays-Bas. Le nom Python vient d'un hommage à la série télévisée Monty Python's Flying Circus dont G. van Rossum est fan. La première version publique de ce langage a été publiée en 1991. [37]

Ce langage de programmation présente de nombreuses caractéristiques intéressantes :

- Il est multiplateforme. C'est-à-dire qu'il fonctionne sur de nombreux systèmes d'exploitation : Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry Pi jusqu'aux supercalculateurs.

- Il est gratuit. Installable sur autant d'ordinateurs (même sur téléphone).

- C'est un langage de haut niveau. Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé.

- C'est un langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++.

- Il est orienté objet. C'est-à-dire qu'il est possible de concevoir en Python des entités qui miment celles du monde réel (une cellule, une protéine, un atome, etc.) avec un certain nombre de règles de fonctionnement et d'interactions.

- Enfin, il est très utilisé en bio-informatique et plus généralement en analyse de données.

IV.4.2. Environnement

Le langage est connu par la simplicité de sa syntaxe et les bibliothèques riches qui le rendent facile à apprendre et à utiliser à des fins différentes, Pour mener notre expérience, nous avons utilisé les modules suivants : [38]

- **Pandas** : Nous avons utilisé Pandas pour lire l'ensemble de données et transformer les résultats en tableaux faciles à lire aux formats xlsx et .csv.
- **Numpy** : pour le calcul scientifique général, remodelage des vecteurs en matrices, normalisationet d'autres opérations mathématiques.
- **Matplotlib** et Bokeh : pour la visualisation.
- **Sklearn** : utilisé pour l'évaluation et le calcul de la matrice de confusion, la précision et le rappel.
- **PIL** : utilisé pour redimensionner les images.
- **Pytorch** : nous avons utilisé le module Pytorch pour l'apprentissage en profondeur, il a de nombreuses fonctionnalités d'apprentissage en profondeur,et c'est aussi plus pythonique que le flux tensoriel, ce qui le rendchoix pour nous.Pytorch offre deux fonctionnalités de haut niveau :

- Calcul tenseur (comme Numpy) avec une forte accélération GPU.
- Réseau neuronal profond construit sur un système de diffusion automatique sur bande. [39]

Le travail a été effectué sur GoogleColab, un outil qui offre pendant 12 heures par jour :

- GPU: 1xTesla K80, ayant 2496 cœurs CUDA, calcul 3,7, 12 Go (11,439 Go utilisables).
- CPU: 1x cœur unique hyper fileté, c'est-à-dire (1 cœur, 2 fils) Processeurs Xeon @ 2,3 GHz (NonTurbo Boost), 45 Mo de cache.
- RAM: 12 Gb disponibles.
- Disque : 320 Go disponibles.

IV.5. Accuracy, Rappel, précision et F-mesure

➢ Accuracy

La justesse est l'un des critères permettant d'évaluer les modèles de classification. De façon non formelle, la **justesse** désigne la proportion des prédictions correctes effectuées par le modèle. Formellement, l'Accuracy est définie ainsi : [45]

$$\boldsymbol{Accuracy = \frac{Nombre\ de\ prédictions\ correctes}{Nombre\ total\ de\ prédictions}}$$

Pour une classification binaire, l'accuracy peut aussi être calculée en termes de positifs et de négatifs comme suit :

$$\boldsymbol{Accuracy = \frac{VP + VN}{VP + VN + FP + FN}}$$

Où *VP* = Vrais positifs, *VN* = Vrais négatifs, *FP* = Faux positifs, et *FN* = Faux négatifs.

Pour nos modèles : Spam (la classe positive 1) ou ham (la classe négative 0) :

Vrai Positif (VP) - **Réalité : spam** - **Prédiction du modèle : spam**	**Faux Positif (FP)** - **Réalité : ham** - **Prédiction du modèle : spam**
Faux Négatif (FN) - **Réalité : spam** - **Prédiction du modèle : ham**	**Vrai négatif (VN)** - **Réalité : ham** - **Prédiction du modèle : ham**

La comparaison des réponses d'un système pour une requête avec les réponses idéales nous permet d'évaluer les deux métriques suivantes :

- **Précision**

Elle est décrite comme la valeur ou la fiabilité du filtre, est calculée en divisant le nombre de messages classés par le filtre comme spam, mais ils sont vraiment limités par le nombre total d'e-mails.

$$Précision = \frac{\text{VP}}{VP + FP}$$

- **Rappel**

Le rappel il est décrit comme le nombre comparatif de messages spam que le filtre a réussi à empêcher d'entrer dans la boîte de réception des e-mails. [45]

Le facteur rappel permet de détecter si le modèle utilisé n'est pas valable puisqu'il ne détecte qu'une partie très réduite des images spam.

$$Rappel = \frac{\text{VP}}{VP + FN}$$

- **F-mesure/ F-score**

Il s'agit d'une mesure de la précision d'un test et est décrit comme la moyenne harmonique pondérée de la précision (*Ps*) et du rappel (*Rs*) du test dans une seule équation.

$$F - mesure = \frac{2 * Rappel * Précision}{Rappel + Précision}$$

IV.6. Mise en œuvre et évaluation

IV.6.1. CNN

Pour ce modèle, nous allons présenter le graphe Loss/Accuracy et la matrice de confusion.

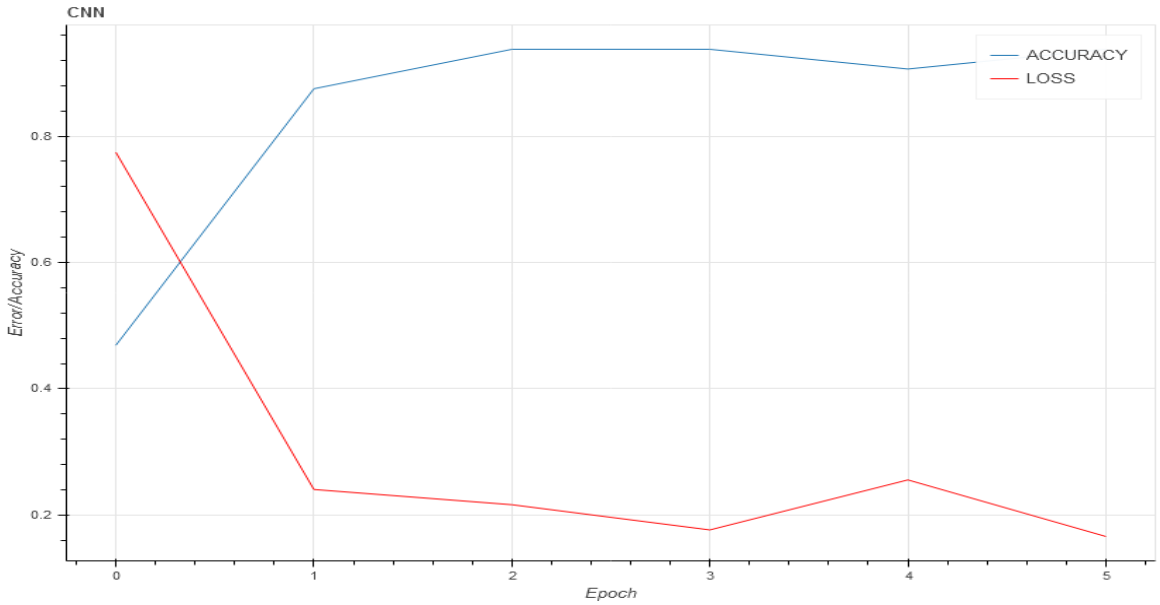


Figure IV.31 Graphe de la Loss/Accuracy de CNN

- **Discussion**

Pour le graphe de la figure 31 nous notons que l'orientation de la courbe de la précision est croissante par contre nous notons que l'orientation de la courbe de l'erreur est décroissante , cela est compatible avec l'augmentation de nombre des Epoch qui égale à 5.

- **Matrice de confusion**

L'Accuracy du modèle CNN égale à 0.95.

Rappel et précision :

	Précision	rappel	F-score
Ham	0.92	1.00	0.96
Spam	1.00	0.92	0.96

- **Discussion de la matrice de confusion**

Dans la matrice de modèle CNN, nous remarquons que :

La précision :

Les images spam (présenté par un 1) le résultat est 100%, donc le modèle a traité toutes les images spam que nous avons deviné, et pour les images non-spam (présenté par un 0) le résultat est 92%, donc y a des images que nous avons devinées non-spam (ham) mais le modèle les traite comme des images spam.

Le rappel :

C'est le cas contraire de la précision, ou les images spam (présenté par 1) le résultat est 92%, donc le modèle ne traite pas toutes les images spam, et pour les images normales (présenté par un 0) le résultat est 100%, donc le modèle traite tous les images non-spam, par exemple dans le cas d'existence des images contiennent écriture de la même couleur….

Nous avons modélisé ces cas par une matrice présente les cas possible de traitement des images spam ou non-spam.

/	Spam(1)	Non-spam(0)
Spam(1)	VP	FN
Non-spam(0)	FP	VN

- **Résultat**

  D'après le résultat du graphe et de la matrice de modèle CNN, nous avons jugé que la précision et le modèle ne sont pas assez bons pour identifier les images spam à partir de la précision. Aussi nous avons vu à travers cette expérience qu'il y'a des images spam ou le modèle les traite comme des images non-spam. Alors, pour cela nous avons décidé d'utilisé un autre modèle, meilleur que le CNN.

IV.6.2. ResNet

Pour ce modèle, nous allons présenter le graphe Loss/Accuracy et la matrice de confusion dans les figures suivantes.

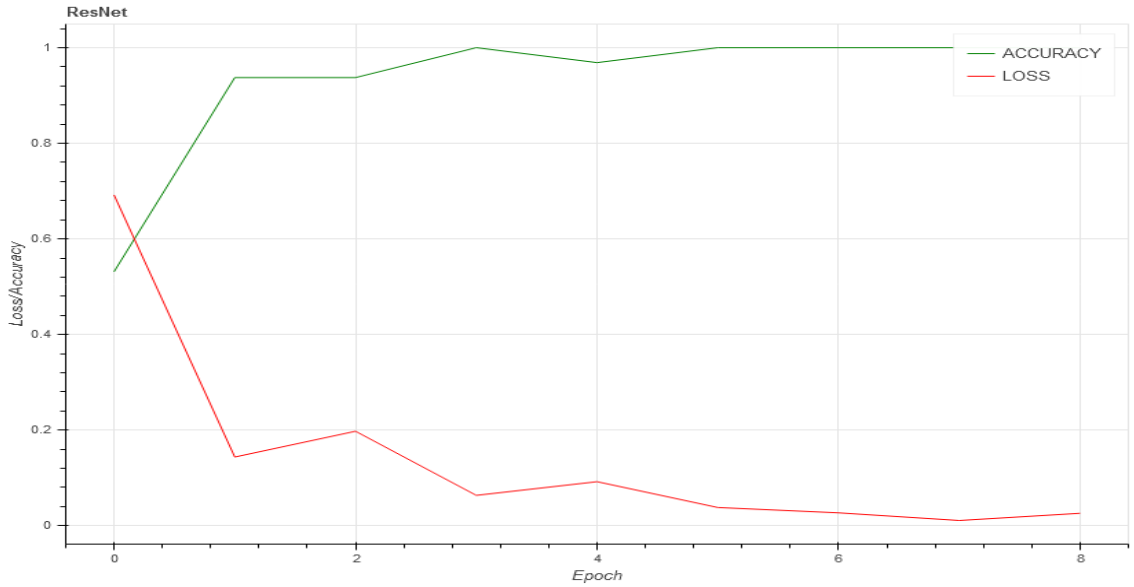


Figure IV.32 Graphe de la précision/perte de ResNet

- **Discussion du graphe**

Pour le graphe de la figure 32 nous remarquons le même résultat du graphe de la figure 33, l'orientation de la courbe de la précision est croissante par contre nous notons que l'orientation de la courbe de perte est décroissant ,cela est compatible avec l'augmentation de Epoch.

- **Matrice de confusion**

L'Accuracy du modèle ResNet égale à 0.98

Rappel et précision :

	Précision	rappel	F-score
Ham	0.98	0.99	0.98
Spam	0.99	0.98	0.99

- **Discussion**

L'Accuracy de ResNet égale à 0.98, meilleure est plus juste par rapport au CNN.

Dans la matrice de modèle ResNet, nous remarquons que :

La précision

Les images spam (présenté par un 1) le résultat est à 99%, donc le modèle a traité presque toutes les images spam que nous avons devinées, et pour les images non-spam (présenté par un 0) le résultat est 98%, donc il est bon par rapport au résultat du modèle CNN.

Le rappel

Les images spam (présenté par un 1) le résultat est à 98%, donc il s'est amélioré par rapport au modèle précédent, et pour les images non-spam (présenté par un 0) le résultat est à 99%, donc le modèle traite presque toutes les images normales.

- **Résultat**

Pour les deux graphes de la figure 31 et 32 nous trouvons que les deux modèle CNN et ResNet sont valables et fonctionnent d'une manière correcte, donc les modèles trouvent que les paramètres de l'algorithme sont bons.

Le modèle ResNet est globalement meilleur que CNN, et est meilleur pour identifier les images spam. Pour aboutir à des résultats plus précis, nous utilisons des modèles préformés, pour mettre en œuvre l'apprentissage par transfert ou nous formons que la couche de classification (la couche entièrement connectée), tandis que les poids et les biais des autres couches restent inchangés. Les modèles que nous avons utilisés sont : ResNet18, AlexNet et VGG11.

IV.6.3. ResNet-18

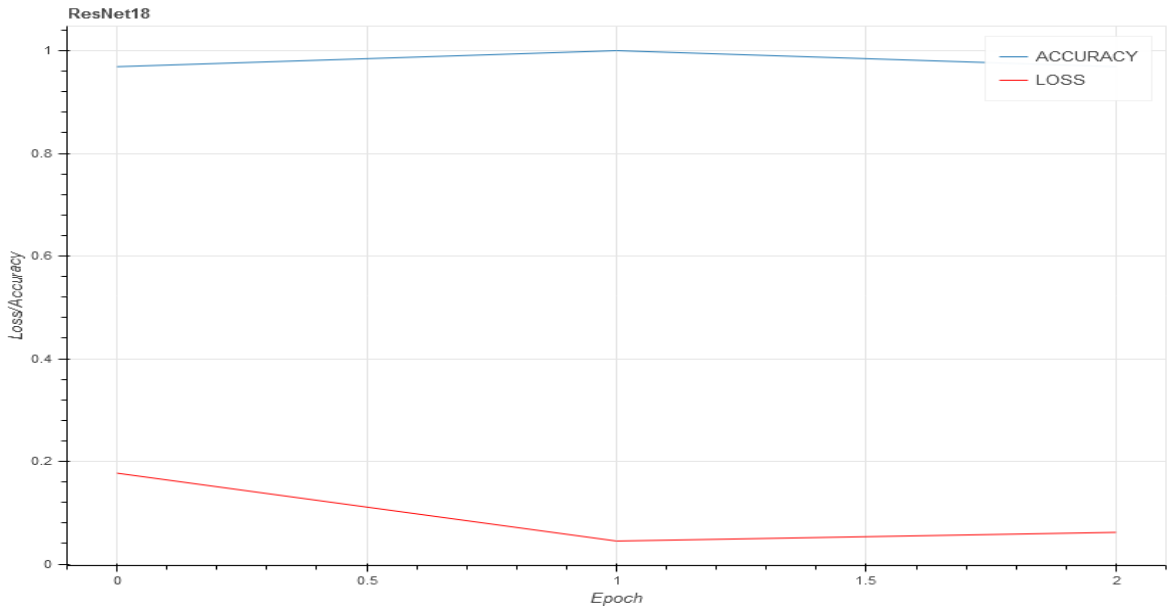


Figure IV.33 Graphe de la précision/perte de ResNet18

- **Matrice de confusion**

L'Accuracy du modèle ResNet-18 égale à 0.99

Rappel et précision :

	Précision	rappel	F-score
Ham	0.99	0.99	0.99
Spam	0.99	0.99	0.99

IV.6.4. AlexNet

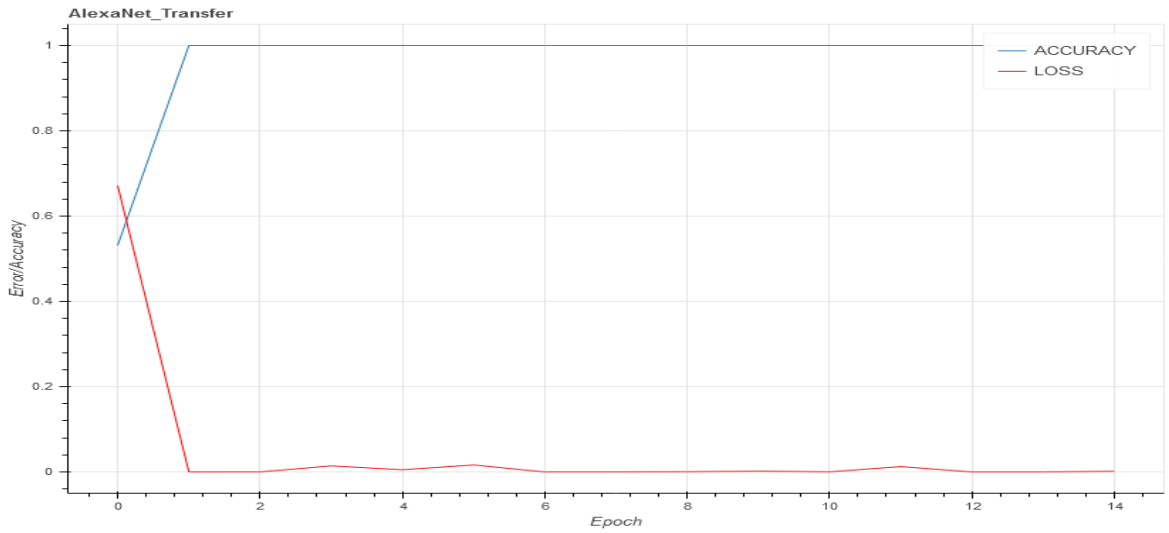


Figure IV.34 Graphe de Loss/Accuracy d'AlexNet.

- **Matrice de confusion**

L'Accuracy du modèle AlexNet égale à 0.99

Rappel et précision :

	Précision	rappel	F-score
Ham	0.99	0.99	0.99
Spam	0.99	0.99	0.99

IV.6.5. VGG-11

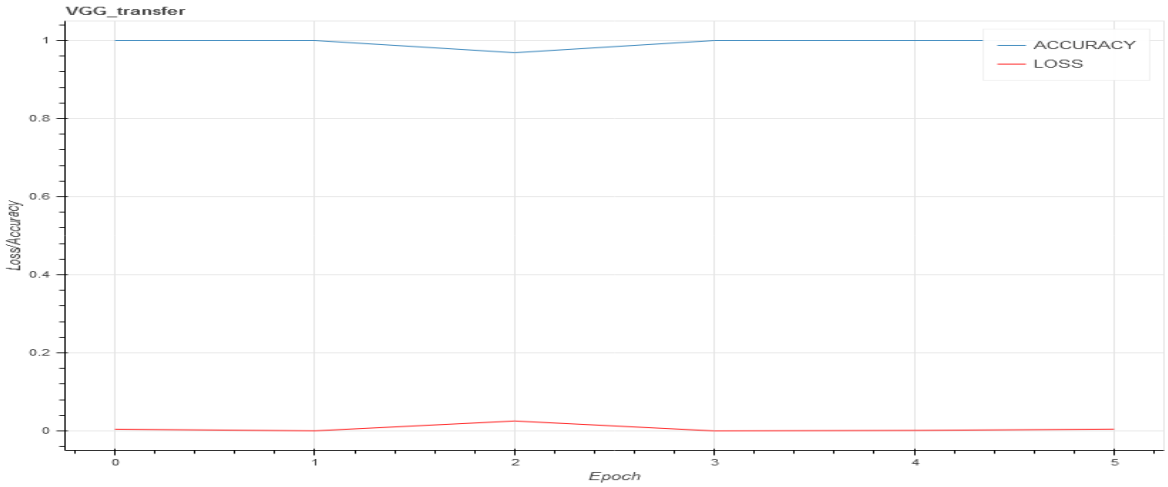


Figure IV.35 Graphe de Loss/Accuracy de VGG-11.

- **Matrice de confusion**

L'Accuracy du modèle VGG-11 égale à 0.99

Rappel et précision :

	Précision	rappel	F-score
Ham	0.99	0.99	0.99
Spam	0.99	0.99	0.99

- **Discussion des figures 33, 34 et 35**

Pour les résultats des 3 graphes et 3 matrices pour les 3 modèles « ResNet-18, AlexNet et VGG-11 » nous trouvons qu'ils ont de bons résultats ou nous trouvons que :

- La courbe de l'accurcy est croissante d'une façon régulière.
- La courbe de Loss est décroissante d'une façon régulière.
- La précision et le rappel ont des valeurs fixes pour les images spam et les images non-spam.

- **Comparaison entre les modèles**

Tableau IV.3 comparaison entre le résultat des modèles

Le modèle	CNN	ResNet	VGG	ResNet-18	AlexNet
Justesse du modèle	95%	98%	**99%**	**99%**	**99%**

Du tableau 3, le meilleur résultat revient aux procédés utilisé dans l'apprentissage par transfert avec une justesse de modèle égale à 99%, ce résultat nous permet de faire la comparaison entre nos procédés et les résultats d'autres méthodes (SVM, Arbre de décision … etc.)

IV.5. Discussion

Dans cette section, nous comparons les performances du procédé proposé avec d'autres techniques de filtrage des images spams sur la base des résultats rapportés dans la littérature pour ces techniques. Le tableau 4 montre les meilleures performances obtenues par filtres anti-spam d'image et les compare aux résultats obtenus avec CNN. Un résumé des jeux de données utilisés par chaque filtre est également fourni.

Tableau IV.4 Performances du CNN par rapport aux autres filtres anti-spam d'image.

Filtre anti-spam	**Wu et.al 2005 [41]**	**Liu et. al 2010[42]**	**Dredze et.al.[43]**	**ITA-ISF 2013[44]**	**Notre approche –CNN-**
Classificateur utilisé	1 classe SVM	SVM	Max entropie, Bayes Network (NB), Arbre de décision	SVM, Bayes Network (NB), Multi layer Perception(MP)	**Transfer Learning (ResNet-18 VGG-11 et AlexNet)**
Meilleure performance obtenue	TP=0.81-0.95	TP=0.97-0.98	A = 0.93-0.99	A=0.989 P = 0.983-0.99 R=0.994-0.995 F-score = 0.89-0.92 TP= 0.994-0.995	**A=0.994 P=0.99 F-score=0.99 TP=0.99**
Dataset utilisé	10000 spam, 1428 ham De spamArc-hive	3112 ;8719 pam,4041 ham,De spamArchive	3239 ;9503 ;127 42 spam, 2550 ham De Dezdze et al ; SpamArchive& collections Personnelles	3209 spam, 1770 ham De Dredze et al, 926 spam,810 ham, d'Image Spam Hunter (ISH)	**930 spam, 811 ham, D'Image Spam Hunter (ISH)**

Les résultats rapportés dans [41] montrent que le taux de détection de spam d'image varie de 81,4% à 95% lors de l'application d'une classe SVM sur les caractéristiques visuelles extraites de l'image spam avec une valeur aberrante dans l'anti spam filtre de 20% et 5%, respectivement. Cependant, puisque le taux de faux positifs augmente en augmentant la valeur aberrante du filtre anti-spam, les auteurs ont adopté les résultats qui correspondent à la valeur

aberrante de 20% (soit 81,4% de taux de détection de spam). Les résultats de détection d'image spam obtenus dans [42] sur la base de l'application du SVM linéaire de perte L2 et du SVM non linéaire avec le noyau RBF gaussien, respectivement, pour différents types d'ensembles de données qui représentent une combinaison d'archives de spam et d'images ham personnelles, sont comparable aux résultats obtenus par le filtre anti-spam proposé où la précision de détection du spam varie de 97% à 98% avec des taux de faux positifs compris entre 1% et 2%. Dredze et al. [43] ont montré que l'image spam peut être caractérisé par un petit ensemble de fonctionnalités, qui comprenait des propriétés de métadonnées du fichier image et d'autres fonctionnalités de bas niveau directement tirées de l'image numérisée, telles que le format de fichier, la taille, le bord et dix au hasard- pixels de couleur générés. Bien que la précision du filtre anti-spam telle que rapportée dans [43] ait atteint des valeurs élevées de 96% et 99%, il reste difficile de savoir si l'utilisation de fonctionnalités extraites des métadonnées d'image serait efficace dans les cas où les spammeurs sont en mesure d'échapper à la détection en compromettant les informations d'image. (Par exemple, continuez à changer les couleurs d'arrière-plan de l'image).

Comparant nos résultats avec les résultats d'ITA-ISF qui ont entrainé leur modèle sur la même base que nous avons utilisé, ils ont obtenu une Accuracy qui égale à 98% en utilisant des techniques plus anciennes (Bayes Network, Multi layer Perceptron), quant aux résultats obtenus dans nos recherches ont atteint une précision de filtre anti spam de 99%, cela, dépasse tous les résultats des recherches mentionnées précédemment, avec un taux de Vrai Positif (TP) égale à 99%.

Donc le CNN dans ses versions modernes (VGG, AlexNet…etc.), est une très bonne technique de classification par rapport aux anciennes techniques, Les CNN réalisent eux-mêmes tout le boulot fastidieux d'extraction et description de *features* lors de la phase d'entraînement, l'erreur de classification est minimisée afin d'optimiser les paramètres du classifieur et les *features* ! De plus, l'architecture spécifique du réseau permet d'extraire des *features* de différentes complexités, des plus simples au plus sophistiquées. L'extraction et la hiérarchisation automatiques des *features*, qui s'adaptent au problème donné, constituent une des forces des réseaux de neurones convolutifs, il donne donc de meilleurs résultats. Nous voyons que notre approche maintient un faible taux de faux positifs tout en atteignant une précision de classification élevée et qu'elle fonctionne très bien par rapport à ces filtres anti-spam.

IV.6. Conclusion

Dans ce chapitre, nous avons détaillé notre mise en œuvre, nous avons décrit l'environnement du travail et les différents outils et les bibliothèques utilisés en expliquant brièvement le rôle de chacune d'entre elles. Puis nous sommes passés à l'évaluation des approches expliquées dans le chapitre 3. Nous avons commencé par présenter et analyser notre ensemble de données que nous avons choisi. Ensuite, nous avons proposé une méthode d'apprentissage automatique pour détecter les images spam, nous l'avons expliqué par des résultats de notre application présentés avec des graphes et des matrices.

Conclusion générale

Grâce à des techniques de spam améliorées, les spammeurs ont réussi à échapper aux techniques de détection de spam traditionnelles telles que la détection basée sur le contenu et l'OCR. Cela a ouvert la porte à des techniques telles que le traitement d'image pour détecter le spam d'image. Une combinaison d'algorithmes d'apprentissage automatique et de traitement d'image peut être utilisée pour constituer des classificateurs puissants. Nous avons développé un classificateur utilisant les réseaux de neurones convolutionels (CNN) avec des propriétés d'image comme ensemble de fonctionnalités, qui a fourni de bons résultats dans la détection des images spam sur un ensemble de données public.

Les réseaux de neurones convolutifs (CNN, ResNet et le Transfer Learning) se sont révélées être les meilleures parmi les algorithmes d'apprentissage supervisé hautement discriminants. Pour cette approche, nous extrayons d'abord un large ensemble caractéristique (*features map*) en s'appuyant sur le fait que l'image spam possède des caractéristiques différentes par rapport à une image authentique, ensuite nous classifions ces images à l'aide de la couche entièrement connectée. Les résultats obtenus par ces classificateurs sont jugés très bons avec une précision pouvant atteindre le seuil de 99%.

Si les spammeurs proposent un nouveau type de spam image, ces schémas de détection devront être ajustés à nouveau pour générer un nouveau modèle d'apprentissage. Au lieu de cela, si nous avons un schéma de détection qui apprend par lui-même, il gérera lui-même ces mises à niveau. Les réseaux de neurones ont le potentiel de nous fournir un tel schéma de détection. Même si une solution basée sur les réseaux neuronaux peut nécessiter beaucoup de calculs, la possibilité de développer un classifieur qui apprend par lui-même semble être une bonne direction.

Le but que nous avons fixé au début « *Le but du projet est d'utiliser spécifiquement les réseaux de neurones convolutifs sur le problème de la classification des images spam afin d'obtenir de meilleurs résultats que ceux qui sont obtenus dans les articles précédents*» est donc atteint avec un résultat très satisfaisant.

Perspective

Il est à noter, que nous avons téléchargé et utiliser directement la base de données d'images (Image Spam Hunter). Il sera souhaitable dans l'avenir d'ajouter d'autres images pour augmenter la capacité de la base de données.

Références

[1] Aneri Chavda, Image Spam Detection. San José State University. 21/5/ 2017.

[3] A Spam Filtering Method Based onMulti-Modal Fusion.

[4] A survey of image spamming and filtering techniques. August 11, 2011.

[5] Machine learning for email spam filtering : review, approaches and open research problems. 2019.

[6] A. Bhowmick, S.M. Hazarika, Machine Learning for E-Mail Spam Filtering : Review, Techniques and Trends. 2016.

[7] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, C.D. Spyropoulos, An experimental comparison of naïve Bayesian and keyword-based anti-spam filtering with personal e-mail messages.

[8] K.P. Clark, A Survey of Content-Based Spam Classifiers. 2008.

[9] J.S. Whissell, C.L.A. Clarke, Clustering for semi-supervised spam filtering in : Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-abuse and Spam Conference. 2011.

[10] S. Roy, V.M. Viswanatham, P.V. Krishna, N. Saraf, A. Gupta, R. Mishra, Applicability of rough set technique for data investigation and optimization of intrusion detection system, in : Quality, Reliability, Security and Robustness in Heterogeneous Network. Springer Berlin Heidelberg. 2013.

[11] Z.I. Pawlak, Rough Sets : Theoretical Aspects of Reasoning about Data, Kluwer Academic. New York. 1991.

[12] N. P_erez-Díaz, D. Ruano-Ord_as, F. Fdez-Riverola, J.R. M_endez, Rough sets for spam filtering : selecting appropriate decision rules for boundary classification, Appl. Soft Comput. 13 aout 2012, pages 1–8.

[13] N.B. Agnieszka, Mining rule-based knowledge bases inspired by rough set theory, 2016.

[14] D. Sculley, G. Wachman, in : W. Kraaij, A.P. de Vries, C.L.A. Clarke, N. Fuhr, N. Kando (Eds.), Relaxed Online SVMs for Spam Filtering, SIGIR, ACM, 2007.

[15] V. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, 1995.

[16] Z.S. Torabi, M.H. Nadimi-Shahraki, A. Nabiollahi, Efficient support vector machines for spam detection, 2015.

[17] S.K. Chen, Y.H. Chang, SVM classifier algorithm, International Conference on Artificial Intelligence and Software Engineering, 2014.

[18] L. Deng, D. Yu, Deep Learning : Methods and Applications, Boston, 2014.

[19] M. Nixon, Feature extraction and image processing, 2008.

[21] Valentina E. Balas, Liliana Perescu-Popescu, Nikos Mastorakis,
Multilayer Perceptron and Neural Networks, septembre 2016.

[22] LeCun Y., Jackel L., Boser B., Denker J., Graf H., Guyon I., Henderson D., Howard R, and Hubbard W,
Handwritten digit recognition : Applications of neural networks chipsand automatic learning. Proceedings of the IEEE.

[23] Farhana Sultana, Abu Sufian, Paramartha Dutta. advancements in images classification using Convolutionel Neural Network.

[25] Deepika Jaswal, Sowmya.V, K.P, Soman Image Classification Using Convolutional Neural Networks, june 2014.

[27] E. Mohammad Sadegh et A. Hossein Karkeh, Study of Residual Networks for Image Recognition, 2015.

[28] H.Kaiming, Z. Xiangyu, R. Shaoqing and S. Jian,
Deep Residual Learning for Image Recognition, 2015.

[31] Simonyan, Karen, & Andrew Zisserman,
" Very deep convolutional networks for large- scale image recognition, 1998.

[32] Deep Learning and Transfer Learning Approaches for Image Classification, 2019.

[33] CNN Architecture Series.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton.
Imagenet classification with deep convolutional neural networks, 2012.

[36] Chiman Kwan, David Gribben, Trac Tran,
Tracking and Classification of Multiple Human Objects Directly in Compressive Measurement Domain for Low Quality Optical Videos.

[37] Patrick Fuchs et Pierre Poulain, Cours de Python. Paris, 2020.

[40] Chergui Oussama, Chergui Meriem,
Image classification using supervised Deep Learning Méthodes, 2019.

[41] C. Wu, K. T. Cheng, Q. Zhu, and Y. L. Wu,
“Using Visual Features for Anti-Spam Filtering,” in ImageProcessing", 2005.

[42] Q. Liu, Z. Qin, H. Cheng, and M. Wan,

"Efficient Modeling of Spam Images," in Intelligent Information Technology and Security Informatics (IITSI), 2010.

[43] R. G. Mark Dredze and A. Elias-Bachrach,
"Learning Fast Classifiers for Image Spam," presented at the in Proc, 2007.

[44] Detecting Image Spam Using Image Texture Features, 2013.

[45] Evaluation des performances dans les SRI.

[47] Lisa Torrey and Jude Shavlik, Transfer Learning.

Webographie

[2] Les chiffres 2019 de l'e-mail. Récupéré sur Les chiffres 2019 de l'e-mail : https://www.arobase.org/actu/chiffres-email.htm, date d'accès : 25 février 2020.

[20] Neural Networks and Introduction to Deep Learning. Date d'accès : 21 Juin 2020.

[24] Tutorial on deep learning [Online] valable sur : Récupéré sur http://deeplearning.net/tutorial/lenet.html, date d'accés : 9 Avril 2020.

[26] How do I calculate backpropagation on a global average pooling layer ? Récupéré sur https://www.quora.com/How-do-I-calculatebackpropagation, 2018. 27 Mai 2020.

[29] M. Rizwan, Residual Networks (ResNets). Récupéré sur
https : //engmrk.com/residualnetworks-resnets, octobre 3 2018. Date d'accès : 12 Mai 2020.

[30] Jason Brownlee, A Gentle Introduction to Transfer Learning for Deep Learning.
Récupéré sur machine learning mastery : https://machinelearningmastery.com/transfer-learning-for-deep-learning, December 2017. Date d'accès 12 Mai 2020

[35] Architecture of AlexNet and its current use.
Récupéré sur https : //iq.opengenus.org/architecture-and-use-of-alexnet, date d'accès : 6 Juin 2020.

[38] Hasan, «Top 20 Most Popular Programming Languages To Learn For Your Open source Project».
Récupéré sur https://www.ubuntupit.com/top-20-mostpopular-programming-languages-to-learn-for-your-open-source-project/. Date d'accès : 22 Mai 2020.

[39] Wikipedia, «pytorch,» [En ligne]. Valable : https://en.wikipedia.org/wiki/PyTorch. Date d'accès 3 Juin 2020.

[46] Machine Learning. Récupéré sur Crush-course-classification : https://developers.google.com/machinelearning/crashcourse/classification/accuracy?hl=fr

Date d'accès : 15 Juin 2020.